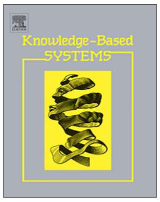
Knowledge-Based Systems 200 (2020) 105980

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Language model based interactive estimation of distribution algorithm

Yang Chen [a,c], Yaochu Jin [b], Xiaoyan Sun [a,*]

[a] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China
[b] Department of Computer Science, University of Surrey, Guildford, GU4 7YX, United Kingdom
[c] School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 23 September 2019
Received in revised form 27 April 2020
Accepted 27 April 2020
Available online 30 April 2020

Keywords:
Estimation of distribution algorithm
Interactive evolutionary algorithm
Language model
Personalized search
Bayesian inference

ABSTRACT

It is very hard, if not impossible to use analytical objective functions for optimization of personalized search due to the difficulties in mathematically describing qualitative problems. To solve such optimization problems, interactive evolutionary algorithms, which can make use of human preferences, are highly desirable. However, due to the lack of effective encoding methods, interactive evolutionary algorithms have been limited to numerically encoded optimization problems. In practice, however, linguistic terms (words) are the most natural expression of human preferences, and they are also commonly used to describe items in personalized search or E-commerce; therefore, language models better suit encoding, and the optimization of personalized search is converted into a dynamic document matching problem. To optimize word-described personalized search, we propose a novel interactive estimation of distribution algorithm. This algorithm combines a language model-based encoding approach, a Dirichlet-Multinomial compound distribution-based preference expression, and a Bayesian inference mechanism. The proposed algorithm is applied to two personalized search cases to demonstrate the capability of the algorithm in ensuring a more efficient and accurate search with less user fatigue.

1. Introduction

Objective functions of personalized optimization problems, e.g. product design, personalized search, and information retrieval, are impossible to precisely define using mathematical expressions since they are highly dependent on user preferences. Although evolutionary computation (EC) has been proven to be a powerful tool for solving complex optimization [1–3] problems, the requirement of precise mathematical definitions cannot be fulfilled in personalized search. In such scenarios, interactive evolutionary computation (IEC) [4,5] is more feasible and efficient as it involves a human user in the evaluation process; they have been developed and successfully applied to various practical problems, such as product design, web page layout design, and anti-collision design of vehicles [5–8]. This paper considers the optimization problems that occur in, for example, the following scenario: A person is searching the web for a particular movie, beginning the search with a query of a few words. The search engine presents a few movie candidates. The user clicks on some of the candidates and saves some of them. Based on the user's actions, the search engine presents some new candidates. This process continues until the user is satisfied with the result. The purpose of this proposed novel method is to speed up the search process and reduce the need for user interference.

IEC requires human evaluations, which can inevitably cause user fatigue given a complex problem. The restriction on population size and evolutionary generation prevents the use of IEC in tackling a wide range of problems [5]. Accordingly, much more attention has been paid to alleviating user fatigue and improving explorations from the following three aspects [9]: (1) the design of friendly human–computer interfaces or novel evaluation modes to reduce the user burden, e.g. evaluating individuals with discrete, fuzzy, or interval numbers [9–11]; (2) the use of a preference surrogate, with a small number of evaluated individuals and then apply it to help with the assessment. With surrogates, IEC can upscale the population size and generation as conventional EC approaches [12,13], which greatly improves the explorations of IEC. (3) the use of knowledge from evolution to modify evolutionary operators to accelerate search and reduce fatigue [14,15]. These studies have greatly enhanced IEC. Although the above-mentioned studies have greatly enhanced IEC , the application of those methods in resolving preference-related complex problems remains a challenging task. Particularly in word-described ones such as online personalized search in E-commerce.

The main reason is that the information covered by the numerical representations for these word-described problems is insufficient to model the preference-related objectives, leading to inefficient or even wrong searches. Sun et al. [9,16] used a limited

[*] Corresponding author.
E-mail address: xysun78@126.com (X. Sun).

https://doi.org/10.1016/j.knosys.2020.105980
0950-7051/© 2020 Elsevier B.V. All rights reserved.

number of numerical values to encode these word-described items in the framework of an interactive genetic algorithm for the personalized search, which made the traditional evolutionary operators easier to implement. Wang et al. [17] modeled TV programs with five attributes in their experiments when studying preference recommendations for personalized search.

These studies are easier to understand and implement; however, such numerical encoding loses a considerable amount of implied semantic information contained in the words. In addition, IEC depends on the evaluations assigned by the users, who have got used to thinking and evaluating with words instead of numbers. The gap between the users' evaluations and numerical encoding results in a need for additional human–computer interactions and inevitably causes more user fatigue. Accordingly, designing a non-numerical encoding method which minimizes loss of semantic information and developing corresponding evolutionary operators becomes essential as this will help to enhance the performance of IEC in solving more practical and complex problems. Furthermore, user preferences or decisions will be influenced greatly by other user comments, and social or group comments should be integrated with IEC so that the current user is able to precisely evaluate the searched solutions. Motivated by the above, we focus on developing an enhanced IEC by integrating social comments, designing a novel encoding and corresponding evolutionary operators for solving problems described with words in the personalized search.

Applying IEC, short for Interactive Evolutionary Computation, to the word/document-described optimization problems relies on establishing a bridge between the textual phenotype evaluated by the user and the numerical genotype operated by EC. The language model Doc2Vec [18,19] is a good choice to convert textual phenotypes into numerical vectors by preserving most of the semantic relationships among the words. Therefore, we employ this model to represent the genotype of a document, i.e., a searched item, including the word description and social comments. Clearly, both are naturally combined in the Doc2Vec based representation. Given this, a new IEC must be developed to gain the advantages both from itself and the model, i.e., Doc2Vec-assisted initialization and interactive evolutionary operators.

This work develops a language model based interactive estimation of distribution algorithm (LMIEDA) to perform the evolutionary optimization in personalized search. In LMIEDA, the Doc2Vec is applied to convert the word/document-described problem into a dynamic document matching one by encoding the word frequency as individuals. A preference function is approximately constructed based on user interactions and is used to estimate the individual's fitness. Based on the fitness, the Dirichlet-Multinomial compound distribution and a Bayesian inference involving the Dirichlet-Multinomial compound distribution is designed to track the user's preference on the variables. With these, the probabilistic model of estimation of distribution algorithm (EDA) and the corresponding sampling are presented. The user's burden here can be greatly reduced since our algorithm is able to estimate the fitness of all individuals without the user.

The main contributions of this study are as follows. (1) To the best of our knowledge, language models have not been used in EC/IEC. As an encoding method, it helps to reduce information loss and naturally introduces social intelligence. (2) Since the encoded variables are discrete (with finite states), the Dirichlet-Multinomial compound distribution is adopted as the probabilistic model of IEDA to be compatible with encoded candidates. (3) The probabilistic model is updated with the help of Bayesian learning to directly track variable distribution, and most conventional EDAs employ Bayesian networks to depict the dependencies between variables. (4) The proposed algorithm is applied to some personalized search for books and movies to prove its effectiveness and efficiency.

The remainder of this paper is organized as follows. Section 2 describes the related work on the personalized search assisted with evolutionary algorithms, the estimation of distribution algorithms (EDAs), and the basic concept of Doc2Vec. The details of the proposed algorithm, including the definition of the word-described personalized search, the framework, the critical encoding, preference expression, and the IEDA, are presented in Section 3. Section 4 addresses the application of the proposed algorithm together with the experimental results and analyses. Conclusions are drawn in Section 5.

2. Related work

2.1. Personalized search assisted with evolutionary algorithms

The task of the personalized search is to find the items that give the user the most satisfaction; therefore, it is an optimization problem in nature. However, what distinguishes personalized search from typical optimization problems is that users, rather than mathematical functions, play the role of the fitness functions. Although it is hardly possible to solve this problem involving cognitive processes only with tractable mathematical calculations, researchers can still describe some algorithms and approximate results in mathematical language. Moreover, solutions can be obtained with the help of those studies involving humans.

For searching the most desired items, user preferences on the searched items reflected by the corresponding interactions are usually quantified and tracked in E-commerce or personalized search [20,21]. Some scholars have conducted relevant studies for addressing the problem where preference is uncertain and hesitant [22–24]. Besides, some research has emphasized on the modeling of user preferences using obtained ratings or interactions without optimization [25,26]. Fujita et al. proposed a zone partition method based on GrC to improve awareness and support rapid decision making at early stages of intelligence analysis [27]. EC approaches have been applied to optimize the parameters of the preference model. To obtain reliable preference models, some research focused on applying EC to optimize or update their structures or parameters rather than optimize the search process [28,29].

Another way to apply EC in personalized search is to develop IEC since the user can easily evaluate the items and express their preferences by means of ranking or interactions. None of the preference models obtained from user interactions in the majority of the related studies is integrated with IEC to accelerate the search process. For the purpose of combing the evolution strategy with the agent-based model to find the rational behavior of a user, Ahn [30] used an agent-based model to imitate rational user behaviors with regard to browsing and collecting product information. To gain advantages from content-based recommender techniques, Kim et al. [31] presented an innovative recommender system to dynamically track user preferences on music; Kant et al. [32] utilized Reclusive Methods (RMs) to handle uncertainty, and the Interactive Genetic Algorithm (IGA) was employed in information retrieval. In previous work [33], the authors present a personalized search inspired fast interactive estimation of distribution algorithm (PSIEDADK) by using the domain knowledge of personalized search. The IEDA was enhanced by maintaining a Bayesian model that described the distribution of user's preference on variables.

In addition, the existing EC-based personalized search [9,16,17] encoded the items with numerical values without considering social intelligence and word/document descriptions, which undermines the accuracy of the modeling. Moreover, encoding the individuals with numeric values results in the contradictions between the mechanism of the EC-part and the human cognitive process within the IEC framework. [34].

2.2. Estimation of distribution algorithms

EDAs [35,36] macroscopically reveal a large amount of information about the population by building an explicit probabilistic model of promising candidate solutions. Then, the offspring population is generated by sampling from a probabilistic model [37–40]. Several methods of building and sampling from the distribution of the population have been proposed, including methods based on dependent chains/trees, factorization, neural trees, Bayesian networks, and genetic programming [37].

Since EDAs are a novel evolutionary optimization paradigm based on genetic algorithms and statistical learning, it is quite natural to integrate a Bayesian learning framework into EDAs. However, little research on this has been reported and researchers preferred taking the Bayesian network to decompose problems [37,39,41–43]. In [44], Zhang proposed Bayesian evolutionary algorithms (BEAs) in which optimization was formulated as a probabilistic process of finding an individual with the maximum a posteriori probability (MAP). Even with the name that looks like a method belonging to Bayesian statistics, BEAs are frequentist since prediction in frequentist statistics often involves finding an optimum point estimate of the parameters and then plugging this estimate into the formula for the distribution of a data point, whereas Bayesian theory calls for the use of the posterior predictive distribution to do predictive inference, i.e. instead of a fixed point as a prediction, a distribution over possible points is returned. In [43], Zhang and Shin proposed a method that estimates the sample distribution with a graphical learning model: the Helmholtz machines. Essentially, this method makes the optimization a sequential probabilistic process of finding the parameters with the maximum likelihood estimation (MLE). Both of the methods above addressed the probabilistic process in a frequentist setting instead of a Bayesian setting, in which a distribution rather than an estimated value could be generated.

With the Bayesian setting, uncertainty and more information over values of the target variable can be expressed using a probability distribution. However, little research on the fusion of the Bayesian learning framework and interactive estimation of distribution algorithms (IEDAs) has been performed.

2.3. Language model Doc2Vec

As far as we know, language models have not been studied in EC/IEC to solve problems described by documents/texts. However, many language models have been proposed over the past decades to model text in the field of Natural Language Processing (NLP), e.g. Bag-of-Words (BOW) [45], TF-IDF [46], Latent Semantic Analysis (LSA) [47], and Probabilistic Latent Semantic Analysis (PLSA) [47]. Moreover, many studies employed word-based models to a wide range of applications [48,49]. Recently, Latent Dirichlet Allocation (LDA) [50], Word2Vec [18], and Doc2Vec [19] have achieved remarkable results in many NLP and machine learning (ML) tasks [51–53]. Particularly, the Doc2Vec achieves state-of-the-art best performance on sentiment analysis [19,54,55]. The focus here is Doc2Vec since it will be used to encode individuals in our work.

Word2Vec is a powerful tool for distributed representation of words with vectors and was presented by Google in 2013. This model uses deep learning to simplify the processing of text to the vector operation in n-dimensional vector space, and the similarity of text semantics is expressed by that in vector space. Word2Vec is widely used in NLP tasks due to its high efficiency and rich semantic information [56–58]. As an extension of Word2Vec, Doc2Vec generates a fixed-length feature as the representation for large blocks of text, e.g. sentences, paragraphs, or entire documents.

Inspired by recent work on learning vector representations of words with neural networks [18,59,60], Le et al. [19] proposed Doc2Vec by concatenating the document vector with a number of word vectors from a document and predicted the following words in the given context; they conducted the research with neural-network-based paragraph vectors, also known as neural language models, under an unsupervised framework.

If an item described with words can be represented with vectors by using Doc2Vec, the word/document-described optimization problems are expected to be effectively solved by using EC/IEC without losing semantic relationships among words.

3. Language model based interactive estimation of distribution algorithm

3.1. Definitions of word/document-described optimization problems

$$\begin{cases} \max & f\left(document\right) \\ \text{s.t.} & document \in H \end{cases} \tag{1}$$

For the word-described personalized search, by naturally combining social intelligence from comments, an item can be expressed as $document = \{description\} \cup \{comments\}$, in which the first part comes from its seller, and $\{comments\}$ with specific meaning on the item come from users. Supposing a user's preference on a *document* is $f\left(document\right)$, the search can be formulated as Eq. (1), where H is the feasible space of searched items. Evolutionary algorithms will be powerful for solving such problems if the objective values , i.e. the fitness, can be calculated. Unfortunately, the value of $f\left(document\right)$ is difficult to explicitly compute since the variable *document* and the preference $f\left(\cdot\right)$ are both qualitative. IECs are more practicable in such cases since an active user is involved in presenting the preferences through direct scoring or interaction-based evaluations. To perform IEC, the variables and the evaluation function must be quantified.

In the field of EC, no algorithm has been developed to encode a *document* into a quantified individual without loss of the semantic relationships. We use the Doc2Vec to convert the word/document-described problem into a dynamic document matching one. The ith *document*, termed as d_i, is regarded as a list of word events without ordering, $\left(w_{d_{i,1}}, w_{d_{i,2}}, \ldots\right)$, in which the $w_{d_{i,k}}$ is a word w_j with position k in the document d_i with w_j coming from the vocabulary $V = \left(w_1, w_2, \ldots, w_{|V|}\right)$. By calculating the word frequency, the vector $\boldsymbol{m}= \left(m_1, m_2, \ldots, m_{|V|}\right)$ could be obtained. Each element indicates the frequency corresponding to each word from the vocabulary, e.g. $m_{|V|}$ is the frequency of the last word in the vocabulary; then, the original problem defined in Eq. (1) can be transformed into a document matching one by optimizing the vector $\boldsymbol{m}$. Concretely, in the proposed LMIEDA (language model based interactive estimation of distribution algorithm), vector $\boldsymbol{m}$ is adopted as candidate document feature. Moreover, as in typical IEC approaches, LMIEDA builds a probabilistic model of promising solutions. The evaluation of the document is given by the user, and a user's preference $f\left(\cdot\right)$ can be approximated as $\hat{f}\left(\cdot\right)$ through her/his interactions. Thus, the problem in Eq. (1) can be further expressed with Eq. (2), where G is the feasible solution space, the value of $\hat{f}\left(\boldsymbol{m}, t\right)$ represents an approximated evaluation on a solution $\boldsymbol{m}$, and t indicates that the value is dynamically updated according to user's interactions. The construction of the vocabulary will be addressed in Section 3.2.

$$\begin{cases} \max \quad E[\hat{f}\,(\boldsymbol{m}, t)] \\ \quad \text{s.t.} \quad \boldsymbol{m} \in G \end{cases} \tag{2}$$

As for document matching, the similarity measure between two documents is critical. With the help of Doc2Vec, we quantify the variable *document*, corresponding to each candidate, as $\boldsymbol{x} = \text{Doc2Vec}(document)$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ with n being the dimension of semantic vector space, where x_i varies in its domain S_i. Moreover, *document* is defined in Table 1. Accordingly, as that is commonly adopted in the text analysis, the Cosine similarity between two documents can be calculated based on their corresponding semantic vectors.

Table 1 presents a summary. The document, the word frequency, and the semantic vector are three expressions of the same document in different spaces; they are all inputs and would be adopted according to different scenarios for making formulas more accurate and understandable.

Given the interactions depicting user preferences on evaluated items being quantitatively expressed, a method is needed to generate offspring based on the quantitative preference, and this could be achieved by iterations. Specifically, a probabilistic generative model is chosen to generate documents, which is expected to dynamically track user preferences. Inspired by the related research [61], the mixture of unigrams is adopted, under which documents are generated by first selecting a topic z for each word and then generating words independently from the conditional multinomial distribution $p\,(w\,|z\,)$. In addition, social intelligence is naturally involved in our method since other users' evaluations of searched items are all covered by this model. With this model, items, e.g. books or movies, can be encoded into individuals. It is noted that this study considers the document description of items in personalized search. EC employs population-based intelligence to solve complex optimization problems, and the population consists of a certain number of individuals. Each individual is a candidate solution to the optimization problem, and the number is population size.

3.2. Main framework

The main framework of our algorithm is shown in Fig. 1. With the help of it, the complete proposed algorithm is first summarized to enhance readers' understanding. As shown, the whole LMIEDA comprises offline and online computation. By training Doc2Vec, offline computation offers encoded items, a vocabulary used for generating individuals, and vector space where the search is conducted. Social comments on each item are collected and involved in this training, which can guarantee a more precise description of items in vector space.

The other component is online computation. In personalized search for interesting movies, an active user first requests a search service by inputting queries when the LMIEDA ***Begins***. Based on the queries, non-personalized ***Item Filtering*** is carried out with a pre-filtered list returned through simply matching the offered queries with the movies stored in a database. However, the number of relevant entries returned from the database could be too overwhelming and become infeasible for the users to evaluate each. Here, the designed interactive evolutionary mechanism helps.

The presented approach LMIEDA falls into the category of EDA where an explicit probabilistic model is built and updated. Specifically, the model initialization is first carried out by (***Randomly Setting Parameters of Probabilistic Model***). Upon the probabilistic model, a population consisting of individuals, i.e., word frequencies, are formed through sampling. Then, a list of items are matched with those individuals for user evaluations. In the following ***Preference Model Assisted User Evaluation***, according to her/his answer to ***if the user feels satisfied or tired?*** during evaluation, the algorithm ***ends*** or continues.

When it is not terminated, LMIEDA conducts ***Probabilistic Model Construction (Update)*** with evaluated individuals. Then, evolution continues in ***Sampling to Generate Offspring*** with the updated model. Offspring individuals generated by the EDA are converted into vectors through the trained model, and these vectors are then compared with stored items. The most similar ones are selected as phenotypes and presented to the user for further interactive evaluations in ***Preference Model Assisted User Evaluation***. Doc2Vec-based conversion, vector comparisons, and EDA evolution iterates until the user finds satisfying solutions. Such a search is expected to be of high efficiency since the user's real-time preference and effective EDA are involved to guide it.

Three critical issues shaded in the figure are explained as follows:

(1) Offline computation consists of the following three substeps: Doc2Vec training, item encoding, and vocabulary generation. Doc2Vec is trained with the corpus made up of the existing descriptions together with the corresponding comments of the stored items and Wikipedia corpus. Usage of Wikipedia is expected to bring more general knowledge to the model because of the limited amount and scope of comments.

Items to be searched are encoded as fixed-length vectors in semantic space by the well trained Doc2Vec.

For the mixture of unigrams, a vocabulary with semantically dissimilar words is needed. To guarantee the dissimilarity, these words are selected by SVM, KMeans and Doc2Vec. Specifically, their expressions in the vector space are first obtained; then, they are classified using SVM and KMeans; the vocabulary is finally generated by evenly choosing tags from those groups. All words in this vocabulary are treated as variables, and a combination of them is just a document-described individual; i.e., sampling and combining words from the vocabulary can get individuals. Then, the items are encoded with Doc2Vec into fixed-length vectors, which reserve semantic information well.

(2) Preference-model-assisted user evaluation: A preference model is constructed according to user interactions, which is usually used as a surrogate to approximate the fitness of individuals for further implementing the selection operation. In our algorithm, however, its role is quite different. The constructed preference model here is only used to rank user evaluated items; then, their corresponding ordering is applied to update the probabilistic model of EDA.

(3) Bayesian inference assisted EDA: Inspired by the Dirichlet-Multinomial compound distribution or mixture of unigrams used in Latent Dirichlet Allocation (LDA) [50] to generate new documents, the distribution is selected to sample words for forming individuals. It serves as the probabilistic model of EDA since it is optimal at producing new documents in a content-based search [50]. Parameters of this model are crucial and here updated based on Bayesian inference and the preference model. In the initialization, all parameters in probabilistic distributions are randomly given.

More details about (2) and (3) will be given in the following descriptions of our language model based IEDA.

3.3. Language model based IEDA

The presented IEDA consists of two parts, one is the user interactions based quantified evaluations on the individuals, and the other is the design of the probabilistic model of the EDA. According to the quantified preference model, the excellent individuals are selected to construct the probabilistic model, and also

Table 1
Notation.

Notation	Formula	Used by	Usage
Document	$\boldsymbol{d}_i = \left(w_{d_{i,1}}, w_{d_{i,2}}, \ldots\right)$	User	Evaluation
Word Frequency	$\boldsymbol{m} = \left(m_1, m_2, \ldots, m_{\lvert V\rvert}\right)$		Optimization
Semantic Vector	$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$	Algorithm	Measurement of item/individual similarity
Vocabulary	$V = \left(w_1, w_2, \ldots, w_{\lvert V\rvert}\right)$		Offspring generation

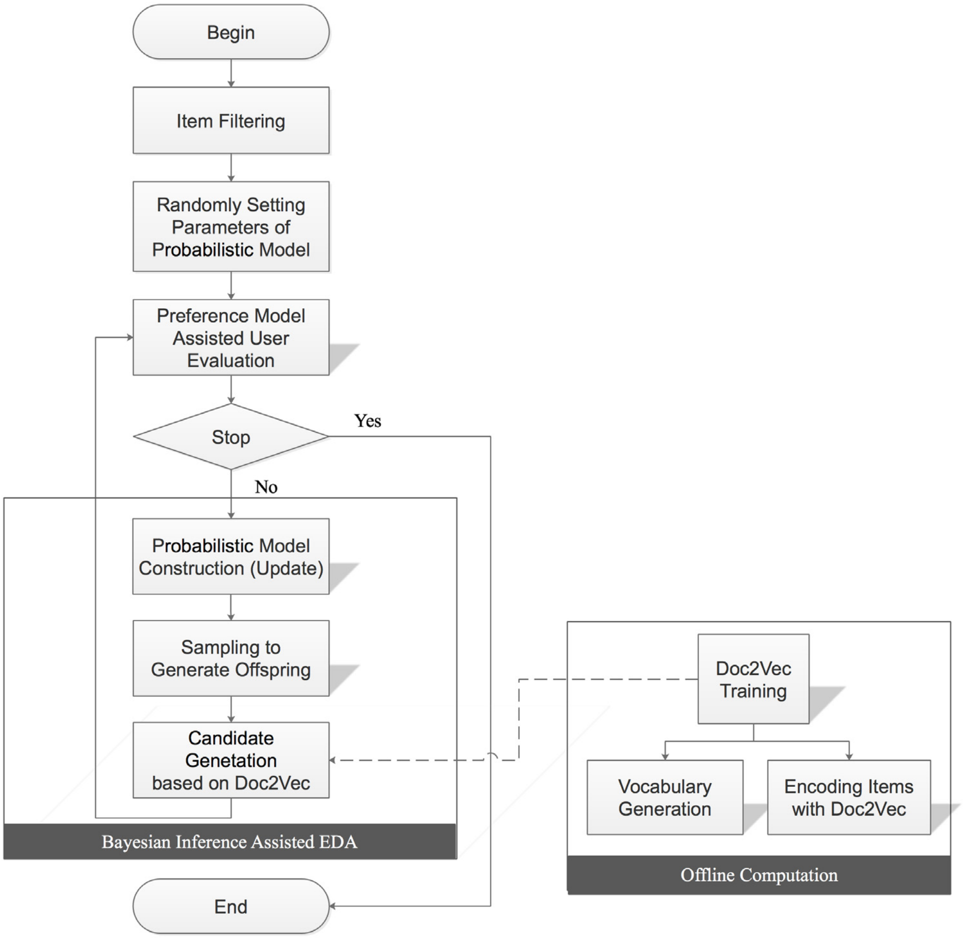

Begin
Item Filtering
Randomly Setting Parameters of Probabilistic Model
Preference Model Assisted User Evaluation
Stop
Yes
No
Probabilistic Model Construction (Update)
Sampling to Generate Offspring
Candidate Genetation based on Doc2Vec
Bayesian Inference Assisted EDA
Doc2Vec Training
Vocabulary Generation
Encoding Items with Doc2Vec
Offline Computation
End

Fig. 1. The framework of LMIEDA.

the preference model is updated to track the user's preference along with the evolution. The most natural interactions of a user are clicking, browsing and saving; accordingly, the preference model is constructed based on these actions.

The main function of the preference model is to quantify the interest of the user on interacted items. Each item i and its corresponding document description $\boldsymbol{d}_i$ together with the word frequency $\boldsymbol{m}_i$ are linked; items that physically exist in merchants' warehouses are only index i in our formulas. In search, these evaluated items paired with user preference are denoted as $T_t = \{(\boldsymbol{m}_i, \hat{f}_i), i = 1, 2, \ldots, M_t\}, t = 1, 2, \ldots$, where $\boldsymbol{m}_i$ is the candidate (word frequency), and its associated item has been evaluated by the user. $\hat{f}_i$ is its associated preference, and t indexes iterations. Within an iteration, we can get $\boldsymbol{m}_i$ easily by collecting all candidates that the user has clicked, browsed, or saved, but $\hat{f}_i$ can only be estimated based on interactions.

Here, the interactive time associated with user interactions is delivered to quantify the user preference and get the $\hat{f}_i$ for the candidate $\boldsymbol{m}_i$. In [33], we have defined the browsing time of three typical interactions, i.e. click-browse-save (**Interaction 1**), click-browse-close (**Interaction 2**), and non-click (**Interaction 3**), and they are denoted as $t_C(\boldsymbol{m}_i)$, $t_S(\boldsymbol{m}_i)$, and $t_{NC}(\boldsymbol{m}_i)$, respectively.

$$\hat{f}\,(\boldsymbol{m}_i) = \begin{cases} rand\left[1 - \alpha \cdot e^{-\frac{t_S(\boldsymbol{m}_i)}{\delta_S}}, 1\right], \text{ if } \boldsymbol{m}_i \text{ gets } \textbf{Interaction1} \\ rand\left[\chi - \beta \cdot e^{-2\cdot\frac{t_C(\boldsymbol{m}_i)}{\delta_S}}, \chi + \beta \cdot (1 - e^{-\frac{t_C(\boldsymbol{m}_i)}{\delta_S}})\right. \\ \left.\cdot e^{-\frac{t_C(\boldsymbol{m}_i)}{\delta_S}}\right), \text{ if } \boldsymbol{m}_i \text{ gets } \textbf{Interaction2} \\ rand\left[0, \alpha \cdot e^{-\frac{t_{NC}(\boldsymbol{m}_i)}{\delta_S}}\right), \text{ if } \boldsymbol{m}_i \text{ gets } \textbf{Interaction3} \end{cases} \quad (3)$$

The value of $\hat{f}$ is sampled from an interval function, as expressed in Eq. (3), which has been designed to depict evaluation uncertainties [33]. In the equation, δ_s is a time scale parameter, and $rand\,[a, b]$ represents a uniform sampling in the interval $[a, b]$.

The sampling step of EDA is then introduced. As mentioned, the personalized search problem has been converted into a document-match problem with Doc2Vec. So, the *document*, which carries both its own description and social comments, deserves careful treatment. Its corresponding word-frequency representation $\boldsymbol{m}$ follows the multinomial distribution with parameter $\boldsymbol{\mu}$ as shown in Eq. (4):

$$\boldsymbol{m} \sim \text{Mult}\left(m_1, m_2, \ldots, m_{|V|} \,|\boldsymbol{\mu}\ , N\right) \tag{4}$$

where N is the number of the words in the *document*, i.e., $N = \sum_{i=1}^{|V|} m_i$. Its parameter $\boldsymbol{\mu}$ is subjected to the following Dirichlet distribution with a hyper-parameter $\boldsymbol{\alpha}$:

$$\boldsymbol{\mu} \sim \text{Dir}\left(\boldsymbol{\mu}\,|\boldsymbol{\alpha}\right) \tag{5}$$

Given a value of $\boldsymbol{\alpha}$, values of $\boldsymbol{\mu}$ are sampled from the Dirichlet distribution. Then, word frequencies can be generated by substituting these sampled $\boldsymbol{\mu}$ values into Eq. (4). To be specific, M_μ values of $\boldsymbol{\mu}$ will be sampled from the Dirichlet distribution, and M_μ corresponding multinomial distributions are thus obtained by substituting each sampled value of $\boldsymbol{\mu}$ into Eq. (4). Last, M_v individuals can be obtained through sampling each multinomial distribution. Here, an individual indicates the word frequencies in its document. The population size is $M = M_\mu \cdot M_v$.

In the initialization, the value of $\boldsymbol{\alpha}$ is randomly given since we have no prior preference for it. The values of $\boldsymbol{\mu}$ as $\left(p_1, p_2, \ldots, p_{|V|}\right)$ are sampled from Eq. (5), and it is used to generate the individual $\boldsymbol{m}$ whose values are the word frequencies of words from $\boldsymbol{V}$. The initial population of M individuals is obtained by repeating the sampling above.

In our target scenario, the algorithm addresses personalized search problems aiming to shorten the time spent in locating user-preferred items. The adopted distribution is a description of the word frequencies of the document corresponding to a user-preferred candidate item. By sampling the updated distribution, several individuals, i.e., word frequencies are obtained. However, users are unable to understand them. To humans, $\boldsymbol{m}$ is meaningless for evaluation. We have to locate the items that are most similar to these individuals, which can be described with the similarity between the associated semantic vectors. Based on the word frequency, we first take a certain number of words from vocabulary to form the "document" corresponding to each individual and adopt Doc2Vec to convert them into vectors, i.e., Doc2Vec($\boldsymbol{m}$). By comparing Cosine similarity, we, therefore, locate these most-similar items that have not been evaluated for user evaluation. The M candidates that are most similar to the generated individuals in vector space are selected and shown to the user for further interactive evaluation. There exists a notation discrepancy between here and $\boldsymbol{x}$ = Doc2Vec(*document*) discussed in Section 3.1 since these individuals that are generated through sampling correspond to word combinations ignoring ordering.

The update of the probabilistic model is then detailed. The offspring generation of EDA relies heavily on the Dirichlet-Multinomial compound distribution, i.e. counts of the words included in a document that represents an item follow a Multinomial distribution whose parameter subjects to a Dirichlet distribution. The posterior distribution for $\boldsymbol{\mu}$, i.e. $p(\boldsymbol{\mu}|D, \boldsymbol{\alpha})$ is obtained through multiplying the prior $p(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha})$ by the likelihood function $p(D|\boldsymbol{\mu}) = \text{Mult}(\boldsymbol{m}|\boldsymbol{\mu}, N)$. It goes in the form

$$p(\boldsymbol{\mu}|D, \boldsymbol{\alpha}) \propto p(D|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{\alpha}) \propto \prod_{v=1}^{|V|} \mu_v^{\alpha_v + D_v - 1} \tag{6}$$

where $|V|$ is the size of vocabulary, and more details can be found in [62].

The steps regarding the parameter update of the compound distribution are explained from the viewpoint of MCMC (Markov-chain Monte-Carlo); also, Gelman–Rubin convergence diagnostic is adopted. $\boldsymbol{\alpha}$ is omitted when no a priori knowledge is available to introduce or to emphasize. For offering readers a better understanding, Eq. (7) presents a high-level description of algorithmic steps.

$$\begin{aligned} &p\left(\boldsymbol{\mu}_{t-1}|D_{t-1}\right) \rightarrow \left\{\boldsymbol{\mu}_{t-1,s}\right\} \rightarrow p\left(\boldsymbol{m}_t|\boldsymbol{\mu}_{t-1}\right) \rightarrow \left\{\boldsymbol{m}_{t,s}\right\} \\ &\quad \rightarrow \left\{D_{t,s}\right\} \rightarrow p\left(\boldsymbol{\mu}_t|D_t\right) \end{aligned} \tag{7}$$

where D_{t-1} and D_t are the observations before and after the update, and similarly, $\boldsymbol{\mu}_t$ and $\boldsymbol{\mu}_{t-1}$ are the posterior and prior to the parameter. Using D_{t-1} in iteration $t-1$, we obtain a current estimate of the probability density of $\boldsymbol{\mu}_{t-1}$, denoted in Eq. (7) as $p\left(\boldsymbol{\mu}_{t-1}|D_{t-1}\right)$. With this current distribution estimate, we first sample many values of $\boldsymbol{\mu}_{t-1}$ using Eq. (5), use those samples in turn to sample candidate values of $\boldsymbol{m}_t$ using (4), and use those sampled vectors to generate a new set of documents. Through semantic matching, the same number of items corresponding to these documents are located for the user to evaluate. Then, the new aggregated vector D_t are calculated with all available user-evaluated samples according to Eq. (8), which is detailed in the following part of this section. With the vector, we can re-estimate the new probability density denoted in Eq. (7) as $p\left(\boldsymbol{\mu}_t|D_t\right)$. Then, the iterative process goes on to step $t+1$.

In our IEDA, the compound distribution is essentially determined by the parameter $\boldsymbol{\mu}$, and it can be calculated with Eq. (7) when parameter $\boldsymbol{\alpha}$ is given. Only when the observations are obtained can the value of $\boldsymbol{\mu}_t$ be determined by sampling the estimated posterior Dirichlet distribution.

For obtaining the observation D_t, the samples with higher f_i from $T_{1:t} = \{T_1, \ldots, T_t\}$ are first selected as $\{(\boldsymbol{m}_i, f_i), i = 1, 2, \ldots, N_T\}$, and then they are aggregated with a weighted sum to get the D_t:

$$D_t = \llcorner \frac{\sum_{i=1}^{N_T} f_i \cdot \boldsymbol{m}_i}{\sum_{i=1}^{N_T} f_i} \lrcorner \tag{8}$$

Here, D_t can reflect the preference of the user. The observation of newly evaluated individuals is expected to mirror the current user's preference since our purpose is to obtain the most satisfactory solution in a short time with the guidance of $\boldsymbol{\mu}_t$. $\llcorner \cdot \lrcorner$ stands for the rounding operation, which takes the nearest integer values of input.

With D_t and Eq. (7), the new posterior distribution $p\left(\boldsymbol{\mu}_t\,|D_t\right)$ can be updated; it is viewed as the prior distribution at the next iteration. Then, we can generate offspring by sampling the updated posterior distribution.

To be clear, another model for multinomial data is defined by integrating out the μ parameter, obtaining

$$p(\boldsymbol{m}|\boldsymbol{\alpha}) = \int p(\boldsymbol{m}|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{\alpha})d\boldsymbol{\mu} \propto \frac{B(\boldsymbol{\alpha}+\boldsymbol{m})}{B(\boldsymbol{\alpha})} \tag{9}$$

where $B(\cdot)$ is the multinomial Beta function that normalizes the Dirichlet.

4. Experiments and analyses

4.1. Experimental setting

Comparisons on the personalized search of two different fields, i.e., movies and books, among the proposed algorithm and other IECs are conducted to prove its effectiveness and efficiency. Movies and books, which are commonly described with text, are chosen as the search target because they cannot be well modeled with the key–value pattern. The data for movies and books (updated in March 2018) are from imdb.com and Douban.com. IMDb is an online database of information related to films, TV programs, and internet streams, including cast, plot summaries, and fan reviews and ratings. Douban is a Chinese social networking service website that allows its users to rate, make comments, and create content related to books, music, and films.

The experimental platform is developed with Python 3.6, Qt 5.9, and MongoDB 3.4. Python is used to construct the algorithm; Qt deals with GUI; MongoDB manages data.

The interface used in our previous study [33], which implements the interactive logic mentioned in Section 3.C and Eq. (3), is adopted. It is made up of four areas that are separately designed for the user to input inquiries, to browse brief introductions of displayed items, to obtain details of and interact with some of the interesting ones among them, and to save his/her favorites.

In addition, similar to [33], to make experimental results objective and convincing, objective experiments without users are needed. So, it is necessary to find a substitute for human users to evaluate items. The method should approximate user preference as accurately as possible. To achieve this, a graphical user interface (GUI) shown in Fig. 2 is developed to extract user preferences through the process of users' ranking all candidate items in the form of pair comparisons.

The interface consists of three areas, including the brief display (area 1), the detail display (area 2), and interactive buttons (area 3). The first user reads brief information about the pair for evaluation. For anyone that wants to learn more by clicking the button "showInDetail", details are shown in area 2. Then, she/he decides the preference between them. The pair of items to be evaluated on the screen is selected following the flow of Quicksort. To avoid being affected by user fatigue, the button "LoadListFromFile" and the button "SaveListToFile" (area 3) are added to allow the user to stop for a rest.

Two groups of experiments were conducted to demonstrate the merits of the proposed algorithm from different perspectives. One is to objectively evaluate the effectiveness of extracted user preferences; the other is to test its practical performance in the real scene involving users.

Four compared algorithms, i.e. interactive genetic algorithm (IGA), personalized search inspired fast interactive estimation of distribution algorithm (PSIEDADK), the support vector machine classification (SVMC), and the logistic regression classification (LRC) based personalized search are chosen. For IGA, three versions with different encoding schemes are presented to show the efficiency of the proposed evolutionary mechanism and encoding, among which one has the same encoding as the proposed algorithm. As for the PSIEDADK and two classification-based personalized search algorithms, they are chosen to demonstrate the performance.

The termination criterion is up to the users. Some users have an expected item, and the search is terminated either when the item is obtained or the user feels too fatigued to carry on.

Similar to authors' previous research [33], four indicators: (1) the number of evaluated solutions, (2) the hit rate, (3) the discounted cumulative cost (DCC), and (4) the search time are adopted. Indicators (1) and (2) are used in parameter and objective experiments without real users. In those involving users, the search time is recorded for measuring the efficiency of algorithms. DCC suits both groups. More evaluated items and longer search time indicate a heavier burden on users, i.e., higher DCC [33].

30 runs of parameter/objective experiments and 20 runs of those involving users are conducted for the statistical analyses.

4.2. Parameters setting

Taking the movie case as an example, the experimental setting including variables, the encoding strategy, and parameters are discussed in detail.

Table 2
Attributes adopted by key–value decimal encoding.

Attribute name	Number of values
Country	156
Genres	28
Rating	10
Votes	187
Year	14

Table 3
Example of a movie in decimal encoding.

Attribute	Country	Genres	Rating	Votes	Year
Phenotype	us	Drama	8	4638	1989
Decimal encoding string	0	5	7	3	10

Table 4
Example of a book in term-based decimal encoding.

Word	···	1980	Time	Gay	Epidemic	Disease	···
Word frequency	···	2	19	6	12	9	···

4.2.1. Optimization variables

In the key–value pattern, variables are specified attributes, with which searched items are expected to be modeled in limited dimensions. Therefore, a combination of attribute values is a solution. In this case, five attributes of movies together with the number of values are listed in Table 2.

4.2.2. Encoding strategy

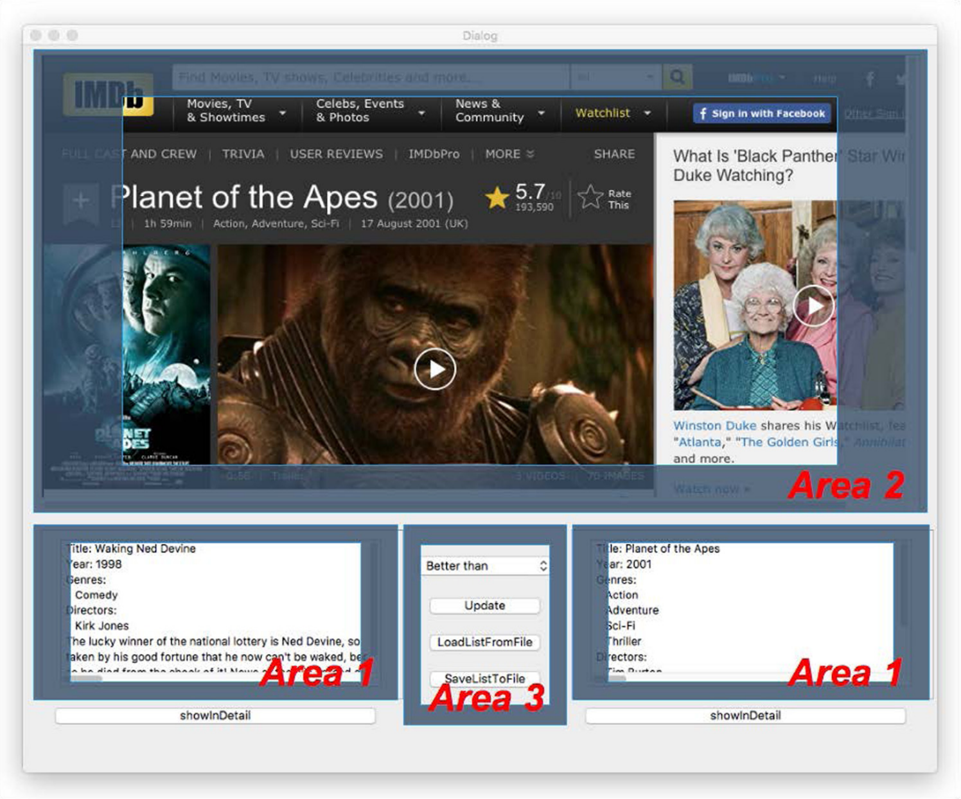
Three different encoding schemes are selected for algorithms. Specifically, they are the key–value decimal encoding (KVDE) made up of searched item attributes, the term-based decimal encoding (TDE) consisting of word frequencies, and the semantic vector real encoding (SVRE) inherited from Doc2Vec. The examples of KVDE and TDE are separately given in Tables 3 and 4. As mentioned, model Doc2Vec is trained to map documents into fixed-length vectors; so, SVRE encodes solutions into fixed-length real strings.

Training parameters of Doc2Vec are set as min_count=2, size=300, workers=8, etc. The min_count indicates that all words with a total frequency lower than this are ignored. The size is the vector dimensionality. Most parameters follow the default setting of Gensim [63].

4.2.3. Parameters

The operator parameters of IGAs with KVDE, TDE, and SVRE are carefully tuned. For IGA-KVDE, the parameters are set as roulette wheel selection, intermediate crossover with the probability being 0.99, and Gaussian mutation with the probability as 0.10; the average and the variation of the Gaussian function are 0 and 0.6, respectively. For IGA-TDE and SVRE, similar settings are adopted except for the crossover (set as k-Point Crossover with k=3) and the variation (set as 0.8).

As for two classification-based methods, LRC-IPSA and SVMC-IPSA, they take SVRE as their encoding strategies. Their parameters are set as: the K-fold cross-validation with K being 3 is used. For the LRC-IPSA, the solver with coordinate descent algorithm comes from LIBLINEAR, and the norm used in the penalization is L^2. For the SVMC-IPSA, the maximal iteration is 160, the Gaussian function is used as the kernel function, and the penalty parameter C of the error term is 1. To PSIEDADK-KVDE, reduction control parameter ε is set as 0.6. As for these aforementioned hyper-parameters, K-fold Cross-Validation assisted tuning has been conducted for guaranteeing the performance of compared algorithms; thus, a convincing conclusion can be drawn.


Dialog
IMDb
Find Movies, TV shows, Celebrities and more...
Movies, TV & Showtimes
Celebs, Events & Photos
News & Community
Watchlist
Sign in with Facebook
FULL CAST AND CREW | TRIVIA | USER REVIEWS | IMDbPro | MORE
SHARE
Planet of the Apes (2001)
5.7/10
193,590
Rate This
1h 59min | Action, Adventure, Sci-Fi | 17 August 2001 (UK)
What Is 'Black Panther' Star Winston Duke Watching?
Winston Duke shares his Watchlist, featuring "Atlanta," "The Golden Girls," Annihilation and more.
Watch now »
Area 2
Better than
Update
LoadListFromFile
SaveListToFile
Area 3
Area 1
Area 1
showInDetail
showInDetail

Title: Waking Ned Devine
Year: 1998
Genres:
Comedy
Directors:
Kirk Jones
The lucky winner of the national lottery is Ned Devine, so taken by his good fortune that he now can't be waked, be

Title: Planet of the Apes
Year: 2001
Genres:
Action
Adventure
Sci-Fi
Thriller
Directors:

Fig. 2. Interface for user preference extraction.

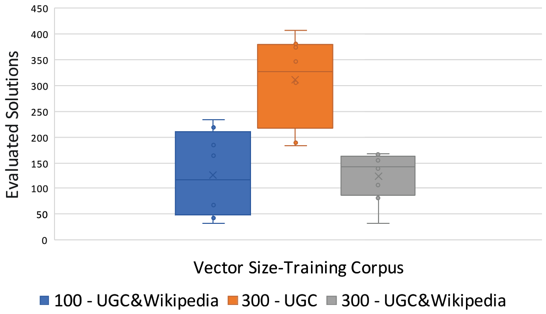


Fig. 3. Performance vs. vector size and training corpus.

4.3. Experiments on parameters

First, two experiments are conducted for determining the values of the critical parameters: vector size, training corpus, vocabulary size $|V|$, and the generated document size $|doc|$. Their influences on the proposed algorithm are conveyed, and the number of evaluated items with a lower value indicates a better performance.

4.3.1. Influence of vector size and training corpus on doc2vec

Eight extracted user preferences on filtered candidate items (movies) are chosen for plotting the box-whisker plot, which shows the changes in the number of evaluated items along with different combinations of vector size and training corpus. As presented in Fig. 3, three plots correspond to different combinations. UGC is short for user-generated content; UGC&Wikipedia indicates that both training corpora are utilized.

The following conclusions can be drawn. (1) Good performance of Dov2vec relies on an appropriate combination of vector size and training corpus. (2) Language model training does require a large corpus as suggested in [19], and Wikipedia corpus is a good choice as a supplement.

4.3.2. Influence of $|V|$ and $|doc|$ on search burden

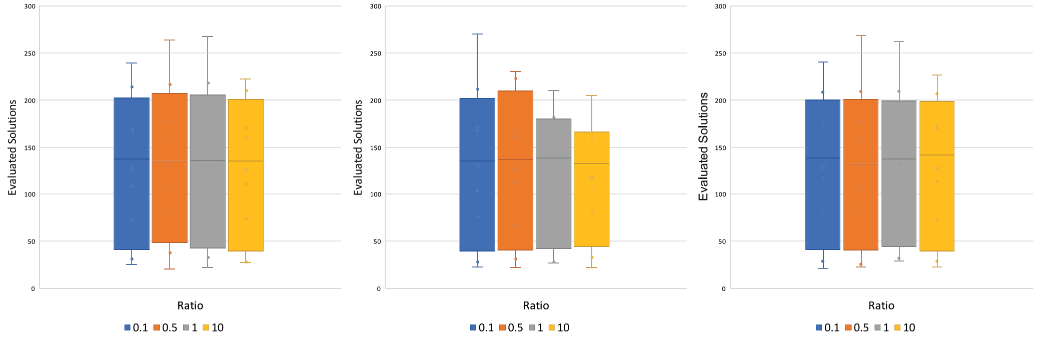
Similar to the aforementioned experiment, the same expected items are set. Given the boxplot of the number of evaluated items along with the values of $|V|$ and $|doc|$ in Fig. 4. The influences of them on the search burden are as follows: (1) It is difficult to draw the conclusion that there are significant differences among different values (of $|V|$ and $|doc|$). Accordingly, there is no necessity to maintain documents with a large $|doc|$ for saving computational cost. (2) The proposed algorithm is robust to the parameters. (3) Generally, algorithms with higher ratios (of $|doc|$ to $|V|$) perform slightly more stable.

4.4. Objective experiments with extracted user preference

Objective experiments are conducted with the extracted user preference model instead of real users. The hit rate is adopted for measuring the performance of all the compared algorithms and is a conventional metric in measuring business performance based on sales. It is redefined as the ratio between the number of all executed runs and the number of acceptable runs. As suggested in [5], the runs with iterations less than 20, i.e., 240 evaluated items (twelve items for user evaluation on each webpage), are regarded as acceptable.

Five compared algorithms including IGA-KVDE, IGA-TDE, IGA-SVRE, PSIEDADK-KVDE, LRC-IPSA-SVRE, and SVMC-IPSA-SVRE are adopted.

To analyze the performance of compared algorithms, sixteen items with different rankings, including eight movies and eight books, are selected as search targets. Average hit rates of 30 runs are listed in Table 5. Their first columns indicate the selling orders and numbers of filtered results of those items.



(a) $|V|=300$ (b) $|V|=600$ (c) $|V|=3000$

Fig. 4. Performance vs. the size of vocabulary and generated documents.

Table 5
Hit rates for sixteen expected solutions.

Ranking	LMIEDA-TDE	IGA-KVDE	IGA-TDE	IGA-SVRE	PSIEDADK-KVDE	L-SVRE	S-SVRE	Group
67/598	97%	67%	**100%**	37%	60%	50%	47%	*Movie*
70/469	**100%**	97%	97%	60%	60%	60%	67%	*Movie*
76/465	**100%**	77%	3%	43%	47%	60%	43%	*Movie*
102/421	**100%**	80%	**100%**	53%	53%	47%	53%	*Movie*
160/546	**100%**	80%	**100%**	37%	43%	40%	43%	*Movie*
262/521	**100%**	0%	**100%**	50%	57%	53%	43%	*Movie*
335/448	**100%**	30%	**100%**	77%	60%	50%	47%	*Movie*
355/712	**100%**	87%	**100%**	60%	50%	43%	60%	*Movie*
48/532	**100%**	38%	72%	52%	53%	56%	36%	*Book*
65/918	**100%**	52%	**100%**	22%	36%	20%	20%	*Book*
177/918	**100%**	44%	**100%**	20%	20%	16%	26%	*Book*
298/313	**100%**	46%	**100%**	50%	47%	36%	48%	*Book*
299/313	**100%**	38%	**100%**	52%	60%	56%	36%	*Book*
310/313	**100%**	32%	**100%**	46%	50%	44%	42%	*Book*
395/532	**100%**	32%	52%	46%	50%	44%	42%	*Book*
760/918	**100%**	56%	**100%**	18%	36%	30%	20%	*Book*

Given Table 5, the conclusions are as follows: (1) The proposed algorithm significantly outperforms all compared ones on most computation cases. (2) Comparing encoding schemes, TDE outperforms the others when IGAs employing different encoding methods come into focus. (3) IECs perform better than machine-learning-based methods.

Good performance relies on both effective optimization modes and encoding methods. TDE helps to carry more information, including social intelligence, and IEDA, utilizing a Dirichlet-Multinomial compound distribution updated by Bayesian inference, offers a compatible processing mechanism, which guarantees the superiority of the designed algorithm.

4.5. Comparative experiments involving real users

To demonstrate the performance in alleviating user fatigue and accelerating search, users are involved. With similar settings to those of Section 4-D, comparisons are conducted among LMIEDA-TDE, IGA-TDE, and LRC-IPSA-SVRE. Search time and DCC for a search of the same sixteen targets are recorded for statistical analyses.

To guarantee the validity of the experiments, runs are reduced from 30 to 20, which is expected to ease side effects caused by the heavy user burden. Those entries with the label † are significantly worse than those of the proposed one with a 0.95 confidence level. As in Section 4.4, those with 240 evaluated items are regarded as failed searches, which means users give up if they cannot obtain the preferred one after evaluating 240 items. Furthermore, those with hit rates less than 60% are regarded as having too many failed searches for the statistical analyses and are marked with the label ‡.

As shown in Tables 6 and 7, similar conclusions can be drawn as those from the objective experiments. Compared with other algorithms, the proposed one is of higher efficiency and better stability, and it effectively reduces user fatigue.

Comparing experimental results between this and the authors' previous work [33], the new method of extracting user preference by pair comparisons works better.

To summarize: (1) The language model that helps to make the problem conversion offers further LMIEDA information through encoding. (2) IEDA, involving the Dirichlet-Multinomial compound distribution as its probabilistic model, performs better than other optimization mechanisms when addressing word-described personalized search. (3) The Bayesian inference and the EDA based evolutionary operator are beneficial for reducing user fatigue, tracking user preference, and guiding the direction of evolution, which has been proven by the computation cases.

5. Conclusions

For solving word/document-described problems that cannot be well encoded with the structural numerical methods, LMIEDA is proposed by integrating the mixture of unigrams, LDA, and Doc2Vec into the EDA framework. From the viewpoint of optimization, language model based encoding, a novel preference expression by use of Dirichlet-Multinomial compound distribution, and a Bayesian inference-enhanced interactive version of EDA (estimation of the distribution algorithm) have been studied to effectively optimize the word/document-described problems. Doc2Vec is first used to encode candidates, and then the search is changed into a dynamic document matching problem. To solve it, a Dirichlet-Multinomial compound distribution assisted with

Table 6
Mean and standard deviation of search time (s) in subjective comparative experiments with sixteen expected solutions.

Ranking	LMIEDA-TDE	IGA-TDE	L-SVRE	Group
67/598	**965.62(293.37)**	1037.89(382.48)	*50%**	*Movie*
70/469	1646.05(345.78)	**1498.3(372.7)**	2339.01(946.87)	*Movie*
76/465	**1334.74(366.93)**	1402.4(396.02)	2561.43(1315.74)**	*Movie*
102/421	**318.67(93.19)**	353.67(146.49)	*47%**	*Movie*
160/546	**1532.23(457.23)**	1459.51(322.29)	*40%**	*Movie*
262/521	**901.38(286.47)**	1297.47(414.59)**	*53%**	*Movie*
335/448	**897.93(347.89)**	1251.66(441.76)**	*50%**	*Movie*
355/712	**1420.82(387.49)**	1824.61(549.19**	*43%**	*Movie*
48/532	**1559.43(276.49)**	2154.52(404.3)**	*55%**	*Book*
65/918	**462.25(129.71)**	483.17(185.95)	*25%**	*Book*
177/918	**277.23(70.93)**	350.1(140.16)**	*30%**	*Book*
299/313	**137.53(31.66)**	157.86(41.48)**	1055.93(580.4)**	*Book*
298/313	**358.54(71.7)**	590.32(207.44)**	1272.28(687.65)**	*Book*
310/313	681.32(139.72)	869.49(218.77)	**536.34(398.24)**	*Book*
395/532	**1614.16(280.43)**	2265.32(593.73)**	*50%**	*Book*
760/918	**413.51(98.06)**	482.24(181.21)**	*30%**	*Book*

*Searches are marked when hit rates are less than 60%.

**Searches are marked when they pass the Welch's t-test with a 0.95 confidence level.

Table 7
Mean and standard deviation of DCC in subjective comparative experiments with sixteen expected solutions.

Ranking	LMIEDA-TDE	IGA-TDE	L-SVRE	Group
48/532	**78.22(7.2)**	91.05(9.33)**	*55%**	*Movie*
65/918	32.37(5.72)	**30.93(7.18)**	*25%	*Movie*
177/918	**24.11(3.07)**	26.97(7.09)	*30%**	*Movie*
298/313	**26.68(3.09)**	38.32(8.64)**	57.25(26.84)**	*Movie*
299/313	20.81(2.5)	**18.07(3.12)**	48.28(23.35)	*Movie*
310/313	42.21(5.4)	48.97(8.24)	**34.6(18.35)**	*Movie*
395/532	**76.11(8.94)**	93.51(16.3)**	*50%**	*Movie*
760/918	**28.77(4.7)**	28.78(8.42)	*30%**	*Movie*
67/598	**46.91(10.17)**	47.08(15.24)	*50%**	*Book*
70/469	65.96(10.4)	**61.42(12.23)**	73.24(40.3)	*Book*
76/465	60.47(12.58)	**60.07(14.81)**	84.68(40.92)	*Book*
102/421	**16.15(4.21)**	16.8(6.52)	*47%**	*Book*
160/546	64.6(16.46)	**60.94(12.16)**	*40%**	*Book*
262/521	**42.98(10.7)**	55.07(14.53)**	*53%**	*Book*
335/448	**44.84(13.12)**	57.68(15.56)**	*50%**	*Book*
355/712	**64.42(13.14)**	76.4(16.02)**	*43%**	*Book*

*Searches are marked with hit rates less than 60%.

**Searches are marked when they pass the Welch's t-test with a 0.95 confidence level.

Bayesian inference is applied to express and track diverse user preference. According to the encoding and preference model, an interactive estimation of distribution algorithm with a Bayesian-inference-based probabilistic model is further developed for generating offspring individuals. The proposed algorithm is applied to two different kinds of personalized search problems, and the results experimentally demonstrate that our algorithm outperforms all compared ones in alleviating user fatigue and speeding up the search, indicating that the presented LMIEDA is competitive for improving the word-described personalized search in E-commerce.

In the future, articulating content-based and collaborative-based personalized searches with the LMIEDA will be studied to improve the performance of the algorithm in word/document-described problems.

CRediT authorship contribution statement

Yang Chen: Conceptualization, Methodology, Software, Writing - original draft. **Yaochu Jin:** Conceptualization, Supervision, Validation, Writing - review & editing. **Xiaoyan Sun:** Conceptualization, Supervision, Software, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the National Natural Science Foundation of China with Grant No. 61876184 and 61473298.

References

[1] Y. Han, D. Gong, Y. Jin, Q. Pan, Evolutionary multi-objective blocking lot-streaming flow shop scheduling with machine breakdowns, IEEE Trans. Cybern. 49 (2019) 184–197.

[2] D. Gong, Y. Han, J. Sun, A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems, Knowl.-Based Syst. 148 (2018) 115–130.

[3] X. Sun, Y. Chen, Y. Liu, D. Gong, Indicator-based set evolution particle swarm optimization for many-objective problems, Soft Comput. 20 (6) (2016) 2219–2232.

[4] Y. Chen, X. Sun, Y. Hu, Federated learning assisted interactive eda with dual probabilistic models for personalized search, in: International Conference on Swarm Intelligence, Springer, 2019, pp. 374–383.

[5] H. Takagi, Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation, Proc. IEEE 89 (9) (2001) 1275–1296.

[6] Y. Chen, X. Sun, D. Gong, X. Yao, DPM-IEDA: Dual probabilistic model assisted interactive estimation of distribution algorithm for personalized search, IEEE Access 7 (2019) 41006–41016.

[7] M. Fukumoto, S. Koga, A proposal for user's intervention in interactive evolutionary computation for optimizing fragrance composition, Commun. Comput. Inf. Sci. 434 (Part I) (2014) 85–89.

[8] A. Oliver, O. Regragui, N. Monmarch, G. Venturini, Genetic and interactive optimization of web sites, in: Proceedings International WWW Conference, vol. 1, 2002, p. 4.

[9] X. Sun, Y. Lu, D. Gong, K. Zhang, Interactive genetic algorithm with CP-nets preference surrogate and application in personalized search, Control Decis. 30 (7) (2015) 1153–1161.

[10] D. Gong, J. Ren, X. Sun, Neural network surrogate models of interactive genetic algorithms with individual's interval fitness, Control Decis. 24 (10) (2009) 1522–1530.

[11] D. Gong, X. Sun, J. Yuan, Interactive genetic algorithms with individual's uncertain fitness, Chinese J. Electron. 4 (October) (2009) 619–624.

[12] Y. Li, Adaptive learning evaluation model for evolutionary art, in: 2012 Ieee Congress on Evolutionary Computation (CEC), 2012, pp. 10–15.

[13] R. Kamalian, E. Yeh, Y. Zhang, A.M. Agogino, H. Takagi, Reducing human fatigue in interactive evolutionary computation through fuzzy systems and machine learning systems, in: IEEE International Conference on Fuzzy Systems, 2006, pp. 678–684.

[14] T. Chugh, K. Sindhya, J. Hakanen, K. Miettinen, An interactive simple indicator-based evolutionary algorithm (I-SIBEA) for multiobjective optimization problems, Lecture Notes in Comput. Sci. 9018 (2015) 277–291.

[15] M. Luque, K. Miettinen, R. Saborido, A.B. Ruiz, An interactive evolutionary multiobjective optimization method based on the WASF-GA algorithm, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 249–263.

[16] X. Sun, L. Zhu, L. Bao, L. Liu, X. Nie, Interactive genetic algorithm with group intelligence articulated possibilistic condition preference model, in: Asia-Pacific Conference on Simulated Evolution and Learning, Springer, 2017, pp. 158–169.

[17] H. Wang, S. Shao, X. Zhou, C. Wan, A. Bouguettaya, Preference recommendation for personalized search, Knowl.-Based Syst. 100 (2016) 124–136.

[18] T. Mikolov, K. Chen, G.S. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, pp. 1–12, arXiv preprint, cs.CL.

[19] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, 2014, arXiv preprint arXiv:1405.4053.

[20] Y. Sun, W. Liu, R. Qiu, C. Huang, Research development of user interest modeling in China, J. Intell. 32 (5) (2013) 145–149.

[21] W. Guoxia, L. Heping, Survey of personalized recommendation systems, Comput. Eng. Appl. 48 (7) (2012) 66–76.

[22] N. Capuano, F. Chiclana, E. Herrera-Viedma, H. Fujita, V. Loia, Fuzzy rankings for preferences modeling in group decision making, Int. J. Intell. Syst. 33 (7) (2018) 1555–1570.

[23] X. Tian, Z. Xu, H. Fujita, Sequential funding the venture project or not? a prospect consensus process with probabilistic hesitant fuzzy preference information, Knowl.-Based Syst. 161 (2018) 172–184.

[24] H. Liao, G. Si, Z. Xu, H. Fujita, Hesitant fuzzy linguistic preference utility set and its application in selection of fire rescue plans, Int. J. Environ. Res. Public Health 15 (4) (2018) 664.

[25] O. Kassak, M. Kompan, M. Bielikova, User preference modeling by global and individual weights for personalized recommendation, Acta Polytech. Hung. 12 (8) (2015) 27–41.

[26] X. Tang, J. Zhou, Dynamic personalized recommendation on sparse data, IEEE Trans. Knowl. Data Eng. 25 (12) (2013) 2895–2899.

[27] H. Fujita, A. Gaeta, V. Loia, F. Orciuoli, Improving awareness in early stages of security analysis: A zone partition method based on GrC, Appl. Intell. 49 (3) (2019) 1063–1077.

[28] M. Abou-Zleikha, N. Shaker, Evolving random forest for preference learning, Lecture Notes in Comput. Sci. 9028 (2015) 318–330.

[29] M. Abou-Zleikha, N. Shaker, M.G. Christensen, Preference learning with evolutionary Multivariate Adaptive Regression Spline model, in: 2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings, 2015, pp. 2184–2191.

[30] H.J. Ahn, Evaluating customer aid functions of online stores with agent-based models of customer behavior and evolution strategy, Inform. Sci. 180 (9) (2010) 1555–1570.

[31] H.H. Kim, E. Kim, J.J. Lee, C. Ahn, A recommender system based on genetic algorithm for music data, Comput. Eng. 6 (2010) 414–417.

[32] V. Kant, K.K. Bharadwaj, A user-oriented content based recommender system based on reclusive methods and interactive genetic algorithm, in: Advances in Intelligent Systems and Computing, in: 201 AISC, vol. 1, Springer, 2013, pp. 543–554.

[33] Y. Chen, X. Sun, D. Gong, Y. Zhang, J. Choi, S. Klasky, Personalized search inspired fast interactive estimation of distribution algorithm and its application, IEEE Trans. Evol. Comput. 21 (4) (2017) 588–600.

[34] H. Xie, X. Li, T. Wang, R.Y. Lau, T.L. Wong, L. Chen, F.L. Wang, Q. Li, Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy, Inf. Process. Manage. 52 (1) (2016) 61–72, [Online]. Available: http://dx.doi.org/10.1016/j.ipm.2015.03.001.

[35] H. Mühlenbein, G. Paass, From recombination of genes to the estimation of distributions I. Binary parameters, in: Parallel Problem Solving from Nature—PPSN IV, Springer, 1996, pp. 178–187.

[36] H. Mühlenbein, J. Bendisch, H.-M. Voigt, From recombination of genes to the estimation of distributions II. Continuous parameters, in: Parallel Problem Solving from Nature—PPSN IV, Springer, 1996, pp. 188–197.

[37] M. Pelikan, M.W. Hauschild, F.G. Lobo, Estimation of distribution algorithms, in: Springer Handbook of Computational Intelligence, Springer Berlin Heidelberg, 2015, pp. 889–928.

[38] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Trans. Evol. Comput. 3 (4) (1999) 287–297.

[39] M. Pelikan, D.E. Goldberg, E. Cantu-Paz, Linkage problem, distribution estimation, and Bayesian networks, Evol. Comput. 8 (3) (2000) 311–340.

[40] P.J.A.L. Larrañaga, A review on estimation of distribution algorithms, in estimation of distribution algorithms: A new tool for evolutionary computation, Springer Sci. Bus. Media 2 (2001) 57–194.

[41] R. Etxeberria, P. Larranaga, Global optimization using Bayesian networks, in: Second Symposium on Artificial Intelligence (CIMAF-99). Habana, Cuba, 1999, pp. 332–339.

[42] C.W. Ahn, R.S. Ramakrishna, D.E. Goldberg, Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world, in: Genetic and Evolutionary Computation – GECCO 2004, Springer, 2004, pp. 840–851.

[43] B.T. Zhang, S.Y. Shin, Bayesian Evolutionary optimization using Helmholtz machines, in: Parallel Problem Solving from Nature PPSN VI, Springer, 2000, pp. 827–836.

[44] B.T. Zhang, A Bayesian framework for evolutionary computation, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), IEEE, 1999, pp. 722–728.

[45] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.

[46] G. Salton, M.J. McGill, Introduction to modern information retrieval, 1986.

[47] T. Landauer, P. Foltz, D. Laham, An introduction to latent semantic analysis, Discourse Process. 25 (2–3) (1998) 259–284.

[48] Y. Wang, M. Wang, H. Fujita, Word sense disambiguation: A comprehensive knowledge exploitation framework, Knowl.-Based Syst. 190 (2020) 105030.

[49] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, H. Fujita, Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering, Inform. Sci. 514 (2020) 88–105.

[50] D.M. Blei, Latent Dirichlet allocation, J. Mach. Learn. Res. 34 (4) (2003) 993–1022.

[51] W. Zhang, R.A. Clark, Y. Wang, W. Li, Unsupervised language identification based on latent Dirichlet allocation, Comput. Speech Lang. 39 (2016) 47–66.

[52] H. Liang, R. Fothergill, T. Baldwin, RoseMerry : A Baseline message-level sentiment classification system, in: The 9th International Workshop on Semantic Evaluation (SemEval 2015), no. SemEval, 2015, pp. 551–555.

[53] R. Ju, P. Zhou, C.H. Li, L. Liu, An efficient method for document categorization based on word2vec and latent semantic analysis, in: Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se, IEEE, 2015, pp. 2276–2283.

[54] L. Niu, X. Dai, J. Zhang, J. Chen, Topic2Vec: Learning distributed representations of topics, in: Proceedings of 2015 International Conference on Asian Language Processing, IALP 2015, 2016, pp. 193–196.

[55] M. Campr, K. Ježek, Comparing Semantic Models for Evaluating Automatic Document Summarization, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9302, Springer, 2015, pp. 252–260.

[56] C. Republic, T. Mikolov, Statistical Language Models Based on Neural Networks (Ph.D. dissertation), 2012.

[57] Y. Bengio, H. Schwenk, J.S. Senécal, F. Morin, J.L. Gauvain, Neural Probabilistic Language Models, in: Studies in Fuzziness and Soft Computing, vol. 194, Springer, 2006, pp. 137–186.

[58] J. Elman, Finding structure in time* 1, Cogn. Sci. 14 (2) (1990) 179–211.

[59] J. Turian, L.A. Ratinov, Y. Bengio, Word representations: A simple and general method for semi-supervised learning, in: ACL 2010, Proceedings of the Meeting of the Association for Computational Linguistics, July 11–16, 2010. Uppsala, Sweden, 2010, pp. 384–394.

[60] A. Mnih, G.E. Hinton, A scalable hierarchical distributed language model, in: Advances in Neural Information Processing Systems, 2008, pp. 1–8.

[61] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using em, Mach. Learn. 39 (2) (2000) 103–134.

[62] M. B. Christopher, Probability distributions, in: Pattern Recognition and Machine Learning, Springer, 2006, pp. 67–136.

[63] R. Rehurek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010, pp. 45–50.