Cognitive Evolutionary Search to Select Feature Interactions for Click-Through Rate Prediction

Runlong Yu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
yrunl@mail.ustc.edu.cn

Xiang Xu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
demon@mail.ustc.edu.cn

Yuyang Ye
Department of Management Science and Information Systems, Rutgers University
Newark, USA
yuyang.ye@rutgers.edu

Qi Liu*
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
qiliuql@ustc.edu.cn

Enhong Chen
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
cheneh@ustc.edu.cn

ABSTRACT

Click-Through Rate (CTR) prediction of intelligent marketing systems is of great importance, in which feature interaction selection plays a key role. Most approaches model interactions of features by the same pre-defined operation under expert guidance, among which improper interactions may bring unnecessary noise and complicate the training process. To that end, in this paper, we aim to adaptively evolve the model to select proper operations to interact on feature pairs under task guidance. Inspired by natural evolution, we propose a general *Cognitive EvoLutionary Search (CELS)* framework, where cognitive ability refers to the malleability of organisms to orientate to the environment. Specifically, we conceptualize interactions as genomes, models as organisms, and tasks as natural environments. Mirroring how genetic malleability develops environmental adaptability, we thus diagnose the fitness of models to simulate the survival rates of organisms for natural selection, thereby an evolution path can be planned and visualized, offering an intuitive interpretation of the mechanisms underlying interaction modeling and selection. Based on the CELS framework, we develop four instantiations including individual-based search and population-based search. We demonstrate how individual mutation and population crossover enable CELS to evolve into diverse models suitable for various tasks and data, providing ready-to-use models. Extensive experiments on real-world datasets demonstrate that CELS significantly outperforms state-of-the-art approaches.

*Qi Liu is the corresponding author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599277

CCS CONCEPTS

• **Information systems** → *Recommender systems*; *Computational advertising*; • **Computing methodologies** → *Search methodologies*; *Bio-inspired approaches*; *Cognitive science*.

KEYWORDS

Cognitive Ability; Evolutionary Learning; Nature Inspired Computing; Feature Selection; Recommender Systems

ACM Reference Format:
Runlong Yu, Xiang Xu, Yuyang Ye, Qi Liu, and Enhong Chen. 2023. Cognitive Evolutionary Search to Select Feature Interactions for Click-Through Rate Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599277

1 INTRODUCTION

Click-Through Rate (CTR) prediction is of great importance in the accurate targeting of intelligent marketing systems [12, 26, 37, 46, 62]. It aims to estimate the ratio of clicks to the impression of a recommended item for a user. Since research has shown that interactions of feature pairs can provide predictive abilities beyond what those features can provide individually [55, 62], this brings out the fundamental research problem of feature interaction selection.

A general feature selection framework consists of four steps [7], that is: 1) generation strategy; 2) evaluation criteria; 3) stopping condition; 4) result validation. Based on the framework, many developed feature selection methods typically fall into three categories: 1) filter; 2) wrapper; 3) embedded. Filter and wrapper methods often suffer from poor robustness and inefficiency, making them unsuitable for large-scale or high-dimensional datasets [59]. In contrast, embedded methods are gaining popularity due to their reduced computational requirements and lesser overfitting issues [27]. Current embedded methods predominantly employ expert-designed operations for feature pair interactions, like factorization machines

(FM) [42] that model interactions via inner product. Shallow models, however, exhibit limited representation capabilities, inspiring the use of implicit deep learning models such as Neural FM (NFM) [16] to model higher-order feature interactions. Despite this, such models capture few low-order interactions [11, 28], a problem addressed by hybrid structures like Wide&Deep, which combines shallow and deep components for learning memorization and generalization [6, 11]. However, most extant methods model interactions of features by the same pre-defined operation under expert guidance, among which improper features and interactions may bring unnecessary noise and complicate the training process.

To that end, we expect an ideal feature interaction selection approach should adaptively evolve the model to select proper operations to interact on feature pairs under task guidance. One way to implement such an evolution of a model is evolutionary learning [59, 66, 68], a nature-inspired meta-heuristic algorithm that resolves complex search problems in machine learning [52]. It also refers to evolutionary AutoML, in which different components of models are automatically determined based on evolution, such as architecture and hyperparameters [52]. While earlier evolution-based feature selection methods performed filters and wrappers [59], these are not practical for massive high-dimensional commercial data. Recent approaches have employed AutoML to search relevant features or neural architectures for embedded methods [22, 30, 31, 47]. For instance, AutoFIS [31] optimizes the relevance of features and interactions but models all interactions with the same pre-defined operation, limiting its adaptability. Although there are neural architecture search (NAS) approaches like AutoCTR and AutoFeature [22, 47], they are almost all built on blocks with complex functionality, such as multilayer perceptrons (MLP) or FM, then rely on an unsatisfiable assumption (known as DARTS [33], which expects the search space to be a continuous, differentiable convex function) to relax the discrete choice of a block to a continuous softmax over all blocks.

Inspired by the evolution and functioning of natural organisms, this paper proposes a general *Cognitive EvoLutionary Search (CELS)* framework, where cognitive ability refers to the malleability of organisms to orientate to the environment [13, 17, 21, 44]. Guided by cognitive science principles, we posit that cognitive functioning, which encompasses factors like consciousness, awareness, memory, problem-solving, and analytical capabilities, is a powerful determinant of its adaptability to the environment [4, 25, 49, 53]. As an example, during early childhood, the human brain is at its most malleable, allowing it to effectively orient to tasks within its environment. This inherent malleability, pivotal for developing various mental activities, serves as a measure of intelligence [18, 21, 44].

In the context of feature interaction selection, our approach emphasizes searching fine-grained basic-level operations rather than coarse-grained upper-level architectures. Specifically, we regard the relationship between interactions and tasks as the relationship between genomes and natural environments. It is easy to understand that, different traits confer different rates of survival and fitness, thereby reflecting the selection process of features and interactions. In a groundbreaking move, we introduce a fitness diagnosis technique expressly tailored for cognitive evolution approaches. This technique stands in stark contrast to the traditional fitness evaluation, which predominantly uses numerical values to quantify a model's fitness. The newly proposed fitness diagnosis technique allows us to explore further into the model, illuminating the capacities of its internal components. This mirrors and impacts the cognitive abilities of the organism. In doing so, an evolutionary path can be mapped out and visualized, thereby enhancing the interpretability of how the model selects operations to interact on feature pairs that suit the task better. Based on the CELS framework, we develop four instantiations including individual-based search: (1,1)-CELS, (1+1)-CELS, and population-based search: (n,1)-CELS, (n+1)-CELS. We summarize the four instantiations in the following:

- **(1,1)-CELS:** We liken features to nucleotides and operations to linkages. To explore the fittest operation that generates a task-friendly interaction of each feature pair, we broaden our search space with various operation types, much like binding rules for nucleotides. An initial model is created with operations randomly assigned to feature pairs. We discriminate the relevance of features and interactions by the online learning optimizer. Mutation, as the source of genetic variations, probabilistically occurs when the relevance of an interaction drops to a threshold, altering the operation of the interaction. Thus, the parent model is replaced by the mutated offspring model for the next generation.
- **(1+1)-CELS:** In contrast to (1,1)-CELS, where the parent model is deterministically replaced by the offspring, (1+1)-CELS contests the offspring model with the parent. Only if the offspring's fitness is at least as good as the parent's, it succeeds the parent. Otherwise, the offspring is discarded. The 1/5 successful rule is introduced to adapt the search region, that is, if previous iterations struggled to improve the model, the current model might be nearing a local optimum, indicating a need to lower mutation probability to exploit the promising region near the optimum.
- **(n,1)-CELS:** Instead of a single parent, the use of population reduces the risk of settling in local optima. To achieve population-based search, (n,1)-CELS initializes n random models. Then, the crossover mechanism is applied to generate offspring from parent models, and mutation is applied to maintain diversity. In (n,1)-CELS, the worst fit parent is discarded and the offspring joins the new parent pool.
- **(n+1)-CELS:** We merge the strategies of (1+1)-CELS and (n,1)-CELS in (n+1)-CELS, where the new parents are selected from the parents and the offspring. Only if the offspring's fitness matches or surpasses the worst parent, it joins the next generation's parents. The 1/5 success rule is also applied to adapt the search regions for the population.

Following CELS evolution, we propose a model functioning stage that leverages selected features and interactions to further capture non-linear interactions, akin to gene decoding. In this stage, we use a Wide&Deep structure, with the deep segment using vectorized interactions fed into an MLP, and the wide segment containing a linear model of features, keeping their relevance as attention units.

2 RELATED WORK

2.1 Feature Interaction Selection

Since research has shown that interactions of feature pairs can provide predictive abilities beyond what those features can provide individually [34, 55, 59, 62], feature interaction selection based

on embedded methods attracts much participation from CTR prediction [10, 32, 58, 67]. Earlier, scholars designed operations for modeling interactions explicitly. Factorization machines (FM) [42] projected features into low-dimensional vectors and modeled interactions via inner product. Field-aware FM (FFM) [19] allowed features to have multiple latent vectors interacting with different fields. However, these models had limited representation capabilities, prompting implicit deep learning models like Attention FM (AFM) [56] and Neural FM (NFM)[16], which stacked deep neural networks atop FM outputs to model higher-order interactions. FNN uses FM to pre-train low-order interactions and then feeds embeddings into an MLP [65]. IPNN (also known as PNN) also uses the interaction results of the FM layer but does not rely on pre-training [40, 41]. Nevertheless, these models lacked interpretability. Lian et al. [28] argued that implicit models focus more on high-order cross features but overlook low-order cross features. Hence, recent advancements propose the Wide&Deep hybrid network structure for learning memorization and generalization [6, 11, 54]. Wide&Deep framework attracts industry partners from the beginning. As for the first Wide&Deep model proposed by Google, it combines a linear model and an MLP [6]. Later on, DeepFM uses an FM layer to replace the shallow part [11]. Similarly, Deep&Cross [54] and xDeepFM [28] take the outer product of features at the bit- and vector-wise level respectively. Though achieving some success, most of them follow a manner, which models interactions of features by the same pre-defined operation under expert guidance and equally enumerates all features and interactions, therein suffering from two main problems. First, they cannot ensure the learning abilities of models because their architectures are poorly adaptable to tasks and data. Second, useless features and interactions can bring unnecessary noise and complicate the training process.

2.2 Evolutionary Learning

Evolutionary learning refers to a class of nature-inspired meta-heuristic algorithms that solve complicated search problems in machine learning [1, 20, 60, 68]. Evolutionary learning also refers to evolutionary AutoML [52]. The general evolution-based feature selection framework consists of five steps [52], including 1) encoding; 2) initialization; 3) search strategy; 4) feature set modeling; 5) model fitness evaluation. Extant researches suggest that evolution-based feature selection can only be applied to filters and wrappers, because of the limitation of model fitness evaluation [52, 59]. Specifically, wrapper methods use the performance of the learning algorithm as its evaluation criterion, while filter methods use the intrinsic characteristics of the data [52, 59]. On the other hand, embedded approaches simultaneously select features and learn a classifier, therefore conventional algorithms cannot evaluate the fitness of the model [52, 59]. Only genetic programming (GP) and learning classifier systems (LCSs) are able to perform embedded feature selection, but they are not practical [7, 14, 24, 29, 39, 52, 59]. To solve the limitation, we propose a fitness diagnosis technique that can reveal the abilities of inside components of the model during training. Recent approaches have employed AutoML to search neural architectures for CTR prediction [22, 30, 47]. However, they are almost all built on blocks with complex functionality, then rely on an unsatisfiable assumption to relax the categorical choice of a block to a continuous softmax over all blocks. Usually, their block is an architecture-level algorithm, hence large population size and massive generations are usually required to address the huge search space issue. In contrast, our work searches fine-grained basic-level operations, and it uses discrete selection rather than the relaxation trick in DARTS [33]. As far as we know, CELS is the first to utilize a meta-heuristic mutation mechanism for operation search.

2.3 Cognitive Ability

Inspired by cognitive science principles, cognitive functioning such as consciousness, awareness, memory, problem-solving, and analytical skills are pivotal in adaptability [25, 36, 49, 53]. These enable organisms to perceive, remember, and react appropriately to environmental shifts. From a cognitive neuroscience perspective, the ability to develop such cognitive functions is defined as malleability, which orients organisms to environment [9, 13, 17, 21]. Conversely, some studies view adaptive behavior and functioning as a form of cognition [21, 44]. Recent developments in cognitive AI advocate for the quantification and simulation of organismic cognitive abilities [8, 53]. In our work, CELS infuses these cognitive abilities into the model, thereby allowing it to be further diagnosed and evolved.

3 PRELIMINARIES

3.1 Problem Statement

We define a general form of feature interaction selection problem. If the dataset consists of N instances $(\boldsymbol{f}, y)$, where $\boldsymbol{f} = [f_1, \ldots, f_m]$ indicates instance features including m fields, and $y \in \{1, 0\}$ indicates a user's click behavior, the feature interaction selection problem can be defined as how to precisely give the predictive result through the learned model $\hat{y} : \mathcal{M}(\boldsymbol{f}, \boldsymbol{g}(\boldsymbol{f})) \mapsto [0, 1]$, where $\boldsymbol{g}$ denotes the set of operations to interact on feature pairs, and $\boldsymbol{g}(\boldsymbol{f})$ denotes the set of interactions. Usually, the prediction model $\mathcal{M}$ suffers a Logloss (cross-entropy loss) function [51], given as:

$$\mathcal{L}(\mathcal{M}) = -\frac{1}{|B|} \sum_{t \in B} y_t \log(\hat{y_t}) + (1 - y_t) \log(1 - \hat{y_t}), \qquad (1)$$

where B denotes the set of instance indices in a mini-batch, $\hat{y}$ denotes the predictive result given through the learned model.

3.2 Operations

As the fundamental components in feature interaction selection, operations are regarded as functions where two individual features are converted into an interaction. For the sake of simplicity, we adopt four representative operations as candidate operations to present instantiations of CELS, i.e., $\boldsymbol{g} = \{\oplus, \otimes, \boxplus, \boxtimes\}$, which are highly used in previous work [22, 30, 47]. As shown in Figure 1, the following operations are available for selection:

- **Element-wise sum** ($\oplus$): It takes two input vectors of dimension $|f|$ and outputs a vector of dimension $|f|$ that contains their element-wise sum. It has no parameters.
- **Element-wise product** ($\otimes$): It takes two input vectors of dimension $|f|$ and outputs a vector of dimension $|f|$ that contains their element-wise product. It has no parameters.
- **Concatenation & feed-forward layer** ($\boxtimes$): It takes two input vectors of dimension $|f|$, concatenates them, and passes them through a feed-forward layer with ReLU activation functions to reduce the dimension of the output vector to $|f|$.

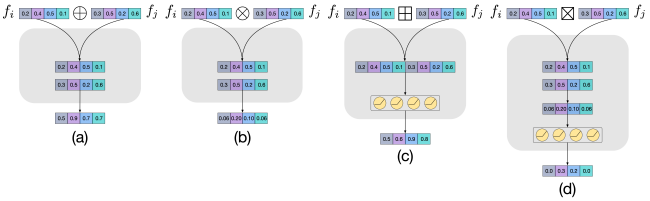

Figure 1: Candidate operations to interact on feature pairs.

- **Element-wise product & feed-forward layer** ($\boxplus$): It takes two input vectors of dimension $|f|$, passes their element-wise product through a feed-forward layer with ReLU activation functions to output a vector of dimension $|f|$.

The complexities of $\oplus$ and $\otimes$ are both $O(|f|)$. Also, the complexities of $\boxtimes$ and $\boxplus$ are both $O(|f|)$. In practice, we simultaneously optimize operations $\boxtimes$, $\boxplus$ with feature embedding $\boldsymbol{f}$. Thus, for an interaction of a feature pair, the complexity is $O(|f|)$.

4 COGNITIVE EVOLUTIONARY SEARCH

In this section, we will initially introduce the general framework of *Cognitive EvoLutionary Search (CELS)*. Then, we will present four instantiations, including individual-based search: (1,1)-CELS, (1+1)-CELS, and population-based search: (n,1)-CELS, (n+1)-CELS. After that, the model functioning stage will be proposed. Finally, we will provide a summary and remark for CELS.

4.1 General Framework of CELS

We consider a feature interaction selection process as an evolutionary search process, which can be viewed as a natural organism striving to evolve better traits for higher rates of fitness. The traits of an organism can be inherited via genomes. We regard the relationship between features and operations as the relationship between nucleotides and linkages. Following various linkages of nucleotides, we extend the operation set with four types of operations as the search space, i.e., $\boldsymbol{g} = \{\oplus, \otimes, \boxplus, \boxtimes\}$. If g_k is a candidate operation from the operation set $\boldsymbol{g}$, an interaction $g_k(f_i, f_j)$ is modeled by the operation g_k applied to a feature pair (f_i, f_j).

For an organism, if the genomic regions decode a phenotype that benefits survival, it will have better fitness; if the phenotype decoded by the genomic regions does not benefit survival, it will have worse fitness. The evolutionary strategy should favor the preservation of beneficial genetic information, which motivates us to measure the importance of features and interactions through relevance parameters. Therefore, our intuitive goal is to discriminate the relevance of features and interactions, so as to enhance the relevant features and interactions, meanwhile, weaken irrelevant features or change some interactions contributing little.

We let $\boldsymbol{\alpha} = \{\alpha_i | 1 \leqslant i \leqslant m\}$ and $\boldsymbol{\beta} = \{\beta_{i,j} | 1 \leqslant i < j \leqslant m\}$ respectively denote the relevance parameters of features $\boldsymbol{f}$ and interactions $\boldsymbol{g(f)}$. The predictive response of the current learning model can be given as follows: $\hat{y} = \mathcal{M}(\boldsymbol{\alpha} \cdot \boldsymbol{f}, \boldsymbol{\beta} \cdot \boldsymbol{g}(\boldsymbol{f})) = \mathcal{M}\big(\sum_{i=1}^{m} \alpha_i \cdot f_i, \sum_{1 \leqslant i < j \leqslant m} \beta_{i,j} \cdot g(f_i, f_j)\big)$, where the α_i is the relevance of the feature f_i, and $\beta_{i,j}$ is the relevance of the interaction $g(f_i, f_j)$.

Traditional fitness evaluation techniques of evolutionary learning primarily utilize the predictive response as a fitness measure for the current model, which then guides the stochastic mutation process to generate offspring models. We argue that this type of fitness measure only provides an overview of the model's overall status, thus the mutation guided by it lacks clear directionality. By contrast, we prefer to diagnose the learning abilities of inside components of the model, thereby enabling the mutation of specific, less effective components. The mutation mechanism is demonstrated in Figure 2. When the relevance of interactions is low (indicated by a lighter color), these are targeted for mutation, meaning that the operations of the interactions change into the other operations.

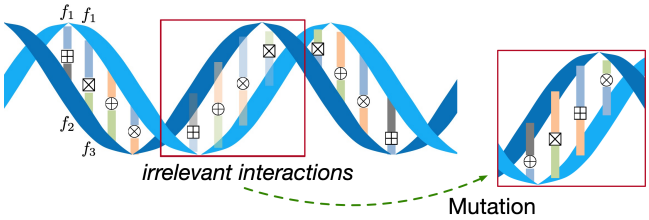

Figure 2: An illustration of the mutation mechanism.

We use the online learning optimizer to discriminate the relevance of features and interactions. Specifically, we propose to optimize $\boldsymbol{\alpha}, \boldsymbol{\beta}$ simultaneously with feature embeddings, where feature embeddings are learned by Adam optimizer [23], while $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are learned by regularized dual averaging (RDA) optimizer [5, 57]. The reason why we can guarantee sparse solutions of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ is because of the truncation mechanism of the RDA optimizer. When the absolute value of the cumulative gradient average value in a certain dimension is less than a threshold, the weight of that dimension will be set to 0, resulting in the sparsity of the relevance [31, 57]. We update $\boldsymbol{\alpha}, \boldsymbol{\beta}$ at each gradient step t with data B_t as:

$$\boldsymbol{\alpha}_{t+1}, \boldsymbol{\beta}_{t+1} = \mathcal{S}_{h(t,\gamma)}\left\{(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0) - \gamma \sum_{i=0}^{t} \nabla \mathcal{L}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i; B_i)\right\}, \quad (2)$$

where $\mathcal{S}_h : v \mapsto \text{sign}(v) \cdot \max\{|v| - h, 0\}$ is the soft-thresholding operator, $\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0$ are initializers chosen at random, γ is the learning rate, $h(t, \gamma) = c\gamma^{1/2}(t\gamma)^{\mu}$ is the tuning function, c and μ are adjustable hyperparameters as a trade-off between accuracy and sparsity. To avoid the expensive inner optimization of the gradient of feature embeddings and relevance $\boldsymbol{\alpha}, \boldsymbol{\beta}$, the parameters are updated together using one-level optimization with gradient descent on the training set by descending on $\boldsymbol{f}, \boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ based on:

$$\nabla_{\boldsymbol{f}} \mathcal{L}(\boldsymbol{f}_{t-1}, \boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_{t-1}) \text{ and } \nabla_{\boldsymbol{\alpha},\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{f}_{t-1}, \boldsymbol{\alpha}_{t-1}, \boldsymbol{\beta}_{t-1}). \quad (3)$$

In this setting, $\boldsymbol{f}, \boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can explore their search space freely until convergence.

To summarize, CELS iteratively diagnoses the relevance of features and interactions of the current model (i.e., the parent) and then mutates the operations of the irrelevant interactions into the other operations to generate the new model (i.e., the offspring). The search terminates when a predefined halting condition is satisfied.

We place CELS within the extensive realm of evolutionary computation. Traditional evolutionary algorithms typically treat individuals as atomic solutions, using a numerical response for fitness evaluation and subsequently discarding less competitive individuals. However, this approach tends to neglect the intrinsic abilities of the models. On the contrary, CELS introduces a novel perspective

Algorithm 1 Cognitive Evolutionary Search: (1+1)-CELS

Input: Training dataset of N instances $(\boldsymbol{f}, y)$, operation set $\boldsymbol{g}$.

1: Randomly create a model $\mathcal{M}$ with initialized relevance $\boldsymbol{\alpha}, \boldsymbol{\beta}$.
2: Generate a offspring model $\mathcal{M}'$ by applying mutation to $\mathcal{M}$.
3: **while** $t < T_{max}$ **do**
4: Update $\mathcal{M}'$ by descending $\boldsymbol{f}, \boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$. ▷ Eq. (3)
5: **if** $\text{mod}(t, \tau) = 0$ **then**
6: **if** $\mathcal{L}(\mathcal{M}') \leq \mathcal{L}(\mathcal{M})$ **then**
7: Update the parent model $\mathcal{M}$ with $\mathcal{M}'$.
8: **end if**
9: Generate a offspring $\mathcal{M}'$ by applying mutation to $\mathcal{M}$.
10: **end if**
11: **if** $\text{mod}(t, ep * \tau) = 0$ **then**
12: Update the mutation probability σ according to the 1/5 successful rule. ▷ Eq. (6)
13: **end if**
14: **end while**
15: **return** the model $\mathcal{M}$.

from cognitive evolution. CELS integrates the concept of cognitive ability, particularly neuro malleability or genetic malleability to develop environmental adaptability. Consequently, CELS moves beyond evaluating an individual based on surface-level "appearances" and opts to closely diagnose the model at the genetic level. Our approach allows for the assessment of a model's intrinsic abilities and facilitates mutation mechanism as targeted.

4.2 (1,1)-CELS

In this subsection, a simple instantiation of CELS for individual-based search is presented. In (1,1)-CELS, an initial model $\mathcal{M}$ is created where operations are randomly assigned to feature pairs. We simultaneously optimize relevance parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ with feature embeddings $\boldsymbol{f}$. Mutation, as the source of genetic variations, probabilistically occurs in model $\mathcal{M}$ when the relevance of an interaction drops to a threshold, which can be formalized as follows:

For an interaction $g_k(f_i, f_j)$, when $\beta_{i,j}$ drops to a threshold λ for every τ steps, the mutation is applied with probability σ, which means that, to regenerate a new interaction, the operation g_k of the interaction $g_k(f_i, f_j)$ mutates into another operation g_l, given as:

$$g_k = \begin{cases} g_l \text{ with probability } \sigma, & \textbf{if} \quad \beta_{i,j} < \lambda, \\ g_k, & \textbf{otherwise}. \end{cases} \tag{4}$$

where g_l is randomly selected from the operation set as $g_l = \{g \mid g \in \boldsymbol{g}, g \neq g_k\}$. After the mutation, the new interaction $g_l(f_i, f_j)$ replaces the irrelevant interaction $g_k(f_i, f_j)$ and its corresponding relevance $\beta_{i,j}$ is reinitialized as $\beta'_{i,j}$. In this way, $\mathcal{M}$ is replaced, its offspring $\mathcal{M}'$ containing new interactions with relevance $\boldsymbol{\beta}'$ and features with relevance $\boldsymbol{\alpha}'$ (inherited from $\boldsymbol{\alpha}$) participates in the next τ steps.

4.3 (1+1)-CELS

Compared to (1,1)-CELS where the parent model is deterministically replaced by the offspring model, (1+1)-CELS generates the offspring model and optimizes it then competes it with the parent model. The following meta-heuristic rule is adopted in (1+1)-CELS to select the parent of the next generation:

$$\begin{cases} \text{discard } \mathcal{M}, & \textbf{if} \quad \mathcal{L}(\mathcal{M}') \leq \mathcal{L}(\mathcal{M}), \\ \text{discard } \mathcal{M}', & \textbf{otherwise}. \end{cases} \tag{5}$$

In this way, only if the offspring's fitness is at least as good as the parent model, it becomes the parent of the next generation. Otherwise, the offspring is discarded.

In general, the value of mutation probability σ can be adapted during the search and may also vary over interactions. For the sake of simplicity, we are initialized with the same value of σ for all interactions and then adapt σ for every $ep * \tau$ steps according to the 1/5 successful rule [3, 63], given as:

$$\sigma = \begin{cases} \sigma / r & \textbf{if} \quad c/ep > 0.2, \\ \sigma * r & \textbf{if} \quad c/ep < 0.2, \\ \sigma & \textbf{if} \quad c/ep = 0.2. \end{cases} \tag{6}$$

where r is a hyperparameter that is suggested to be set beneath 1, and c is the times that a replacement happens (i.e., $\mathcal{M}'$ is preserved) during the past $ep * \tau$ steps. Eq. (6) is designed based on the following intuition. A large c implies that the search process frequently found better models in the past iterations, and the current model might be far away from the optimum. Thus, the mutation probability should be increased (by $1/r$ times) to help the search process explore the global promising region. On the other hand, if the search process frequently failed to achieve a better model in the past iterations, the current model might be close to a local optimum, so the mutation probability should be reduced (by r times) to help the search process exploit the promising region near the local optimum. Algorithm 1 outlines the pseudo-code of the (1+1)-CELS. By contrast, (1,1)-CELS skips the model competition at line 6, as well as the adaption of the mutation probability at lines 11-13.

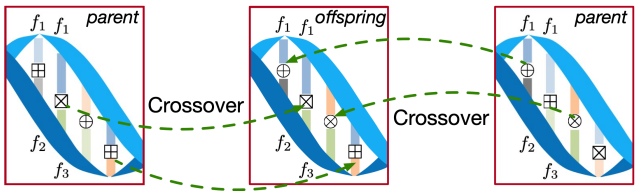

Figure 3: An illustration of the crossover mechanism.

4.4 (n,1)-CELS

The use of population (rather than generating the offspring model from a single parent) has been proven to make search processes less prone to settle in local optima [15, 38, 50]. To achieve a population-based search with the population size of n $(n > 1)$, (n,1)-CELS initializes n random models as a population: $\mathcal{P} = \{\mathcal{M}_1, \cdots, \mathcal{M}_v, \cdots, \mathcal{M}_n\}$. For a model $\mathcal{M}_v$, we use $\boldsymbol{\alpha}^{\mathcal{M}_v}$ and $\boldsymbol{\beta}^{\mathcal{M}_v}$ to respectively denote the relevance of features and interactions. The models in the population may have various operations for interacting on a feature pair (f_i, f_j), i.e., $g^{\mathcal{P}}_{i,j} = \{g^{\mathcal{M}_1}_{i,j}, \cdots, g^{\mathcal{M}_v}_{i,j}, \cdots, g^{\mathcal{M}_n}_{i,j}\}$.

To iteratively generate an offspring model from the population, we propose a crossover mechanism applied to multiple parent models. We choose to select the fittest operation (of which interaction has the largest relevance) from the population to interact on the

Algorithm 2 Cognitive Evolutionary Search: (n+1)-CELS

Input: Training dataset of N instances $(\boldsymbol{f}, y)$, operation set $\boldsymbol{g}$.

1: Randomly create a population $\mathcal{P}$ of n models, of which any $\mathcal{M}_v \in \mathcal{P}$ has initialized its relevance $\boldsymbol{\alpha}^{\mathcal{M}_v}$ and $\boldsymbol{\beta}^{\mathcal{M}_v}$.
2: Generate a offspring model $\mathcal{M}'$ by applying crossover to $\mathcal{P}$.
3: Update $\mathcal{M}'$ by applying mutation to $\mathcal{M}'$.
4: **while** $t < T_{max}$ **do**
5: Update $\mathcal{M}'$ by descending $\boldsymbol{f}, \boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$. ▷ Eq. (3)
6: **if** $\mathrm{mod}(t, \tau) = 0$ **then**
7: Select the worst parent $\mathcal{M} = \arg\max_{\mathcal{M}_v \in \mathcal{P}} \mathcal{L}(\mathcal{M}_v)$.
8: **if** $\mathcal{L}(\mathcal{M}') \leq \mathcal{L}(\mathcal{M})$ **then**
9: Update the population $\mathcal{P}$ by replacing $\mathcal{M}$ with $\mathcal{M}'$.
10: **end if**
11: Generate a offspring $\mathcal{M}'$ by applying crossover to $\mathcal{P}$.
12: Update $\mathcal{M}'$ by applying mutation to $\mathcal{M}'$.
13: **end if**
14: **if** $\mathrm{mod}(t, ep * \tau) = 0$ **then**
15: Update the mutation probability σ according to the 1/5 successful rule. ▷ Eq. (6)
16: **end if**
17: **end while**
18: **return** the best model in $\mathcal{P} : \mathcal{M} = \arg\min_{\mathcal{M}_v \in \mathcal{P}} \mathcal{L}(\mathcal{M}_v)$.

feature pair for the offspring model, given as:

$$g_{i,j}^{\mathcal{M}'} = \arg \max_{g_{i,j}^{\mathcal{M}_v} \in g_{i,j}^{\mathcal{P}}} \beta_{i,j}^{\mathcal{M}_v}. \tag{7}$$

We illustrate the crossover mechanism of two parents in Figure 3. If the relevance of interactions of a parent is small (shown as lighter color), the operations should be selected from the other parents whose relevance of the interactions is large. Meanwhile, interactions of the offspring inherit their relevance from respective parents.

After the crossover, the mutation mechanism is applied to the current model $\mathcal{M}'$ to maintain the diversity of the population. Then, we optimize $\mathcal{M}'$ and let it replace the worst parent in the population, which can be seleted by $\mathcal{M} = \arg\max_{\mathcal{M}_v \in \mathcal{P}} \mathcal{L}(\mathcal{M}_v)$.

4.5 (n+1)-CELS

Compared to (n,1)-CELS, (n+1)-CELS generates the offspring via crossover and mutation, then only if the offspring's fitness is at least as good as the worst parent, it replaces the worst parent in the population. Otherwise, the offspring is discarded. We also use the 1/5 successful rule to adapt the mutation probability σ for the population. Algorithm 2 outlines the pseudo-code of the (n+1)-CELS. By contrast, (n,1)-CELS skips the model competition at line 8, as well as the adaption of the mutation probability at lines 14-16.

4.6 Model Functioning

Through recurrent processes of replication and transcription, genetic information is decoded to create corresponding protein sequences. This procedure proposes a spectrum of potential functions for an organism. In an attempt to mimic this natural process, we retrain the model. Relevant features and interactions are selected according to their relevance fitness parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$. If $\alpha_i = 0$ or $\beta_{i,j} = 0$, the corresponding features or interactions are fixed to be discarded permanently. To further capture non-linear interactions with selected relevant features and interactions, in the model functioning stage, we use a Wide&Deep structure, with the deep segment using vectorized interactions fed into an MLP, and the wide segment containing a linear model of features:

$$\hat{y} = \mathrm{Sigmoid}\Big(\boldsymbol{w}_{wide}[\alpha_1 \cdot f_1, \ldots, \alpha_m \cdot f_m] + \mathrm{MLP}\Big(\big[\beta_{1,2} \cdot g(f_1, f_2), \ldots, \beta_{m-1,m} \cdot g(f_{m-1}, f_m)\big]\Big) + b\Big), \tag{8}$$

where $\boldsymbol{w}_{wide}$ is the weight vector of the linear model, and b is the bias. The relevance $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are fixed and serve as attention units.

4.7 Summary and Remark

We instantiate CELS with a series of meta-heuristics, which deeply explore the role of cognitive evolution in feature interaction selection, but the instantiation of CELS is not limited to our proposals. According to the canonical nomenclature of evolution strategies [2, 14], practitioners can easily generalize CELS to derivatives as (n,κ)-CELS and (n+κ)-CELS. Besides, the complexities of the mechanisms of CELS such as mutation and crossover are linear in practice.

5 EXPERIMENTAL EVALUATION

We conduct extensive experiments on three publicly available benchmarked datasets to investigate:

- **RQ1.** How does the effectiveness of the four implementations of CELS compare to other baseline models?
- **RQ2.** What are the training costs of running CELS? Is it practical from an efficiency standpoint?
- **RQ3.** How does the model evolve to select appropriate operations under task guidance?

5.1 Datasets

We use three publicly available advertising datasets for CTR prediction in the experiments, i.e., Criteo[1], Avazu[2], Huawei[3]. The statistics are reported in Table 2. We describe the three advertising datasets and the pre-processing steps below.

- **Criteo**, a renowned CTR prediction benchmark dataset by Criteo AI Lab, encompasses billions of data points, with a small subset released during the 2013 Criteo Display Advertising Challenge. We use data from "day 6-12" for training and evaluation, transforming 13 numerical fields into one-hot features via bucketing, with infrequent features (appearing less than 20 times) labeled as a dummy "other" feature.
- **Avazu**, released in the 2014 Avazu Click-Through Rate Prediction contest. This dataset contains user mobile behaviors, including ad clicks. It comprises 23 feature fields spanning from user/device features to ad attributes. We select a 10-day data subset for training and evaluation.
- **Huawei**, released in the 2020 Huawei DIGIX Advertisement CTR Prediction, consists of seven consecutive days of advertising behavior data. It encompasses 35 feature fields, ranging from user/device features to ad attributes.

[1] https://www.kaggle.com/c/criteo-display-ad-challenge/data
[2] https://www.kaggle.com/c/avazu-ctr-prediction/data
[3] https://www.kaggle.com/louischen7/2020-digix-advertisement-ctr-prediction

Table 1: Performance comparison. Impr is the relative AUC improvement. * and $^\diamond$ represent significance level p-value $< 10^{-3}$ and p-value < 0.05 of comparing CELS with the best baseline (indicated by underlined numbers).

Model Name	Criteo			Avazu			Huawei		
	AUC(%)	Logloss	Impr(%)	AUC(%)	Logloss	Impr(%)	AUC	Logloss	Impr(%)
LR [43]	77.84	0.4692	4.28	76.27	0.3896	4.90	0.7658	0.1322	4.06
FM [42]	79.40	0.4583	2.23	78.95	0.3746	1.34	0.7860	0.1281	1.39
AFM [56]	79.85	0.4520	1.65	78.67	0.3759	1.70	0.7925	0.1259	0.56
FFM [19]	80.70	0.4449	0.58	79.04	0.3738	1.23	0.7945	0.1245	0.30
Wide&Deep [6]	79.84	0.4523	1.67	79.28	0.3723	0.92	0.7916	0.1258	0.67
Deep&Cross [54]	79.87	0.4522	1.63	79.35	0.3719	0.83	0.7947	0.1242	0.28
NFM [16]	80.42	0.4469	0.93	79.13	0.3730	1.11	0.7910	0.1256	0.75
DeepFM [11]	80.36	0.4481	1.01	79.39	0.3715	0.78	0.7917	0.1247	0.66
IPNN [40]	80.92	0.4420	0.31	79.70	0.3698	0.39	0.7939	0.1240	0.38
OPNN [40]	81.02	0.4417	0.19	79.43	0.3715	0.73	0.7937	0.1242	0.40
xDeepFM [28]	80.94	0.4421	0.28	79.63	0.3707	0.48	0.7950	0.1239	0.24
AutoInt [48]	80.82	0.4433	0.43	79.29	0.3725	0.91	0.7909	0.1262	0.76
AutoGroup [30]	80.89	0.4426	0.35	79.82	0.3691	0.24	0.7949	0.1244	0.25
AutoFIS-FM [31]	80.62	0.4452	0.68	79.45	0.3712	0.70	0.7887	0.1268	1.04
AutoFIS-IPNN [31]	80.94	0.4422	0.28	79.78	0.3695	0.29	0.7945	0.1245	0.30
(1,1)-CELS	81.11	0.4406	0.07	79.87	0.3685	0.18	0.7951	0.1239	0.23
(1+1)-CELS	81.12	0.4405	0.06	79.90	0.3686	0.14	0.7962	0.1237	0.09
(n,1)-CELS	81.14	0.4403	0.04	79.95	0.3681	0.08	0.7962	0.1236	0.09
(n+1)-CELS	**81.17***	**0.4400***	-	**80.01***	**0.3678***	-	**0.7969**$^\diamond$	**0.1229**$^\diamond$	-

Table 2: The statistics of three real-world datasets.

Dataset	#Instances	#Categories	#Fields	Positive ratio
Criteo	45,840,617	34,290,882	39	0.34
Avazu	40,428,967	9,449,445	23	0.20
Huawei	41,907,133	1,096,074	35	0.04

Table 3: Training cost of CELS (GPU hours).

Dataset	(1,1)-CELS	(1+1)-CELS	(n,1)-CELS	(n+1)-CELS
Criteo	~0.54	~0.54	~0.82	~0.82
Avazu	~0.72	~0.72	~1.00	~1.00
Huawei	~0.49	~0.50	~0.60	~0.60

5.2 Experimental Settings

5.2.1 Metrics. Two widely used metrics, AUC (Area Under Curve) and Logloss (cross-entropy loss) are selected for evaluation.

5.2.2 Baselines. The baseline models we employ for comparison are the standard and contemporary state-of-the-art models, which include: LR [43], FM [42], AFM [56], FFM [19], Wide&Deep [6], Deep&Cross [54], NFM [16], DeepFM [11], IPNN and OPNN [40], xDeepFM [28], AutoInt [48], AutoGroup [30], AutoFIS-FM and AutoFIS-IPNN [31]. We calculate the p-values for CELS and the best baseline by repeating the experiments five times by changing the random seeds. The two-tailed pairwise t-test is performed to detect the significant difference. We use * and $^\diamond$ to represent significance level p-value $< 10^{-3}$ and p-value < 0.05.

5.2.3 Hyperparameter Settings. Optimal hyperparameters of each model are identified through grid search, and sensitivity analyses for key hyperparameters of CELS are provided in the **Appendix**.

5.2.4 Implementation Details. To implement CELS[4], we use RDA [5, 57] as the online optimizer to discriminate the relevance of features and interactions, with the learning rate $\gamma = 10^{-3}$, adjustable hyperparameters $c = 0.5, \mu = 0.8$. We set the mutation mechanism as the mutation threshold $\lambda = 0.2$, the mutation probability $\sigma = 0.5$, and the mutation step size $\tau = 10$. For (1+1)-CELS and (n+1)-CELS, we set the 1/5 successful rule as the adaptation hyperparameter $r = 0.99$, the adaptation step size $ep = 10$. For (n,1)-CELS and (n+1)-CELS, we set the population size as $n = 4$. In the model functioning stage, we set the depth of MLP as 2 with 400 neurons per layer.

[4]We have released the source code of CELS at https://github.com/RunlongYu/CELS.

5.3 Performance Comparison

Table 1 reports the performance of CELS averaged over five repetitions and compared baselines on three datasets. Impr is the relative AUC improvement. In practice, an improvement of 0.001-level in AUC or Logloss is usually regarded as being significant, because it will lead to a large increase in the company's revenue due to a large user base, which has been pointed out in many existing articles [35, 45, 61, 64]. From the experimental results, we have the following key observations:

Firstly, most neural network models surpass shallow models such as LR and FM. This suggests that MLPs are capable of learning non-linear interactions and providing representation capabilities. Additionally, OPNN, AutoGroup, and xDeepFM, representatives of interactions modeled by element-wise outer products, inner products, and compressed interaction networks, respectively, are the best performing baseline models on Criteo, Avazu, and Huawei. This aligns with our assertion that we cannot definitively say which pre-designed operations are superior, given their limited adaptability to tasks and datasets.

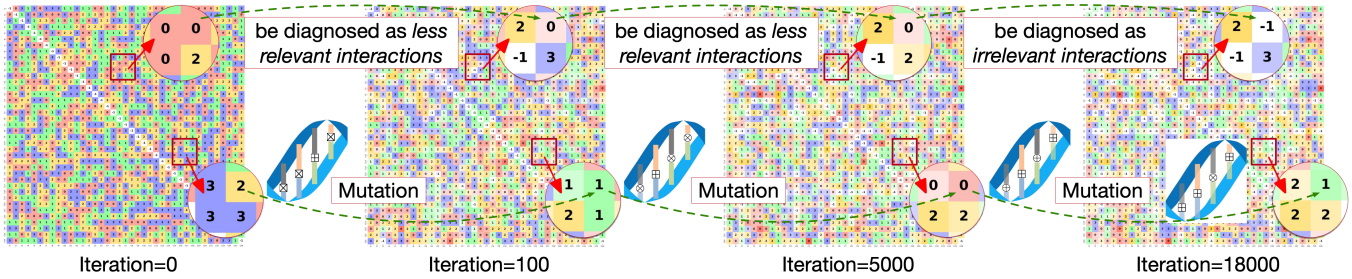

Figure 4: Visualization of the evolution path traced by gene maps of (1,1)-CELS algorithm on Criteo dataset.

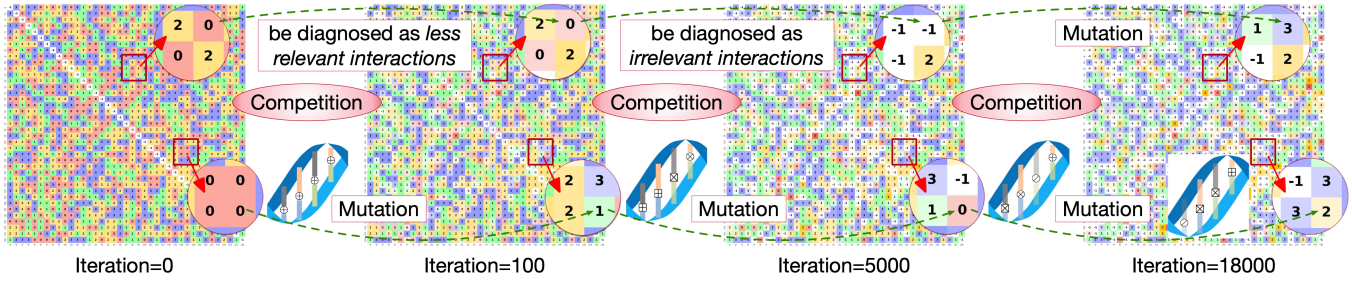

Figure 5: Visualization of the evolution path traced by gene maps of (1+1)-CELS algorithm on Criteo dataset.

Secondly, instantiations of CELS show substantial improvement over baselines in terms of AUC and Logloss on all three datasets. This gain in performance can be attributed to adaptively modeling interactions by evolving to find suitable operations. Instead of equally enumerating all features and interactions, instantiations of CELS can diagnose the relevance of features and interactions, so as to enhance relevant features and relevant interactions, and weaken irrelevant features or mutate interactions contributing little.

Third, from the performance comparison of instantiations of CELS, we can observe that (1+1)-CELS outperforms (1,1)-CELL, and (n+1)-CELS outperforms (n,1)-CELS. These indicate that competition among the offspring model and parent models is effective, and the 1/5 successful rule can adapt the mutation probability to converge the search region. Meanwhile, the superior performance of (n,1)-CELS over (1,1)-CELL and the superior performance of (n+1)-CELS over (1+1)-CELS indicate that, the use of population can make search processes more diverse, so as to be not easily affected by the initial models and less prone to settle in local optima.

Beyond accuracy, we also consider the training cost of four CELS instantiations as shown in Table 3. All experiments are conducted on a Linux server with a single NVIDIA Tesla A100 GPU. The training cost aligns with our analysis, with the entire training process for each dataset taking less than an hour. This represents a significant improvement in efficiency compared to previous automated machine learning approaches that often required multiple GPUs running for days [47]. The training cost on the Avazu dataset is slightly longer, mainly due to the higher set dimension of embeddings compared to the other datasets.

5.4 Visualization of Evolution Path

To clarify how the model evolves to select suitable operations under task guidance, we visualize the evolution path of CELS on the Criteo dataset, which comprises 13 integer feature fields "$I1 \sim I13$" and 26 categorical feature fields "$C1 \sim C26$". If we use the following encoding $\oplus = 0, \otimes = 1, \boxplus = 2, \boxtimes = 3$, the diagnosed fitness of models can be represented as a matrix. Additionally, we assign distinct colors to operations to construct a **gene map** of the model, where each gene indicates an interaction, i.e., red "0", green "1", yellow "2", blue "3". For example, green "1" in the block "$I1 \times C12$" means that element-wise product $\otimes$ is diagnosed as the fittest operation for feature $I1$ to interact with feature $C12$. To express the relevance of interactions, we emphasize certain genes based on the relevance parameters. Darker colors denote interactions that are diagnosed as more relevant ones, while lighter colors denote less relevant ones.

The evolution paths of individual-based search, i.e., (1,1)-CELS and (1+1)-CELS, traced by gene maps are respectively visualized in Figure 4 and Figure 5. The evolution paths of population-based search, i.e., (n,1)-CELS and (n+1)-CELS, traced by gene maps are respectively visualized in Figure 6 and Figure 7. For each evolution path, we zoom in on the gene maps of some local genomes and illustrate their nucleotides and linkages (i.e., features and operations), therein the less relevant interactions are shown as the lighter color.

For individual-based search, we can observe that, in Figure 4 and Figure 5, operations were randomly assigned to model all interactions at the beginning, and interactions shared equal relevance. Later, the gene map evolved rapidly in the early iterations, some interactions were discovered as relevant interactions (the color becomes darker), while most of the others became less important (the color becomes lighter), and some mutated into new interactions. The relevance parameters of some interactions were reduced and truncated to 0. We discarded these irrelevant interactions, so their genes became white "−1" (we use $\oslash$ to encode these linkages in the illustration). Finally, the search process tends to converge, and model's internal operations barely change. Comparing (1,1)-CELS

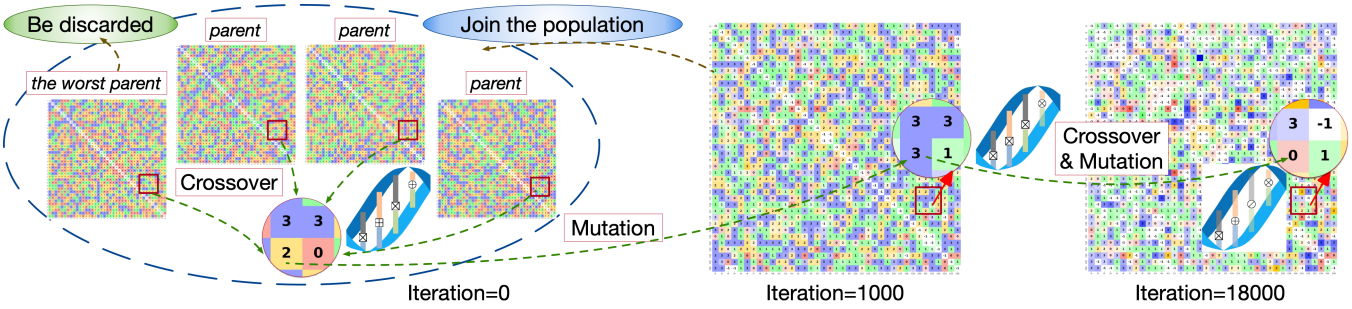

Figure 6: Visualization of the evolution path traced by gene maps of (n,1)-CELS algorithm on Criteo dataset.

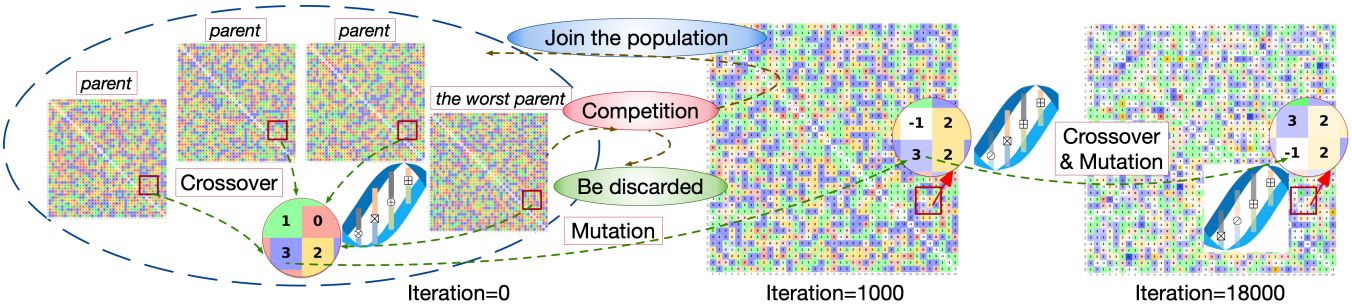

Figure 7: Visualization of the evolution path traced by gene maps of (n+1)-CELS algorithm on Criteo dataset.

and (1+1)-CELS, the convergence speed of (1+1)-CELS is faster than that of (1,1)-CELS, which benefits from the competition between the parent model and the offspring model. Owing to the competition and the 1/5 successful rule, the mutation probability can be adapted to converge the search region.

The biggest defect of individual-based search is that it is highly affected by initialization. Although the model tends to be adaptable to tasks and data through mutation and evolution, the final model still has many similarities with the initial model. This is largely due to the fact that the search space is high-dimensional and the individual-based search has insufficient exploration power and is prone to fall into local optima. Population-based search overcomes this defect with more parents and the crossover mechanism.

We can observe that, in Figure 5 and Figure 6, operations were randomly assigned to four initial models at the beginning. Afterward, the crossover mechanism was applied to four parents, resulting in a new model, which is mutated to generate offspring. After massive crossover and mutation, the final models evolved by population-based search show few similarities with any initial models. This is further consistent with our proposal that the use of population can facilitate the genotypic diversity of models, which makes the search process less prone to settle in local optima and better explore the global regions.

Meanwhile, excessive exploration can cause the search process not to converge. We can observe that the convergence speed of (n,1)-CELS is rather slow. By comparison, the convergence speed of (n+1)-CELS is faster owing to the competition among parents and offspring and the 1/5 successful rule. Therefore, we believe (n+1)-CELS achieves an ideal balance between exploration and exploitation in model evolution.

6 CONCLUSION AND FUTURE WORK

This paper presents a nature-inspired evolutionary learning approach to select feature interactions for CTR prediction, namely *Cognitive EvoLutionary Search (CELS)*. CELS is a fresh approach to intelligent marketing and feature interaction selection, as it can adaptively evolve the model to select proper operations to interact on feature pairs under task guidance, thereby enhancing prediction performance. When viewed from the model's cognitive ability standpoint, CELS brings the model's genetic malleability, endowing it with the adaptability to varying environments. From an evolutionary computation perspective, the fitness diagnosis technique is instrumental in assessing a model's intrinsic abilities and driving a targeted mutation mechanism. This enables the visualization of an evolution path. Based on the CELS framework, we develop four instantiations including individual-based search and population-based search. We conducted extensive experiments on three publicly available advertising datasets. Experimental results have proved that CELS can significantly outperform state-of-the-art approaches.

The CELS framework of utilizing individual mutation and population crossover is new thinking that helps us build models under task guidance and is not limited to instantiations in this paper. In future work, we will pay more attention to the internal order of model evolution and establish a theoretical framework for its evolution behavior. Furthermore, we encourage more task-oriented instantiations to be proposed based on our work.

ACKNOWLEDGMENTS

This research was funded by grants from the National Key Research and Development Program of China (No. 2021YFF0901003) and the National Natural Science Foundation of China (No. U20A20229).

REFERENCES

[1] Thomas Back. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press.

[2] H. Beyer. 2007. Evolution strategies. *Scholarpedia* 2, 8 (2007), 1965. https://doi.org/10.4249/scholarpedia.1965 revision #193589.

[3] Hans-Georg Beyer and Hans-Paul Schwefel. 2002. Evolution strategies–a comprehensive introduction. *Natural Computing* 1 (2002), 3–52.

[4] Scott L Boyar, Grant T Savage, and Eric S Williams. 2022. An adaptive leadership approach: The impact of reasoning and emotional intelligence (EI) abilities on leader adaptability. *Employee Responsibilities and Rights Journal* (2022), 1–16.

[5] Shih-Kang Chao and Guang Cheng. 2019. A generalization of regularized dual averaging and its dynamics. *arXiv preprint arXiv:1909.10072* (2019).

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[7] Manoranjan Dash and Huan Liu. 1997. Feature selection for classification. *Intelligent Data Analysis* 1, 1-4 (1997), 131–156.

[8] Yanyan Dong, Jie Hou, Ning Zhang, and Maocong Zhang. 2020. Research on how human intelligence, consciousness, and cognitive computing affect the development of artificial intelligence. *Complexity* 2020 (2020), 1–10.

[9] Dennis Garlick. 2002. Understanding the nature of the general factor of intelligence: the role of individual differences in neural plasticity as an explanatory mechanism. *Psychological Review* 109, 1 (2002), 116.

[10] Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. 2021. An embedding learning framework for numerical features in ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 2910–2918.

[11] Huifeng Guo, Ruiming Tang, et al. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 1725–1731.

[12] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual graph enhanced embedding neural network for ctr prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 496–504.

[13] Angela Gutchess. 2014. Plasticity of the aging brain: new directions in cognitive neuroscience. *Science* 346, 6209 (2014), 579–582.

[14] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, Mar (2003), 1157–1182.

[15] Jun He and Xin Yao. 2002. From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 6, 5 (2002), 495–511.

[16] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*. 355–364.

[17] William Huitt and John Hummel. 2003. Piaget's theory of cognitive development. *Educational Psychology Interactive* 3, 2 (2003), 1–5.

[18] William E Hyland et al. 2022. Interest–ability profiles: An integrative approach to knowledge acquisition. *Journal of Intelligence* 10, 3 (2022), 43.

[19] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*. 43–50.

[20] Chia-Feng Juang, Ching-Yu Chou, and Chin-Teng Lin. 2022. Navigation of a fuzzy-controlled wheeled robot through the combination of expert knowledge and data-driven multiobjective evolutionary learning. *IEEE Transactions on Cybernetics* 52, 8 (2022), 7388–7401.

[21] Kenneth S Kendler, Eric Turkheimer, Henrik Ohlsson, Jan Sundquist, and Kristina Sundquist. 2015. Family environment and the malleability of cognitive ability: A Swedish national home-reared and adopted-away cosibling control study. *Proceedings of the National Academy of Sciences (PNAS)* 112, 15 (2015), 4612–4617.

[22] Farhan Khawar, Xu Hang, Ruiming Tang, Bin Liu, Zhenguo Li, and Xiuqiang He. 2020. Autofeature: Searching for feature interactions and their architectures for click-through rate prediction. In *ACM International Conference on Information and Knowledge Management (CIKM)*. 625–634.

[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[24] Ron Kohavi and George H John. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97, 1-2 (1997), 273–324.

[25] V Lee and A Thornton. 2021. Animal cognition in an urbanised world. *Frontiers in Ecology and Evolution* 9 (2021).

[26] Pan Li et al. 2021. Dual attentive sequential learning for cross-domain click-through rate prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 3172–3180.

[27] Xiaoping Li, Yadi Wang, and Rubén Ruiz. 2022. A survey on sparse learning models for feature selection. *IEEE transactions on Cybernetics* (2022), 1642–1660.

[28] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 1754–1763.

[29] Jung-Yi Lin, Hao-Ren Ke, Been-Chian Chien, and Wei-Pang Yang. 2008. Classifier design with feature selection and feature extraction using layered genetic programming. *Expert Systems with Applications* 34, 2 (2008), 1384–1393.

[30] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. 2020. AutoGroup: Automatic feature grouping for modelling explicit high-order feature interactions in CTR prediction. In *Proceedings of the 43rd international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*. 199–208.

[31] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, et al. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2636–2645.

[32] Huan Liu and Hiroshi Motoda. 2012. *Feature selection for knowledge discovery and data mining*. Vol. 454. Springer Science & Business Media.

[33] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*.

[34] Huijie Liu, Han Wu, Le Zhang, Runlong Yu, Ye Liu, Chunli Liu, Minglei Li, Qi Liu, and Enhong Chen. 2022. A hierarchical interactive multi-channel graph neural network for technological knowledge flow forecasting. *Knowledge and Information Systems* 64, 7 (2022), 1723–1757.

[35] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *2011 IEEE 11th international conference on data mining*. IEEE, 407–416.

[36] Qi Liu, Runze Wu, Enhong Chen, Guandong Xu, Yu Su, Zhigang Chen, and Guoping Hu. 2018. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology* 9, 4 (2018), 1–26.

[37] Ze Lyu, Yu Dong, Chengfu Huo, and Weijun Ren. 2020. Deep match to rank model for personalized click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 34. 156–163.

[38] Rammohan Mallipeddi and Ponnuthurai N Suganthan. 2008. Empirical study on the effect of population size on differential evolution algorithm. In *2008 IEEE Congress on Evolutionary Computation*. IEEE, 3663–3670.

[39] Durga Prasad Muni, Nikhil R Pal, and Jyotirmay Das. 2006. Genetic programming for simultaneous feature selection and classifier design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36, 1 (2006), 106–117.

[40] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, et al. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[41] Yanru Qu, Bohui Fang, Weinan Zhang, et al. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems* 37, 1 (2018), 1–35.

[42] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining (ICDM)*. IEEE, 995–1000.

[43] Matthew Richardson et al. 2007. Predicting clicks: estimating the click-through rate for new ads. In *International Conference on World Wide Web*. 521–530.

[44] Natascha Schaefer, Carola Rotermund, Eva-Maria Blumrich, et al. 2017. The malleable brain: plasticity of neural circuits and behavior–a review from students to students. *Journal of Neurochemistry* 142, 6 (2017), 790–811.

[45] Qixiang Shao, Runlong Yu, Hongke Zhao, Chunli Liu, Mengyi Zhang, Hongmei Song, and Qi Liu. 2021. Toward intelligent financial advisors for identifying potential clients: a multitask perspective. *Big Data Mining and Analytics* 5, 1 (2021), 64–78.

[46] Shu-Ting Shi, Wenhao Zheng, et al. 2020. Deep time-stream framework for click-through rate prediction by tracking interest evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 5726–5733.

[47] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 945–955.

[48] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *ACM International Conference on Information and Knowledge Management (CIKM)*. 1161–1170.

[49] Lukasz Stasielowicz. 2020. How important is cognitive ability when adapting to changes? A meta-analysis of the performance adaptation literature. *Personality and Individual Differences* 166 (2020), 110178.

[50] Ke Tang, Peng Yang, and Xin Yao. 2016. Negatively correlated search. *IEEE Journal on Selected Areas in Communications* 34, 3 (2016), 542–550.

[51] Wanjie Tao, Yu Li, Liangyue Li, Zulong Chen, Hong Wen, Peilin Chen, Tingting Liang, and Quan Lu. 2022. SMINet: State-aware multi-aspect interests representation network for cold-start users recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 36. 8476–8484.

[52] Akbar Telikani, Amirhessam Tahmassebi, et al. 2021. Evolutionary machine learning: A survey. *Comput. Surveys* 54, 8 (2021), 1–35.

[53] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, et al. 2020. Neural cognitive diagnosis for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 34. 6153–6161.

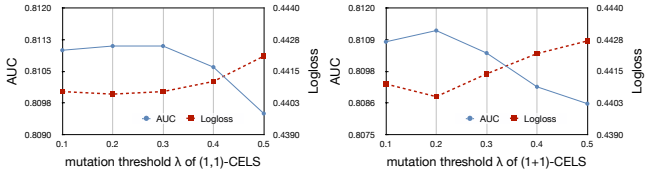

Figure 8: Mutation threshold of individual search on Criteo.

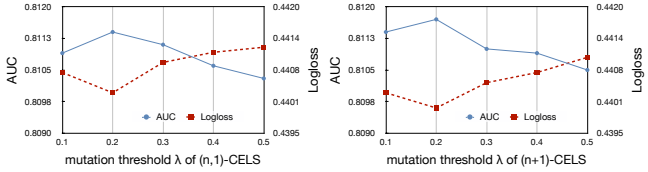

Figure 9: Mutation threshold of population search on Criteo.

[54] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[55] Zhiqiang Wang, Qingyun She, and Junlin Zhang. 2021. MaskNet: Introducing feature-wise multiplication to ctr ranking models by instance-guided mask. *arXiv preprint arXiv:2102.07619* (2021).

[56] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 3119–3125.

[57] Lin Xiao. 2009. Dual averaging method for regularized stochastic learning and online optimization. *Advances in Neural Information Processing Systems* 22 (2009).

[58] Yuexiang Xie, Zhen Wang, Yaliang Li, Bolin Ding, Nezihe Merve Gürel, Ce Zhang, Minlie Huang, Wei Lin, and Jingren Zhou. 2021. Fives: Feature interaction via edge search for large-scale tabular data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*. 3795–3805.

[59] Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. 2015. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation* 20, 4 (2015), 606–626.

[60] Hongyun Ye, Zhiwei Ni, and Enhong Chen. 2005. A mixed algorithm of integrated learning based evolutionary decision tree. In *Proceedings of Digital Anhui Doctoral Science and Technology Forum*.

[61] Runlong Yu, Qi Liu, Yuyang Ye, Mingyue Cheng, Enhong Chen, and Jianhui Ma. 2022. Collaborative list-and-pairwise filtering from implicit feedback. *IEEE Transactions on Knowledge & Data Engineering* 34, 06 (2022), 2667–2680.

[62] Runlong Yu, Yuyang Ye, Qi Liu, Zihan Wang, Chunfeng Yang, Yucheng Hu, and Enhong Chen. 2021. Xcrossnet: Feature structure-oriented learning for click-through rate prediction. In *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference (PAKDD)*. Springer, 436–447.

[63] Runlong Yu, Hongke Zhao, Zhong Wang, Yuyang Ye, Peining Zhang, Qi Liu, and Enhong Chen. 2019. Negatively correlated search with asymmetry for real-parameter optimization problems. *Jisuanji Yanjiu yu Fazhan/Computer Research and Development* 56, 8 (2019), 1746 – 1757.

[64] Kai Zhang, Hao Qian, Qing Cui, Qi Liu, Longfei Li, Jun Zhou, Jianhui Ma, and Enhong Chen. 2021. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*. 984–992.

[65] Weinan Zhang et al. 2016. Deep learning over multi-field categorical data. In *European Conference on Information Retrieval*. Springer, 45–57.

[66] Hongke Zhao, Xinpeng Wu, et al. 2021. CoEA: A cooperative–competitive evolutionary algorithm for bidirectional recommendations. *IEEE Transactions on Evolutionary Computation* 26, 1 (2021), 28–42.

[67] Guorui Zhou, Na Mou, Ying Fan, et al. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 5941–5948.

[68] Zhi-Hua Zhou, Yang Yu, and Chao Qian. 2019. *Evolutionary learning: Advances in theories and algorithms*. Springer.

A HYPERPARAMETER STUDIES

Empirically, the mutation threshold is perceived as the hyperparameter with the most significant influence on CELS as it directly impacts the conditions for triggering mutation. Moreover, for population-based searches, the population size is a paramount

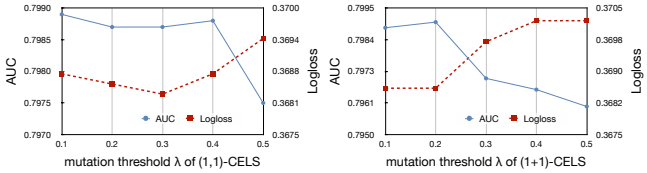

Figure 10: Mutation threshold of individual search on Avazu.

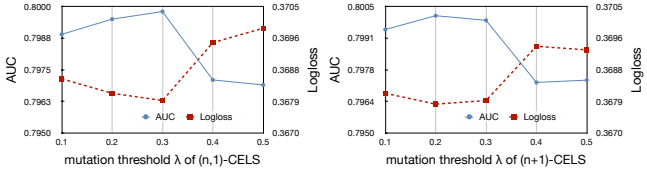

Figure 11: Mutation threshold of population search on Avazu.

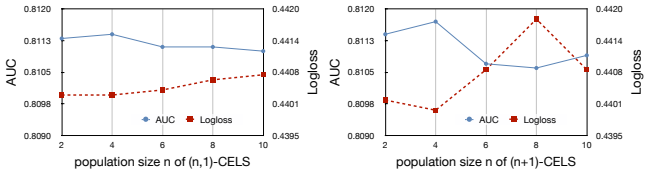

Figure 12: Population size n of population search on Criteo.

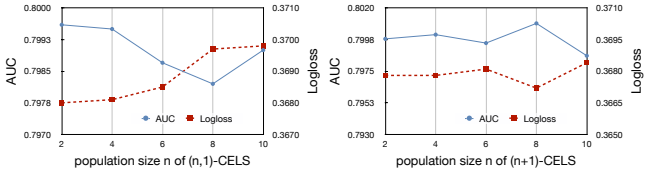

Figure 13: Population size n of population search on Avazu.

hyperparameter. To this end, we investigate the impact of hyperparameters of CELS, including the mutation threshold λ and the population size n. Figures 8 to 13 illustrate the experimental results on Criteo and Avazu datasets in terms of AUC and Logloss.

For the mutation threshold λ, as demonstrated in Figures 8 to 11, instantiations of CELS perform optimally with a smaller mutation threshold. Various instantiations exhibit similar patterns as the mutation threshold increases, namely a marked decline in model performance. This downturn is attributed to an overly random mutation caused by a large mutation threshold. When the mutation threshold is set to 0.5, it is almost equal to our initialized relevance parameters. Over-random mutation can obscure the evolutionary direction, resulting in generally subpar model performance.

For the population size n, as shown in Figures 12 to 13, the performance of (n,1)-CELS and (n+1)-CELS remains largely unaltered as the population size n elevates from 2 to 10. We postulate that when a large number of parents exist, each offspring undergoing the crossover mechanism derives from multiple parents, hence inheriting relevant interactions. However, this would weaken the relative relevance of interactions of an individual model, i.e., the offspring. In other words, it is hard to discriminate the relevance of interactions of this offspring, so the mutation of it is ineffective. This can also be ascribed to the pitfalls of excessive exploration.