Unsupervised Learning Style Classification for Learning Path Generation in Online Education Platforms

Zhicheng He1, Wei Xia1,*, Kai Dong2, Huifeng Guo1,*, Ruiming Tang1,*, Dingyin Xia2, Rui Zhang3
1Huawei Noah’s Ark Lab, 2Huawei CBG Edu AI Lab, 3www.ruizhang.info
Shenzhen, China
{hezhicheng9, xiawei24, dongkai4, huifeng.guo, tangruiming, xadingyin}@huawei.com; rayteam@yeah.net

ABSTRACT
Online education, which educates students that cannot be present at school, has become an important supplement to traditional education. Without the direct supervision and instruction of teachers, online education is always concerned with potential distractions and misunderstandings. Learning Style Classification (LSC) is proposed to analyze the learning behavior patterns of online learning users, based on which personalized learning paths are generated to help them learn and maintain their interests.

Existing LSC studies rely on expert-labeled, which is infeasible in large-scale applications, so we resort to unsupervised classification techniques. However, current unsupervised classification methods are not applicable due to two important challenges: C1 the unawareness of the LSC problem formulation and pedagogy domain knowledge; C2 the absence of any supervision signals. In this paper, we give a formal definition of the unsupervised LSC problem and summarize the domain knowledge into learnable model components (which addresses C1) and heuristics (which addresses C2). On top of that, a novel Deep Unsupervised Classifier with domain Knowledge (DUCK) is proposed to convert the discovered conclusions and domain knowledge into learnable model components (which addresses both C1 and C2), which significantly improves the effectiveness, efficiency, and robustness. Extensive offline experiments on both public and industrial datasets demonstrate the superiority of our proposed methods. Moreover, the proposed methods are now deployed in the Huawei Education Center, and the ongoing A/B testing results verify the effectiveness of the methods.

KEYWORDS
learning style classification, unsupervised classification, educational data mining, deep clustering, user behavior analysis

ACM Reference Format:

1 INTRODUCTION
In the ongoing COVID-19 pandemic, many schools are closed worldwide [27]. Online education (a.k.a. online learning, e-learning, etc.) has since been adopted by more governments and schools to maintain the teaching schedule. Huawei Education, which aims to unlock education everywhere for everyone, has been providing online education solutions for years1. Supported by advanced communication technologies, remote students are taught in real-time through online courses, collaborative classroom, and intelligent after-school guidance, etc. Along with the achieved progresses, however, new problems also emerge. Compared with traditional face-to-face instructions, interaction patterns between teacher and students are completely changed as illustrated in Figure 1. Online users (a.k.a. students or learners2) interact with teachers in an indirect manner through the online system, leading to potential inadequate instructions, distractions, or even dropouts [20]. To help users understand the courses and engage them in learning, it is necessary to explore and exploit their learning behavior patterns.

Figure 1: Illustration of traditional and online education.

Learning style modeling is such a pedagogy research that focuses on revealing users’ learning behavior patterns [7, 11, 23]. As shown in Table 1, the famous and widely applied Felder-Silverman Learning Style Model (FSLM) [11] depicts users’ learning behaviors according to how they perceive, sense, process, and understand new knowledge. Take information sensory as an example, users with the visual learning style prefer to learn from visual materials like pictures and videos, while verbal users prefer texts and audio. The teaching quality can be greatly improved if users’ learning styles can be correctly classified and properly utilized. Thus Learning Style Classification (LSC) is an important task for both traditional and online education.

Traditional pedagogy LSC studies are based on surveys, case studies, and expert analysis [7,14,19]. Apart from that, researchers also resort to machine learning techniques to solve the LSC task.

1https://e.huawei.com/en/solutions/industries/education
2To avoid misunderstandings, we only use the word “user” in the rest of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Woodstock ’18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06. $15.00
https://doi.org/10.1145/1122445.1122456
Table 1: The Felder-Silverman learning style model [11].

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Label</th>
<th>Learning preferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perception</td>
<td>Sensory</td>
<td>sights, sounds, sensations</td>
</tr>
<tr>
<td></td>
<td>Intuitive</td>
<td>possibilities, insights, hunches</td>
</tr>
<tr>
<td>Sensory</td>
<td>Visual</td>
<td>pictures, diagrams, graphs</td>
</tr>
<tr>
<td></td>
<td>Verbal</td>
<td>words, sounds</td>
</tr>
<tr>
<td>Processing</td>
<td>Active</td>
<td>through activities or discussions</td>
</tr>
<tr>
<td></td>
<td>Reflective</td>
<td>through introspection</td>
</tr>
<tr>
<td>Understanding</td>
<td>Sequential</td>
<td>in continual steps</td>
</tr>
<tr>
<td></td>
<td>Global</td>
<td>in large jumps, holistically</td>
</tr>
</tbody>
</table>

Due to the need for expert knowledge, existing machine learning based approaches are built on carefully labeled but small datasets [1, 10, 30, 31], which lead to inefficiency and ineffectiveness in large-scale applications. Further, questionnaire-based user labeling is infeasible in online education platforms due to poor user experience. Without enough supervision, classifiers fail to learn the correct mapping from user features to learning style labels, which is a severe and challenging problem in LSC research. In this paper, we resort to unsupervised classification techniques.

Unsupervised classification is a classic research issue that assigns samples to one of the inherent categories without using labeled training samples [3, 22, 26]. Existing research works can be roughly divided into two major groups: i) pre-training methods that apply classifiers pre-trained on other datasets; ii) rule-based methods derived from expert-made rules. However, both kinds of approaches are inapplicable because LSC differs from other unsupervised classification problems with two challenges. First, LSC is less understood in the machine learning community, and the solutions should accord with existing pedagogy domain knowledge (C1). Second, no external labels are available to supervise the classifier training in terms of neither pre-training nor transfer learning (C2).

To solve the above challenges, we build from scratch two principled LSC solutions. Through extensive pedagogy literature survey and data analyses, we give a formal definition of the unsupervised LSC problem in machine learning language and summarize the expert knowledge into problem-solving heuristics (for C1). A rule-based algorithm is first designed to explore the use of pedagogy domain knowledge and provide a naive solution (for C2). However, the rule-based algorithm is vulnerable to data noises. When online changes happen, laborious trial-and-error analyses is needed to adjust the settings. To overcome these obstacles, a new Deep Unsupervised Classifier with domain Knowledge (DUCK) is proposed to further improve model effectiveness, robustness, and efficiency. The DUCK model consists of a deep clustering backbone model to learn representations and conduct clustering in a self-supervised manner, based on which expert-made rules are used to classify data (for C2). To better integrate the domain knowledge, a behavioral preference constraint is put forward to extract user behavior patterns, and a smoothing component is also designed to understand the intrinsically skewed label distributions (for C1). As a result, DUCK model not only improves the performances significantly, but also has great efficiency and robustness. In summary, the contributions of this paper are as follows:

- To our best knowledge, we are the first to give a formal definition of the unsupervised LSC problem in machine learning language. We elaborate the properties of the unsupervised LSC problem in terms of both literature and data analysis, based on which some useful domain knowledge is summarized.
- Under the guidance of the summarized domain knowledge, both a naive rule-based algorithm and a novel model named DUCK are proposed to solve the unsupervised LSC problem. Our DUCK model not only overcomes the absence of labeled data, but also integrates the domain knowledge well.
- Experiments on both public and industrial datasets validate the superiority of the proposed methods. Moreover, our methods are now deployed in the Huawei Education Center, and the online A/B testing results further show the important value of this work.

2 RELATED WORKS

2.1 Learning Style Classification

Learning style models refer to a series of pedagogy researches that reveal how people learn and prefer to learn [11, 15]. Many different learning styles have been proposed to discover learning preferences from different perspectives such as the VARK model [11], the Kolb’s learning styles [19], and the VARK model [23]. Researchers also focus on the learning style classification (LSC) task [1, 10, 31] where the proposed approaches can be divided into traditional and machine learning based categories.

Traditional pedagogy LSC research is based on surveys, case studies, and expert analysis [7, 12, 14, 19]. For example, the Solomon questionnaire [25] is adopted in [8] to analyze learning styles and improve multimedia teaching. While Prithishkumar and Michael use the VARK questionnaire to classify the sensory modality preferences in learning. Due to the reliance on expert analysis, traditional approaches are troubled by the low efficiency thus not applicable to online environments.

Machine learning techniques have also been adopted to pedagogy researches. He et al. apply topic model to assess the curriculum design [18]. He et al. propose to use linear regression models to detect users’ dropout risks [17]. When it comes to LSC research, Aissou et al. propose a fuzzy c-means algorithm to classify students under the FSLSM [11]. Dutsimas and Temdee classify students under the VARK model with decision trees [10]. However, due to the high prices in acquiring ground-truth labels, existing models are all built on small datasets [30], which leaves two major concerns: i) the low extensibility, and ii) the risk of losing effectiveness in face of large datasets. In face of million-scale users, traditional human-labeled labeling becomes infeasible. So we propose an unsupervised LSC approach that solves the reliance on data labeling.

2.2 Unsupervised Classification

Unsupervised classification refers to a series of research works that classify samples without using labeled training data [3, 6, 22, 26]. Due to the various difficulties in data labeling, unsupervised classification has a wide range of applications including sentiment classification [6, 26], image classification [16, 22], and remote sensing [21]. To overcome the lack of supervision signals, pre-training [4], clustering [16, 29], and rule-based [6, 21, 26] approaches have all been explored to make the best use of data semantics and domain knowledge. As no pre-training datasets are available, this paper utilizes both clustering and expert-made rules.
3 PRELIMINARY

For the ease of elaboration, in this section, we first address some fundamental concepts including the notations, the adopted label system, and the problem definition. After that, we summarize the pedagogy background knowledge into problem-solving heuristics.

3.1 The FSLSM

The most vital issue of learning style research is the choice of learning style labels as many label systems have been proposed. This paper adopts the Felder-Silverman Learning Style Model (FSLSM) [11] which has found the most widespread applications and best fits the online education scenarios. As shown in Table 1, FSLSM classifies users from four orthogonal dimensions each of which has two opposite labels. Each user can be tagged with four FSLSM labels describing her/his learning styles from different perspectives. Two labels from different dimensions are orthogonal to each other, thus the LSC task can be split into four independent tasks. In this paper, we only consider the Sensory dimension because of its importance in learning material usage. However, the proposed method can be easily extended to other dimensions. Moreover, we add a new label “Neutral” to describe those non-significant users who are neither “Visual” nor “Verbal”, resulting into a label set \( L = \{0, 1, 2\} \) where 0, 1, and 2 represents “Visual”, “Neutral”, and “Verbal” respectively.

\[ l_i = \underset{l \in L}{\arg \max} P(l|u_i, b_i; \theta), \quad \forall u_i \in U, \]

where \( \Theta \) is the model parameter set and \( P(\cdot; \theta) \) is the probability function which can be implemented as a rule-based algorithm, a Bayesian model, or a deep neural network. Due to the absence of labeled data for the LSC task, we resort to the unsupervised classification approaches.

**Definition 2: Unsupervised LSC.** Given a set of unlabeled users \( U \) and the learning style label set \( L \), an unsupervised LSC algorithm classifies all users into \( |L| \) classes each of which indicates a unique learning style label in \( L \).

3.2 Problem Formulation

Before elaborating the technical details of proposed methods, the basic definitions need to be addressed

**Definition 1: Learning Style Classification (LSC).** Suppose there are \( |U| \) online learning users \( U = \{u_1, u_2, \ldots, u_{|U|}\} \), each user leaves some learning behavior logs \( B = \{b_1, b_2, \ldots, b_{|U|}\} \), LSC aims to assign a learning style label \( l_i \in L \) to each user \( u_i \) according to her/his learning behavior log \( b_i \).

In machine learning terminologies, LSC maximizes the posterior probability of \( u_i \)’s learning style label \( l_i \) given the behavior log \( b_i \)

\[ l_i = \underset{l \in L}{\arg \max} P(l|u_i, b_i; \theta), \quad \forall u_i \in U, \]

where \( \Theta \) is the model parameter set and \( P(\cdot; \theta) \) is the probability function which can be implemented as a rule-based algorithm, a Bayesian model, or a deep neural network. Due to the absence of labeled data for the LSC task, we resort to the unsupervised classification approaches.

3.3 Pedagogy Domain Knowledge

As a classic pedagogy research topic, abundant theories and findings [1, 5, 7, 14] about the LSC task have been proposed and provide valuable insights. Here we summarize the most important findings into two heuristics that guide our problem-solving. **Heuristic 1: User Behavioral Preference.** On the whole, visual users prefer to interact more with visual learning materials but less with verbal materials, and vice versa for verbal users.

In traditional pedagogy studies, expert-made questionnaires (e.g., the Solomon questionnaire [25]) are adopted to collect users’ behavioral preferences on learning materials, which is inefficient in large-scale online applications. What is more, both the learning materials and interaction forms are getting richer and more diverse in the many online education platforms. So there needs an automatic method to extract user behavioral preferences without human laboring. In this paper, for a better exploration and utilization of user behavioral preferences, we propose to integrate such heuristic into an unsupervised classifier in a learnable way. Specifically, anchor features are first selected by experts to quantitatively measure the behavioral preferences in each class. And a constraint component is further proposed to promote the classification results to fit the desired class-wise behavioral preferences. **Heuristic 2: Label Distribution Skewness.** In an unbiased sampling of users, the learning style label distribution is always skewed to the visual label.

As revealed in many pedagogy studies [12], there are always more visual users than verbal users. In some cases, researchers may even classify all users as visual [28]. However, it is an ineffective result to judge most users as visual due to the loss of distinctiveness. A reasonable classification result should make a trade-off between the inherent label distribution skewness and the usability of classification results. To achieve that, this paper proposes a new label distribution loss to tune the label distribution, thus liberates us from the tedious trial-and-error tests.

3.4 The Proposed Solutions

Inspired by the summarized heuristics, this work proposes both an intuitive rule-based algorithm and a deep unsupervised classifier for the LSC task, as illustrated in Figure 2. The rule-based algorithm is enlighten by both **Heuristic 1** and the unsupervised sentiment classification algorithms [6, 26] that decide the semantic orientation of a text according to the sentiment words in it. As shown in the gray block of Figure 2, anchor features are first extracted from the behavior logs, which serve similarly with the sentiment words. After that, anchor features are manually labeled as visual or verbal, and rules are made to calculate users’ statistical significance on them. Then users are classified by the overall significance on all anchor features. Finally, evaluation and parameter tuning are conducted for further improvements. To further improve robustness and effectiveness, a novel Deep Unsupervised Classifier with domain Knowledge (DUCK) is proposed, as in the golden block of Figure 2. A DEC backbone model (blue block) [29] is devised to learn data representations and soft clustering in a self-supervised manner. On top of that, a behavioral preference loss and a label smoothing loss are proposed to integrate **Heuristic 1** and **Heuristic 2** respectively into model learning. For the ease of understanding, in the rest of this paper, we first present the rule-based algorithm to help clarify the problem-solving workflow, followed by the description of the deep model.

4 RULE-BASED APPROACH

Due to the absence of labeled samples, conventional supervised learning solutions are infeasible, so a rule-based algorithm is first
proposed to explore the utilization of domain knowledge. The rule-based algorithm is inspired by the unsupervised sentiment classification algorithms [6, 26] that classify texts with positive words (e.g., “wonderful” and “beautiful”) as positive, while classify those with negative words (e.g., “awful” and “disgusting”) as negative. Similarly, we propose to classify users’ learning styles by rules on annotated behavior features, i.e., the anchor features.

4.1 Anchor Features

Anchor features are behavior features that can obviously describe users’ learning styles, based on which classification rules can be made. To extract anchor features, in this paper, we focus on the proactive behaviors which provide the best evidences for revealing their true intents. For example, if a user always proactively clicks or zooms in pictures but seldom clicks texts, she/he is likely to be a visual user. On the other side, if a user passively sees some picture within a text, it contributes little to the probability of being visual. After a lot of analysis (details can be referred to in section 6.2), all K representative proactive user behaviors are chosen as anchor features such as zoom in a picture or reply a comment

\[ \mathcal{A} = \{a_1, a_2, \ldots, a_K\}. \]  

(2)

The anchor features are sorted in a visual to verbal order. For example, if K < k, \( a_k \) is manually judged as more visual than \( a_{k-1} \). Then the numerical anchor feature values are calculated for each user \( u_i \)

\[ h_i = \{x_{i,a_1}, x_{i,a_2}, \ldots, x_{i,a_K}\}, \]  

(3)

where \( x_{i,a_k} \) denotes the \( k \)-th anchor feature value. For example, \( x_{i,a_1} \) may calculate the frequency of zooming in pictures, while \( x_{i,a_K} \) is the normalized count of comments \( u_i \) has clicked.

4.2 Feature Significance

After obtaining \( h_i \) for each user, significance scores are further calculated to eliminate the biases brought by anchor feature value distributions. A user’s significance score on the \( k \)-th anchor feature is decided by two threshold values \( \omega_{k,0} \) and \( \omega_{k,1} \) as

\[ s_{i,k} = \begin{cases} 1 & h_{i,k} > \omega_{k,1} \\ -1 & h_{i,k} < \omega_{k,0} \\ 0 & \omega_{k,0} \leq h_{i,k} \leq \omega_{k,1} \end{cases}. \]  

(4)

If \( h_{i,k} > \omega_{k,1} \), the positive significance score \( s_{i,k} = 1 \) is assigned. And if \( h_{i,k} < \omega_{k,0} \), the negative significance score \( s_{i,k} = -1 \) is assigned. Otherwise it is decided as non-significant with \( s_{i,k} = 0 \). It is non-trivial to manually set all 2K threshold values because different features have different distributions. In consideration of both the distinctiveness and fairness between anchor features, we use two percentile points \( \rho_0, \rho_1 \in \{1, 2, \ldots, 99\} \) to calculate the threshold values for each anchor feature \( a_k \):

\[ \omega_{k,0} = \text{Percentile}(\{h_{1,k}, h_{2,k}, \ldots, h_{|U|,k}\}, \rho_0), \]

\[ \omega_{k,1} = \text{Percentile}(\{h_{1,k}, h_{2,k}, \ldots, h_{|U|,k}\}, \rho_1), \]  

(5)

where Percentile(·) calculates the percentile value according to the given percentile points.

4.3 Classification Rules

Intuitively, if a user is always positively significant on visual anchor features but negatively significant on verbal ones, she/he is probably a visual user. Following that, users are classified by the average significance score on all anchor features

\[ \bar{s}_i = \frac{1}{K} \sum_{k=1}^{K} s_{i,k} \times \text{Type}(a_k). \]  

(6)

where the switch function \( \text{Type}(·) \) returns 1 for visual \( a_k \)’s but -1 for verbal \( a_k \)’s. Regulated by the switch function, a visual user always obtains a high and positive \( \bar{s}_i \) value, while a verbal user gets a low and negative \( \bar{s}_i \). So two thresholds \(-1.0 < \tau_0 < \tau_1 < 1.0 \) are adopted to decide the learning style labels by distinguishing the \( \bar{s}_i \) values,

\[ l_i = \begin{cases} 0 & \bar{s}_i > \tau_1 \\ 1 & \tau_0 \leq \bar{s}_i \leq \tau_1 \\ 2 & \bar{s}_i < \tau_0 \end{cases}. \]  

(7)

4.4 Parameter Tuning

There are four hyper parameters in the rule-based algorithm, i.e., the percentile points \( (\rho_0, \rho_1) \) and the significance threshold values \( (\tau_0, \tau_1) \). Following the conventional practice, trial-and-error tests are adopted to search for the best parameter settings. However, such a tuning process is not only laborious and time-consuming, but also vulnerable to data noises and online changes. For example, when the APP’s GUI is updated or some new function is added, distributions of anchor features will be influenced. Consequently, the previous parameter setting turns ineffective and tuning is needed again. To further improve robustness and effectiveness, a deep model is proposed in the next section.
5 DEEP UNSUPERVISED CLASSIFICATION

The rule-based algorithm has three main drawbacks: i) the poor robustness; ii) the underuse of rich user features; and iii) the neglect of Heuristic 2. To overcome these drawbacks, a novel Deep Unsupervised Classifier with domain Knowledge (DUCK) is proposed.

5.1 Backbone Model Architecture

All mainstream deep classifiers consist of two vital building components, i.e., representation learning and supervised label prediction. The representation learning component is responsible for feature semantic extraction and exploitation, while the supervised label prediction component interprets the supervision signals for parameter optimization. However, the absence of labels makes the supervised prediction infeasible. So the proposed DUCK model takes a clustering module to learn representations and mapping rules for label prediction. The clustering module should meet two conditions: i) it can be optimized with gradient techniques; ii) it is applicable to large-scale datasets. In this paper, we choose the classic Deep Embedded Clustering (DEC) model [29] (blue block in Figure 2) and leave other clustering algorithms to future works.

5.1.1 Representation Learning. To represent a user \( u_i \in U \), a feature vector \( x_i \in \mathbb{R}^d \) is extracted from the behavior log \( b_i \)

\[
x_i = \text{Extract}(b_i),
\]

where Extract(·) covers all offline feature pre-processing steps including feature selection, smoothing, and z-score normalization. To incorporate useful information as much as possible, more features are extracted into \( x_i \) than the \( K \) anchor features.

To facilitate subsequent learning processes, a multi-layer perceptron (MLP) component is applied to learn user representation \( e_i \in \mathbb{R}^d \) from the raw \( x_i \).

\[
e_i = \text{MLP}(x_i),
\]

where MLP(·) is made up with stacked dense layers with the ReLU activation function and dropout. Following the original work [29], MLP(·) is initialized with a stacked denoising autoencoder.

5.1.2 Soft Clustering. With the learned representations, users are first clustered, based on which they are further classified with rules. The soft probability of user \( u_i \) belonging to the \( j \)-th cluster is estimated according to the t-distribution similarity between \( e_i \) and the cluster centroid \( c_j \in \mathbb{R}^d \).

\[
q_{ij} = \frac{1 + ||e_i - c_j||^2}{\sum_{j'=1}^{K} (1 + ||e_i - c_{j'}||^2)^{-1}},
\]

where \( || \cdot ||^2 \) is the \( L_2 \) norm. The \( c_j \)s are first initialized by clustering the initialized user representations, then optimized during training. Clustering algorithms like KMeans [2] or Gaussian Mixture Model (GMM) [24] can be adopted as analyzed later in experiments.

5.1.3 Clustering Loss. For parameter optimization and clustering refinement purposes, DEC designs a self-supervised clustering loss function. The intuition is that high-confidence clustering assignments (i.e., big \( q_{ij} \)s) should be encouraged, while low-confidence clustering assignments (i.e., small \( q_{ij} \)s) should be punished. The target clustering probabilities are designed to sharpen the distribution of \( q_{ij} \)s:

\[
p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'}^{[L]} q_{ij'}^2/f_{j'}},
\]

where \( f_j = \sum_{i=1}^{[U]} q_{ij} \) is the soft clustering frequency. As illustrated in Figure 2, high-confidence \( q_{ij} \)s get even higher \( p_{ij} \)s, while small \( q_{ij} \)s get almost zero \( p_{ij} \)s. By minimizing the KL divergence between \( q_{ij} \)s and \( p_{ij} \)s, clear margins between different \( q_{ij} \)s are gradually achieved for each user \( u_i \).

\[
J^c = \text{KL}(P|Q) = \sum_{i=1}^{[U]} \sum_{j=1}^{[L]} p_{ij} \log \frac{p_{ij}}{q_{ij}},
\]

thus the clustering results are more convincing. The final cluster label \( c_i \) of \( u_i \) is estimated by finding the nearest centroid in the embedding space as

\[
c_i = \arg \min_j ||e_i - c_j||^2 = \arg \max_j q_{ij},
\]

where \( j \in \{1, 2, \ldots, [L]\} \). Equation (13) can be used to infer the clustering of testing users. For classification, the number of clusters is fixed as the number of learning style labels, i.e., \( [L] = 3 \).

5.1.4 Label Mapping Rules. After the deep model training and cluster label inference, each user is assigned into one of the clusters \( C_1, \ldots, C_{[L]} \). However, the class labels are still unclear because the clustering is obtained from inherent user behavioral similarities rather than supervised learning. Inspired by Heuristic 1, the learning style label of each cluster is first decided by the overall behavioral preferences of the contained users. So the same class label can then be assigned to all users within it.

The anchor features are used to decide the learning style label of each cluster. For all users in the \( j \)-th cluster \( C_j \), the average anchor feature vector is first calculated:

\[
\bar{h}_j = \text{Avg}(\{b_i | u_i \in C_j\}),
\]

where \( \text{Avg}(·) \) calculates the element-wise average value of a set of vectors. Further, a histogram is drawn to show the distribution feature values in a visual to verbal order. Due to the behavioral preferences in Heuristic 1, histogram of the visual cluster will have a descending trend, while a ascending trend can be observed for the verbal cluster, as illustrated in Figure 3. To obtain a quantitative measurement, the Pearson Correlation Coefficient (PCC) between \( \bar{h}_j \) and the anchor feature index \( k = \{1, \ldots, K\} \) is calculated:

\[
\rho_j = \text{PCC}(\bar{h}_j, k),
\]

where PCC(·, ·) calculates the PCC between two vectors. If \( \rho_j \rightarrow -1 \), a descending trend can be observed in the histogram, which indicates the visual class as in Figure 3. Otherwise \( \rho_j \rightarrow 0 \) and \( \rho_j \rightarrow +1 \) imply the neutral and verbal classes respectively. According to the above analysis, we rank all \( [L] = 3 \rho_j \)s, where the minimal, median, and maximal values correspond to the visual, neutral, and verbal classes respectively.

5.2 Behavioral Preference Constraint

The pedagogy domain knowledge is obviously underused in the backbone model, because Heuristic 1 is only used in the post-training label mapping and Heuristic 2 is not even considered.
Thus the classification results are likely to deviate from the pedagogy assumptions. To better solve the LSC problem, we attempt to integrate the two heuristics into model learning with two new components as in Figure 2.

5.2 Early Label Assignment

5.2.1 Early Label Assignment. Both the rule-based approach and the backbone model consider Heuristic 1 at the post-training stage. To make the best use of domain knowledge, we propose to consider it during model learning. However, the behavioral preference is loosely defined at the class-level rather than a strict constraint applied to individual users. Because the desired behavioral preferences differ between any two learning styles, applying Heuristic 1 in the learning process means that the class labels should be known during training, which is inconsistent with the backbone. To solve that, we improve the backbone with an early label assignment right after the initialization of clustering centroids.

After the clustering initialization, the PCCs \( \rho_{ij} \) are calculated for each of the \( |L| \) clusters as in Equation (15). With the same rules described in section 5.1.4, the \( j \)-th class can be assigned with a class label \( |C_j| \), which will be fixed in subsequent model training. On the one hand, with the indicated behavioral preference information, each \( C_j \) will enhance the representation learning of users in \( C_j \). On the other hand, the improved user representations will in turn promote the clustering to better fit the early assigned labels.

5.2.2 Behavioral Preference Assessment in Training. Since \( C_j \) is assigned a class label \( |L| \), right from the beginning, the desired class-wise behavioral preference trend is definite, thus the behavioral preference constraint can be applied. Within each training step, the soft clustering probabilities \( q_{ij} \) are used to decide the current cluster labels of all users as in Equation (13). Therefore, the behavioral preference of users in the \( j \)-th cluster \( C_j \) can be assessed with the PCC \( \rho_j \) of the mean anchor feature vector \( \bar{h}_j \) as in Equation (15).

5.2.3 Behavioral Preference Loss. Based on the early assigned class label \( |C_j| \) of each cluster, the target feature distribution PCC \( \hat{\rho}_{C_j} \) can be induced as in section 5.1.4, i.e., -1, 0, and 1 for the visual, neutral, and verbal clusters respectively. Then the behavioral preference loss is proposed to minimize the difference between the current \( \rho_j \) and the target \( \hat{\rho}_{C_j} \):

\[
J_P = \frac{1}{|L|} \sum_{j=1}^{|L|} |\rho_j - \hat{\rho}_{C_j}|. \tag{16}
\]

Through optimizing \( J_P \), the behavioral feature distribution of each cluster is driven to gradually fit Heuristic 1.

5.3 Label Distribution Smoothing

Apart from the behavioral preferences, the label distribution skewness in Heuristic 2 is also an important issue. On the one hand, equal-sized classification results do not fit the pedagogy domain knowledge. On the other hand, extremely skewed label distribution makes the classification results indistinguishable and unusable. To make a trade-off, the \( L^2 \) loss of the soft clustering matrix \( Q \) is adopted to smooth the clustering probabilities

\[
J^s = \frac{1}{|S|} \|Q\|_2^2 = \frac{1}{|S|} \sum_{i=1}^{|U|} \sum_{j=1}^{|L|} q_{ij}^2. \tag{17}
\]

With \( J^s \), DUCK can find a balancing point between the label distribution skewness and the usability of classification results.

5.4 Model Optimization

Finally, the newly proposed behavioral preference loss and the label smoothing loss are combined together with the clustering loss,

\[
J = \lambda_1 J^P + \lambda_2 J^q + \lambda_3 J^s, \tag{18}
\]

where \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \) are hyper-parameters to weigh different losses. For model optimization purpose, the Adam optimizer is adopted.

6 EXPERIMENTS

6.1 Datasets

To evaluate the effectiveness of the proposed approaches, we conduct experiments on both public and industrial datasets.

MOOC Dataset. The MOOC dataset is a publicly available dataset for online education research [13]. Many user behavior logs have been tracked and recorded. In the two-year learning activity logs, about 180 million behavior events are recorded. After deleting the inactive or abnormal users, about 460 thousand users are extracted for experiments, based on which 22 user behavioral features and 9 anchor features are extracted.

Industrial Dataset. The Industrial dataset is collected from the Huawei Education Center. User behavior events are tracked by the APP and reported to the backstage servers. Logs within one month are taken for experiments where approximately 1 million anonymous users are sampled, and about 38 million behavior events are extracted. After some data analysis and processing, 15 behavior features are selected and 7 of them are used as anchor features.

6.2 Anchor Feature Selection

The anchor features are used to decide the learning style label of each cluster and to construct the behavioral preference constraint, we show how the anchor features are extracted here. Following Heuristic 1, the more active an individual user interact with visual learning materials, the more likely her/his learning style is visual, and vise versa. So we focus on users’ frequent proactive behaviors conducted on representative learning materials. Table 2 shows some representative proactive behaviors such as play a video and create a comment, while the full list is not provided due to space limitation. After that, the occur rate of each proactive behavior is calculated by the normalized occurrence count, and the indicative degree is manually judged accordingly. Finally, the strongly indicative behaviors are selected as anchor features. Examples of the MOOC dataset are in Table 2, where three anchor features are selected.
Table 2: Examples of the anchor feature selection process.

<table>
<thead>
<tr>
<th>Behavioral object</th>
<th>Behavior</th>
<th>Style</th>
<th>Occur</th>
<th>Indicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>courseware,</td>
<td>play</td>
<td>Visual</td>
<td>1.0</td>
<td>Strong</td>
</tr>
<tr>
<td>figure, etc.</td>
<td>click</td>
<td>Visual</td>
<td>0.4626</td>
<td>Strong</td>
</tr>
<tr>
<td>text about,</td>
<td>close</td>
<td>Verbal</td>
<td>0.2253</td>
<td>Weak</td>
</tr>
<tr>
<td>comment, etc.</td>
<td>create/save</td>
<td>Verbal</td>
<td>0.0910</td>
<td>Weak</td>
</tr>
<tr>
<td>audio, etc.</td>
<td>close/delete</td>
<td>Verbal</td>
<td>0.0005</td>
<td>Weak</td>
</tr>
</tbody>
</table>

6.3 Evaluation Metrics

Due to the absence of groundtruth labels, three unsupervised metrics are adopted for offline evaluations, i.e., BPL, NES, and DBI. To further verify the effectiveness of proposed methods, online A/B testing is also conducted and evaluated by the dropout rate and learning efficiency, which will be described later in Section 7.

- **BPL.** BPL is short for Behavior Preference Loss which is used to assess the understanding of the user behavioral preferences as in Equation (16). It is calculated on the final classification results and lower values mean better performances.
- **NES.** NES is short for Normalized Entropy Score which is adopted to evaluate the category imbalances:

\[
NES = -\frac{1}{\log_2 |L|} \sum_{j=1}^{|L|} \frac{|C_j|}{|U|} \log_2 \frac{|C_j|}{|U|} \tag{19}
\]

A lower NES value means a more skewed learning style label distribution. But extremely low NES values (i.e., NES = 0) are undesirable as the classification results will be unusable.
- **DBI.** DBI is short for the Davies-Bouldin Index [9]. It assesses the data separation state between clusters. A lower DBI indicates better separated classes. We further divide DBI with the user number to reduce the value range.

6.4 Compared Methods

Both classic clustering algorithms and deep models are adopted for comparison. As the baselines provide only clustering results, we improve them with the same label mapping rules in section 5.1.4.

- **KMeans.** The classic KMeans [2] algorithm that iteratively updates the sample clustering and centroids.
- **Gaussian Mixture Model (GMM),** GMM [24] clusters data by discovering the Gaussian distributions.
- **BIRCH.** BIRCH [32] first inserts samples into a feature tree as the leave nodes, then merges them for clustering.
- **DEC-KM.** The standard DEC model with the KMeans centroid initialization, which is our backbone model.
- **DEC-GMM.** The KMeans initialization of DEC is replaced with GMM for comparison.
- **Rule-based approach.** The proposed rule-based approach is described as in Section 4.
- **DUCK-KM.** Our proposed DUCK model with the KMeans initialization as described in Section 5.
- **DUCK-GMM.** A variant of our DUCK model where the KMeans initialization is replaced with GMM.

Detailed experimental settings are provided in Appendix A.

6.5 Overall Performance Evaluation

Table 3 exhibits the experimental performances of compared methods. From the presented results, several observations can be found.

- First, the lowest BPL scores are always achieved by our proposed methods (DUCK-KM < DUCK-GMM < Rule-based < others), which demonstrates the effectiveness of our methods in discovering and utilizing the user behavioral preferences (**Heuristic 1**). Because the pedagogy domain knowledge is neglected, both traditional (KMeans, GMM, and BIRCH) and deep (DEC-KM and DEC-GMM) baselines fail to understand the behavioral preferences and obtain high BPL scores.
- Secondly, both traditional (KMeans, GMM, and BIRCH) and deep (DEC-KM and DEC-GMM) fail to discover the distribution skewness (**Heuristic 2**) with high NES values. The rule-based approach successfully reveals the skewness on the MOOC dataset (NES=0.6136 for training and NES=0.6126 for testing) but fails on the sparse Industrial dataset (NES=0.8931 for training and NES=0.8931 for testing). However, DUCK-KM successfully understands **Heuristic 2** and achieves significantly lower NES values on both datasets. Due to different data densities, DUCK-GMM achieves the lowest NES values on the MOOC dataset but fails on the Industrial dataset.
- Thirdly, KMeans always achieves the lowest DBI scores, while the rule-based approach consistently achieves the highest DBI scores. In contrary, the DUCKs successfully balance the understanding of domain knowledge and better clustering separation, where DUCK-GMM achieves the second best DBI scores on both datasets.

Taken overall, we claim that our DUCK models perform best.

6.6 Ablation Study

The proposed DUCK model improves the DEC backbone with the behavioral preference constraint and the label distribution smoothing. To evaluate the effectiveness of the proposed components, the ablation study is conducted by removing each component and results into three sub-models, i.e., DUCK/BPC (the behavioral preference constraint is removed), DUCK/LDS (the labels distribution smoothing is removed), and DUCK/BPC/LDS (the original DEC backbone without either component). Experimental results are presented in Table 4. Due to the lack of space, only the BPL and NES scores on the MOOC dataset are reported, and the KMeans centroid initialization is adopted. As can be observed, the removal of either component leads to significant performance drops, which demonstrates the necessity and effectiveness of the proposed components. Furthermore, DUCK/LDS performs better than DUCK/BPC in NES on both training and testing sets. The reason is that the behavioral preference constraint plays the key role in understanding and utilizing domain knowledge, based on which the label distribution smoothing component further tunes the classification results.

6.7 Behavioral Preference Analysis

**Heuristic 1** provides the most important pedagogy domain knowledge in LSC tasks, i.e., the behavioral preference. To visualize how it is reflected in the classification results, histograms of the mean anchor feature values are drawn in Figure 4. Due to space limit, only...
Table 3: The overall performance evaluations, all metrics are the lower the better. The best performances are boldfaced and the second best results are underlined.

<table>
<thead>
<tr>
<th>Models</th>
<th>MOOC training</th>
<th>MOOC testing</th>
<th>Industrial training</th>
<th>Industrial testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BPL</td>
<td>NES</td>
<td>DBI</td>
<td>BPL</td>
</tr>
<tr>
<td>KMeans</td>
<td>0.7357</td>
<td>0.9093</td>
<td>1.2724</td>
<td>0.7340</td>
</tr>
<tr>
<td>GMM</td>
<td>0.6747</td>
<td>0.9952</td>
<td>1.4221</td>
<td>0.6768</td>
</tr>
<tr>
<td>BIRCH</td>
<td>0.6840</td>
<td>0.9952</td>
<td>1.4255</td>
<td>0.6820</td>
</tr>
<tr>
<td>DEC-KM</td>
<td>0.7323</td>
<td>0.9684</td>
<td>1.4420</td>
<td>0.7308</td>
</tr>
<tr>
<td>DEC-GMM</td>
<td>0.7341</td>
<td>0.9695</td>
<td>1.4331</td>
<td>0.7369</td>
</tr>
<tr>
<td>Rule-based</td>
<td>0.3440</td>
<td>0.6136</td>
<td>2.5945</td>
<td>0.3453</td>
</tr>
<tr>
<td>DUCK-KM</td>
<td>0.1487</td>
<td>0.6201</td>
<td>1.9514</td>
<td>0.1472</td>
</tr>
<tr>
<td>DUCK-GMM</td>
<td>0.2416</td>
<td>0.4793</td>
<td>1.2978</td>
<td>0.1810</td>
</tr>
</tbody>
</table>

Table 4: The ablation study results.

<table>
<thead>
<tr>
<th>Models</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BPL</td>
<td>NES</td>
</tr>
<tr>
<td>DUCK</td>
<td>0.1487</td>
<td>0.6201</td>
</tr>
<tr>
<td>DUCK/BPC</td>
<td>0.7281</td>
<td>0.9503</td>
</tr>
<tr>
<td>DUCK/LDS</td>
<td>0.3437</td>
<td>0.7767</td>
</tr>
<tr>
<td>DUCK/BPC/LDS</td>
<td>0.7323</td>
<td>0.9684</td>
</tr>
</tbody>
</table>

7 ONLINE DEPLOYMENT

As illustrated with the solid green blocks in Figure 6, the proposed LSC solutions have been deployed in the Huawei Education Center which is a widely used auxiliary teaching system in China. Learning style labels guide the system to generate personalized learning paths that accord with users’ learning preferences, thus reduces dropout and improves efficiency. In a recommended learning sequence, visual users get more visual learning materials (purple blocks) while verbal user get more text-like materials (gold blocks).

![Online deployment of the proposed LSC approaches in the learning path generation system.](https://appgallery.huawei.com/app/C101178177)

7.1 System Overview

In offline training, user behavior logs are used for feature processing and model training. The logs include user behavior events such as click, create, read, and write. We use Huawei’s self-developed log collection system to store and collect users’ online behavior data. Data processing and feature engineering are then conducted on the big data platforms. After that, the proposed models are trained with Intel Xeon 6278C CPUs (26 cores), Tesla V100’s GPU, and 240GB Host memory. The trained models are then stored online and predict the learning style labels for users in real-time.

In online serving, when a user requests a learning sequence on the APP, an acquisition request is send to server. The request is first parsed to obtain user ID and other necessary information. Based on these data, the system further generates the real-time user features, retrieves the learning profiles (e.g., learning targets and learned courses), and predicts the learning style. By taking all these information into account, a personalized learning sequence is generated which is composed of various learning materials including micro-courses, articles, videos, audio, and pictures. The
predicted learning style label has an important impact on the learning sequence generation process. Usually, a visual user’s learning sequence contains more picture-like materials, while a verbal user obtains more text-like materials as illustrated in Figure 6.

7.2 Online A/B Testing

The online A/B testing is conducted and evaluated with two vital metrics, i.e., the dropout rate (DR) and the learning efficiency (LE). DR is defined as the ratio between lost users and active users

\[ DR = \frac{LU(T_1)}{AU(T_2)} \]

where LU(T1) counts users who have left the APP for at least T1 days and AU(T2) counts all active users in the past T2 days. LE is defined as the average learning time for a new knowledge point

\[ LE = \frac{LT(T_2)}{KP(T_2)} \]

where LT(T2) summarizes the learning time and KP(T2) summarizes the number of learned knowledge points of all users within T2 days. We set T1 and T2 as 30 and 180 respectively.

Due to the long time it takes to make a full and prudent online evaluation, A/B testing of the rule-based approach starts from December 2021 and is still ongoing, while the DUCK model is still waiting for scheduling. The only difference between experimental and comparison groups is the use of learning style labels. So far, the rule-based approach achieves a 1.2% decrease on DR and a 3.5% increase on LE. Such significant improvements successfully verify the effectiveness of our work.

8 CONCLUSION AND FUTURE WORK

This paper focuses on solving the new and vital unsupervised learning style classification task in online education platforms. A rule-based approach is first provided as a naive solution, based on which a novel model named DUCK is further proposed. With a clustering objective, data representations are learned from the intrinsic feature semantics. After that, a behavioral preference constraint and a label distribution smoothing component are further devised to make better use of the pedagogy domain knowledge. Offline experimental results on both public and industrial datasets verify the effectiveness of proposed methods. Moreover, the online deployment and A/B testing results validate the value of this work. In future work, we will explore the use of self-supervised learning techniques on the LSC task.

REFERENCES

A EXPERIMENTAL SETTINGS

A.1 Implementation Details

The sklearn package is adopted to implement the KMeans, GMM, and BIRCH algorithms\(^4\). A public Tensorflow version of the standard DEC-KM\(^5\) is used, based on which we further implement the DEC-GMM, DUCK-KM, and DUCK-GMM algorithms by ourselves. For all compared methods, both datasets are randomly split into training and testing sets with a 80 to 20 ratio. After all unsupervised algorithms are trained, the same label mapping rules are used to find the final class labels as described in section 5.1.4.

A.2 Hyperparameter Settings.

For all compared methods, the number of clusters or classes is set to 3. The same data pre-processing steps are shared across all methods including feature processing, anchor feature selection, and the split of training and testing sets. Other parameter settings are selected with grid search on each dataset as listed in Table 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentile point (\mu_0)</td>
<td>50</td>
</tr>
<tr>
<td>Percentile point (\mu_1)</td>
<td>75</td>
</tr>
<tr>
<td>Classification threshold (r_0)</td>
<td>-0.5</td>
</tr>
<tr>
<td>Classification threshold (r_1)</td>
<td>0.1</td>
</tr>
<tr>
<td>KM initialize runs</td>
<td>20</td>
</tr>
<tr>
<td>KM maximum #iter</td>
<td>100</td>
</tr>
<tr>
<td>GMM covariance type</td>
<td>spherical</td>
</tr>
<tr>
<td>GMM initialize runs</td>
<td>5</td>
</tr>
<tr>
<td>GMM maximum iteration #iter</td>
<td>20</td>
</tr>
<tr>
<td>BIRCH merging threshold</td>
<td>0.05</td>
</tr>
<tr>
<td>BIRCH maximum sub-cluster #iter</td>
<td>30</td>
</tr>
<tr>
<td>DEC/DUCK-KM MLP size</td>
<td>[128]</td>
</tr>
<tr>
<td>DEC/DUCK-GMM MLP size</td>
<td>[64, 256]</td>
</tr>
<tr>
<td>DEC/DUCK-KM initialize #iter</td>
<td>5000</td>
</tr>
<tr>
<td>DEC/DUCK-KM finetune #iter</td>
<td>250</td>
</tr>
<tr>
<td>DEC/DUCK-KM initialize #iter</td>
<td>10000</td>
</tr>
<tr>
<td>DEC/DUCK-GMM initialize #iter</td>
<td>800</td>
</tr>
<tr>
<td>DEC/DUCK-GMM finetune #iter</td>
<td>10000</td>
</tr>
<tr>
<td>Loss weights, (\lambda_1, \lambda_2, \lambda_3)</td>
<td>1.0, 1.0, 1.0, 1.0, 1.0</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.00005</td>
</tr>
<tr>
<td>Batch size</td>
<td>512</td>
</tr>
</tbody>
</table>

\(^4\)https://scikit-learn.org/stable/modules/clustering.html
\(^5\)https://github.com/HaebinShin/dec-tensorflow