Learning with Feature and Distribution Evolvable Streams

Zhen-Yu Zhang [1] **Peng Zhao** [1] **Yuan Jiang** [1] **Zhi-Hua Zhou** [1]

Abstract

In many real-world applications, data are collected in the form of a stream, whose feature space can evolve over time. For instance, in the environmental monitoring task, features can be dynamically vanished or augmented due to the existence of expired old sensors and deployed new sensors. Furthermore, besides the evolvable feature space, the data distribution is usually changing in the streaming scenario. When both feature space and data distribution are evolvable, it is quite challenging to design algorithms with guarantees, particularly theoretical understandings of generalization ability. To address this difficulty, we propose a novel discrepancy measure for data with evolving feature space and data distribution, named the *evolving discrepancy*. Based on that, we present the generalization error analysis, and the theory motivates the design of a learning algorithm which is further implemented by deep neural networks. Empirical studies on synthetic data verify the rationale of our proposed discrepancy measure, and extensive experiments on real-world tasks validate the effectiveness of our algorithm.

1. Introduction

In many real-world tasks, data are usually accumulated over time and collected from open and dynamic environments, and thus they are evolving naturally. In particular, the feature space of the streaming data can evolve over time, where previous features vanish and new features appear. For instance, we deploy sensors in the ecosystem to collect data, in which the signal returned from each sensor corresponds to a feature. Due to the limited-lifespan of each sensor, we need to replace the worn-out sensors by new ones. Therefore, features corresponding to previous sensors (previous features) vanish and features corresponding to current sensors (new features) appear. A similar situation also occurs in mining high-value users in online recommendations. Due to the privacy issues, we may only trace the customers' rating record, where customers' ratings on each commodity can be regarded as a feature. Gradually, some commodities (previous features) are dropped, whereas some new ones (new features) are added, so the feature space of each new customer is changing over time.

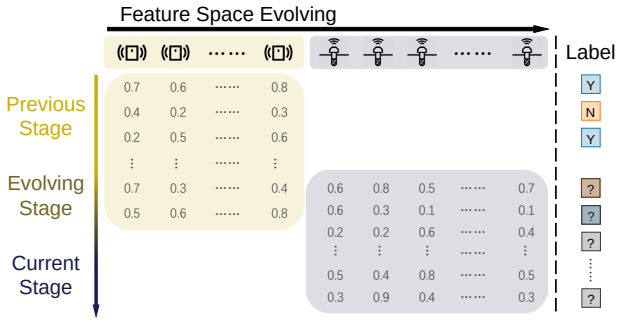


Figure 1. Illustration for Feature and Distribution Evolving Streaming Learning. In environment monitoring task, both feature space and data distribution might change in the streaming data.

To exploit historical data from previous feature space, it is crucial to bridge the gap between the current feature space and the previous one. The intuition is formulated in the pioneering work of Hou et al. (2017), whose crucial observation is that the feature space does not change arbitrarily in general; instead, there usually exists an *evolving stage* where both previous and current features are available. As shown in Figure 1, we can spread a set of new sensors *before* the old ones wear out in the environment monitoring task, because we usually know how long their battery will run out. Therefore, Hou et al. (2017) propose to establish the relationship between previous and current feature spaces by learning a mapping matrix based on the data in the evolving stage. Through mapping the current data onto previous feature space, one can exploit the historical data or classifier to help learning a new classifier in the current feature space.

Besides the evolvable feature space, the distribution of streaming data is usually evolving in non-stationary environments (Sugiyama & Kawanabe, 2012; Bifet et al., 2018). For example, in the aforementioned ecosystem monitoring

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: Yuan Jiang <jiangy@lamda.nju.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

task, patterns of the data collected from the sensors could be changing due to the climate change or other environmental non-stationarity, which results in a changing data distribution in the data stream. However, it is non-trivial to deal with the distribution change in feature space evolving streams, because the strategy of learning the mapping function between two different feature spaces is no longer reliable when distribution is changing. Even if a mapping function is learned in the evolving stage, we cannot directly apply it to the current data as the distribution has changed.

We formulate these real-world learning problems as the *Feature and Distribution Evolving Stream Learning* (FDESL), where both feature space and data distribution could be evolvable in the data streams. It is challenging to design approaches with sound theoretical guarantees, particularly understandings of the generalization ability. To deal with this difficulty, we resort to the technique of *discrepancy minimization* (Ben-David et al., 2007; Mansour et al., 2009; Cortes et al., 2019) instead of learning the mapping functions, with the purpose of designing effective algorithms with sound theoretical guarantees. However, existing discrepancy minimization approaches are not applicable because the feature space changes in our scenario. Therefore, it is desired to design a discrepancy measure across two different feature spaces, and further provide effective algorithms with sound theoretical guarantees. In this work, we achieve this goal by proposing a new discrepancy measure via exploiting the aligned (unlabeled) data in the evolving stage as a bridge. We further derive the generalization error analysis based on the proposed discrepancy measure, and the theory motivates the algorithm design.

Our Contributions. The main contributions are three-fold. First, we introduce and investigate a novel learning problem, namely, Feature and Distribution Evolving Stream Learning (FDESL), which encompasses a variety of real-world applications. Second, we define the discrepancy measure for the feature space and distribution evolvable streams to characterize the FDESL problem, called the *evolving discrepancy*. Furthermore, based on the discrepancy measure, we derive the generalization analysis. The theory guides the design of the learning algorithm, which is subsequently implemented by the deep neural networks to leverage its powerful feature representation ability. The empirical studies on synthetic data verify the rationale of our proposed discrepancy measure. Extensive experiments on real-world datasets also validate the effectiveness of our approach.

The rest of this paper is organized as follows. We first briefly review the related work in Section 2 and then present the problem formulation in Section 3. Next, we propose the theory and algorithm in Section 4, following with empirical studies on both synthetic and real-world data in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

As the data are usually collected from open and dynamic environments, it is of great importance to facilitate the learning system with capability of dealing with the environmental changes (Dietterich, 2017), which is also one of the key requirements of *learnware* (Zhou, 2016). In this paper, we focus on the feature space change and distribution change.

Feature Space Change. To deal with data streams with evolving feature space, recent studies propose to exploit the relationship between previous feature space and the current one, so that the historical data can be further leveraged. Hou et al. (2017) learn a mapping from the evolving stage to recover previous features and then ensemble two models learned from the recovered features and current ones, respectively. Hou & Zhou (2018) assume the existence of overlapping features and design a "compress-then-expand"-style algorithm to exploit the knowledge in the evolving features. Some recent works investigate the relationship in evolving features and further extend the current evolved feature space to arbitrary ones (He et al., 2019; Beyazit et al., 2019), which demonstrate encouraging results. Nevertheless, they do not consider the distribution change problem in the steaming data, and their theoretical properties of the generalization ability are generally unclear.

Distribution Change. Distribution change frequently appears in the streaming data and has drawn considerable research interest in recent years (Gama et al., 2014; Gomes et al., 2017). Essentially, there is no hope to exploit the historical data to provide a meaningful prediction if the data distribution can arbitrarily change. Therefore, it is crucial to make assumptions on distribution-changing streams. Typically, most previous works assume that the distribution of nearby data is closer to the new data, and thus plenty of approaches are proposed, e.g., the sliding window scheme (Kuncheva & Žliobaitė, 2009) and the forgetting factor mechanisms (Klinkenberg, 2004; Zhao et al., 2019). Another important category falls into the ensemble based approaches, where they adaptively add or drop base classifiers learning from historical data and dynamically adjust weights when dealing with new coming data items (Kolter & Maloof, 2005; Elwell & Polikar, 2011; Zhao et al., 2020). However, the approaches designed for distribution evolvable data streams assume that the feature space is fixed, so they cannot apply to our FDESL scenario, where the feature space is also changing over time.

We finally mention that pioneering works of discrepancy minimization are designed to minimize the divergence between two distributions on a fixed feature space, e.g., the KL-divergence in the case of KLIEP (Sugiyama et al., 2008) and the Maximum Mean Discrepancy for Kernel Mean Matching (Gretton et al., 2012). Ben-David et al. (2007) propose the generalization analysis of discrepancy minimization ap-

proaches based on the $\mathcal{H}\Delta\mathcal{H}$-divergence, which is later extended to arbitrary loss functions by Mansour et al. (2009). After that, Mohri & Medina (2012) use the $\mathcal{Y}$-divergence to provide a tighter learning guarantee. These discrepancy based generalization bounds motivate the Discrepancy Minimization (DM) algorithms (Cortes & Mohri, 2014; Cortes et al., 2019). However, we remark that these DM approaches are defined on the fixed feature space and are not appropriate for the FDESL scenario considered in this paper.

3. Problem Formulation

In this section, we formulate the problem of Feature space and Data distribution Evolving Stream Learning (FDESL).

In streaming data learning, at each time, a batch of data is received where only their features are available. We require to predict their labels before receiving the true labels. In our scenario, both feature space and data distribution of the consecutive data batches might be changing. We state the specific setting in the following.

Consider the two consecutive batches in the data stream. Let $\mathcal{X}_P \subseteq \mathbb{R}^{d_1}$ be the feature space of the previous batch of size m and $\mathcal{X}_C \subseteq \mathbb{R}^{d_2}$ be the feature space of the current batch of size n, where $d_1 \neq d_2$. We denote by $\mathcal{Y}_P = \{-1, +1\}$ the label space for the previous batch. Following the pioneering work (Hou et al., 2017), we assume the existence of *evolving data* across two consecutive batches. As shown in Figure 1, there exists a small amount of data across the two consecutive batches. By exploiting these evolving data, we can bridge the gap between two consecutive batches with different feature spaces. More specifically, we split the two batches into three stages: the previous stage, the evolving stage, and the current stage.

- Previous stage: in the previous stage, we have labeled data $S_P = \{(\mathbf{x}_{P_1}, y_{P_1}), \ldots, (\mathbf{x}_{P_{m-k}}, y_{P_{m-k}})\}$, where $(\mathbf{x}_{P_i}, y_{P_i}) \in \mathcal{X}_P \times \mathcal{Y}_P$.
- Evolving stage: in the evolving stage, the data samples across two consecutive data batches have both feature representations in $\mathcal{X}_P$ and $\mathcal{X}_C$. We denote by $S_{\widetilde{P}} = \{\mathbf{x}_{\widetilde{P}_{m-k+1}}, \ldots, \mathbf{x}_{\widetilde{P}_m}\}$ the evolving data on previous data batch and $S_{\widetilde{C}} = \{\mathbf{x}_{\widetilde{C}_1}, \ldots, \mathbf{x}_{\widetilde{C}_k}\}$ on the current data batch, where $\mathbf{x}_{\widetilde{P}_i} \in \mathcal{X}_P$ and $\mathbf{x}_{\widetilde{C}_j} \in \mathcal{X}_C$.
- Current stage: in the current stage, we have unlabeled data $S_C = \{\mathbf{x}_{C_{k+1}}, \ldots, \mathbf{x}_{C_n}\}$, where $\mathbf{x}_{C_j} \in \mathcal{X}_C$.

Notice that the evolving stage does not last for a long time, namely, we have $k \ll m$ and $k \ll n$. As in the ecosystem monitoring task, the evolving stage is just used to switch the sensors. In our formulation, we do not use the labels of $S_{\widetilde{P}}$ in the evolving stage, to avoid the potential problems caused by the non-stationary environments. That is, we exploit the aligned unlabeled data as a bridge to link the two batches with different feature spaces.

Table 1. Main Notations and Corresponding Definitions

Notation	Definition
$S_P = \{(\mathbf{x}_P, y_P)\} \in \mathbb{R}^{(m-k)\times d_1}$	Previous data in previous batch
$S_{\widetilde{P}} = \{\mathbf{x}_{\widetilde{P}}\} \in \mathbb{R}^{k\times d_1}$	Evolving data in previous batch
$S_{\widetilde{C}} = \{\mathbf{x}_{\widetilde{C}}\} \in \mathbb{R}^{k\times d_2}$	Evolving data in current batch
$S_C = \{\mathbf{x}_C\} \in \mathbb{R}^{(n-k)\times d_2}$	Current data in current batch
$g \in \mathcal{G}$	Classifier in previous feature space
$h \in \mathcal{H}$	Classifier in current feature space

Moreover, besides the evolvable feature space, the data distribution could also change in the data stream, particularly when the data are collected from open and dynamic environments. The data distribution within each stage is supposed stationary, while distribution can change across the stages. Specifically, the distribution of $S_{\widetilde{P}}$ differs from that of S_P, and the distribution of $S_{\widetilde{C}}$ differs from that of S_C. Table 1 summarizes the main notations.

For the FDESL problem, our goal is to learn a well-generalized classifier for the current data S_C. For the previous batch, suppose we are given a family of decision functions $\mathcal{G}$, in which each function $g : \mathcal{X}_P \mapsto \mathbb{R}$. While for the current mini-batch, we denote by $\mathcal{H}$ the hypothesis set, where each function $h : \mathcal{X}_C \mapsto \mathbb{R}$. We consider a loss function $\ell : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}_+$ non-negative and Lipschitz-continuous. For any hypothesis $g \in \mathcal{G}$ and the labeling function f_P on the previous stage, we denote by $R_{\mathcal{D}_P}(g, f_P)$ the expected risk and $\widehat{R}_P(g, y_P)$ the empirical risk,

$$R_{\mathcal{D}_P}(g, f_P) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_P}[\ell(g(\mathbf{x}), y)],$$

$$\widehat{R}_{S_P}(g, y_P) = \frac{1}{m}\sum_{i=1}^{m} \ell(g(\mathbf{x}_{P_i}), y_{P_i}).$$

We further denote by $\boldsymbol{\alpha}$ the weights on sample S_P, and thus the weighted empirical risk is defined as

$$\widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P) = \frac{1}{m}\sum_{i=1}^{m} \alpha_i \ell(g(\mathbf{x}_{P_i}), y_{P_i}).$$

A similar definition can be obtained for the current stage with hypothesis $h \in \mathcal{H}$ and labeling function f_C by an analogous argument. Thereby, the learning problem consists of selecting a hypothesis h with a small expected risk $R_{\mathcal{D}_C}(h, f_C)$ with respect to the current stage.

Our analysis will assume that the loss function ℓ is convex and that it further verifies the following Lipschitz-like smoothness condition (Bousquet & Elisseeff, 2002).

Definition 1 (σ-admissibility)**.** A loss function ℓ is σ-admissible with respect to the hypothesis class $\mathcal{G}$ if there exists $\sigma \in \mathbb{R}_+$ such that for any two hypothesis $g, g' \in \mathcal{G}$ and for all $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$,

$$|\ell(g(\mathbf{x}), y) - \ell(g'(\mathbf{x}), y)| \leq \sigma\,|g(\mathbf{x}) - g'(\mathbf{x})|\,.$$

We note that the admissibility property holds for most of the common loss functions, including the quadratic loss and most other loss functions where the hypothesis set and the set of output labels are bounded by some $M \in \mathbb{R}_+$: $\forall g \in \mathcal{G}, \forall \mathbf{x} \in \mathcal{X}, |g(\mathbf{x})| \leq M$ and $\forall y \in \mathcal{Y}, |y| \leq M$. We provide more examples and discussions in Section A.2 of Supplemental Materials.

4. Theory and Algorithm

In this section, we establish the theory and algorithm for learning with feature and distribution evolvable streams. Specifically, we first analyze the generalization ability of the evolving data stream, in which the key ingredient is the proposed *evolving discrepancy*. The generalization error bound further motivates the design of the Evolving Discrepancy Minimization (EDM) algorithm, which is later implemented by the deep neural networks.

4.1. Evolving Discrepancy

In the streaming data learning problem, it is crucial to exploit knowledge from historical data to help learning from current data. However, when learning with feature and distribution evolvable streams, an immediate challenge arising here is that the current data are in *different* feature space and data distribution with previous data. Therefore, it is necessary to bridge the inconsistency between current and previous data, or simply, two consecutive data batches.

As mentioned previously, in our setting, there exists an evolving stage where the instances have feature representations of both two feature spaces. We then propose the notion of *evolving discrepancy*, which is defined based on the evolving feature space. The evolving discrepancy essentially measures the discrepancy of two consecutive batches in the feature and distribution evolvable streams.

Definition 2 (Evolving Discrepancy)**.** The evolving discrepancy of two consecutive batches S_P and S_C is defined as

$$\begin{aligned}
&\mathrm{disc}_E(S_P, S_C) \\
&= \mathrm{disc}_{\mathcal{Y}}(S_{P_{\boldsymbol{\alpha}}}, S_{\widetilde{P}_{\boldsymbol{\beta}}}) + \mathrm{align}(S_{\widetilde{P}_{\boldsymbol{\beta}}}, S_{\widetilde{C}_{\boldsymbol{\beta}}}) + \mathrm{disc}_{\mathcal{Y}}(S_{\widetilde{C}_{\boldsymbol{\beta}}}, S_C) \\
&= \sup_{g\in\mathcal{G}, h\in\mathcal{H}} \Big\{ \big|\widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P) - \widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, y_{\widetilde{P}})\big| \\
&+\sigma d_1(g, h, \boldsymbol{\beta}) + \big|\widehat{R}_{S_{\widetilde{C}_{\boldsymbol{\beta}}}}(h, y_{\widetilde{C}}) - \widehat{R}_{S_C}(h, y_C)\big| \Big\}, \qquad (1)
\end{aligned}$$

where we denote by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ the bounded empirical weights over previous sample S_P, evolving sample $S_{\widetilde{C}}$, and $S_{\widetilde{P}}$; $\mathrm{align}(S_{\widetilde{P}_{\boldsymbol{\beta}}}, S_{\widetilde{C}_{\boldsymbol{\beta}}}) = \sigma d_1(g, h, \boldsymbol{\beta}) = \sigma \sum_{i=1}^{k} \beta_i |g(\mathbf{x}_{\widetilde{P}_i}) - h(\mathbf{x}_{\widetilde{C}_i})|$ is the L_1 distance of g and h with weights $\boldsymbol{\beta}$. Meanwhile, $y_{\widetilde{P}}$ and $y_{\widetilde{C}}$ are two notations of the label of the sample in the evolving stage, which are actually the same.

Remark 1. The evolving discrepancy measures the discrepancy between two consecutive data batches, where the key characteristic is that their feature spaces and data distributions can be different. In Definition 2, the first term measures the discrepancy from the previous stage to the evolving stage, the third term is the discrepancy from the evolving stage to the current stage, and the second term aligns hypotheses g and h via the aligned data in the evolving stage. Intuitively, the evolving discrepancy establishes the relationship of consecutive batches through the aligned data of the evolving stage. Notice that we do not make use of their labels in the evolving stage. Moreover, the weights of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are introduced to alleviate possible distribution changes in the evolving stage, which can be set as $\mathbf{1}$ when the distribution is stationary.

Remark 2. The evolving discrepancy generalizes the notion of $\mathcal{Y}$-discrepancy introduced by Mohri & Medina (2012) to the feature space evolvable scenarios. The $\mathcal{Y}$-discrepancy is defined on two distributions $(\mathcal{D}_P, f_P)$ and $(\mathcal{D}_Q, f_Q)$ with the *same* feature space, concretely,

$$\mathrm{disc}_{\mathcal{Y}}(\mathcal{D}_P, \mathcal{D}_Q) = \sup_{g\in\mathcal{G}} \big|R_{\mathcal{D}_P}(g, f_P) - R_{\mathcal{D}_Q}(g, f_Q)\big|,$$

where f_P and f_Q are the labeling functions. The evolving discrepancy can deal with feature space evolvable scenarios in stark contrast to the $\mathcal{Y}$-discrepancy, and will recover the $\mathcal{Y}$-discrepancy when the feature space does not change.

Based on the evolving discrepancy, we are now ready to provide a generalization error bound on the current stage in terms of the previous data and their evolving discrepancy.

Theorem 1. *Let $\mathcal{G}$ and $\mathcal{H}$ be two families of classifiers, which might be associated with different feature spaces. Suppose that loss function ℓ is L-Lipschitz and σ-admissible. Then, for any $\delta > 0$, with probability at least $1-\delta$, we have*

$$\begin{aligned}
R_{\mathcal{D}_C}(h, f_C) \leq \widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P) + \mathrm{disc}_E(S_P, S_C) \\
+ 2L\mathfrak{R}_n(\mathcal{H}) + M_C\sqrt{\frac{\log(1/\delta)}{2n}}, \qquad (2)
\end{aligned}$$

where $M_C = \sup_{\mathbf{x}_C\in\mathcal{X}, h\in\mathcal{H}} \ell(h(\mathbf{x}_C), y_C)$ and $\mathfrak{R}_n(\mathcal{H})$ is the Rademacher complexity of the function family $\mathcal{H}$.

Proof Sketch. In order to bound the expected risk of the current stage by labeled data in previous feature space and their discrepancy, we align two hypotheses with different feature spaces through *aligned data in the evolving stage* by using the σ-admissible property of loss functions.

We start from the standard Rademacher complexity based generalization error bound. Removing the terms that cannot be optimized, for any $\delta > 0$, with probability at least $1-\delta$, the expected risk in the current data can be bounded by the the following three terms: the weighted empirical risk in the evolving stage, the $\widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, y_{\widetilde{P}})$, the hypotheses alignment $|\widehat{R}_{S_{\widetilde{C}_{\boldsymbol{\beta}}}}(h, y_{\widetilde{C}}) - \widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, y_{\widetilde{P}})|$ and the discrepancy from the evolving stage to the current stage $\mathrm{disc}_{\mathcal{Y}}(S_{\widetilde{C}_{\boldsymbol{\beta}}}, S_C)$.

Since we have the labeled data in the previous stage only, we need to rewrite the weighted empirical risk $\widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, y_{\widetilde{P}})$ in the evolving stage in terms of the weighted empirical risk $\widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P)$ in the previous stage and the discrepancy from previous stage to the evolving stage $\mathrm{disc}_{\mathcal{Y}}(S_{P_{\boldsymbol{\alpha}}}, S_{\widetilde{P}_{\boldsymbol{\beta}}})$.

With the key observation that each data item in the evolving stage enjoy both feature representations and actually share the same label, we can bound the hypotheses alignment term by exploiting the σ-admissibility of loss functions, that is,

$$\left|\widehat{R}_{S_{\widetilde{C}_{\boldsymbol{\beta}}}}(h, y_{\widetilde{C}}) - \widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, y_{\widetilde{P}})\right| \leq \sigma \sum_{i=1}^{k} \beta_i \left|g(\mathbf{x}_{\widetilde{P}_i}) - h(\mathbf{x}_{\widetilde{C}_i})\right|.$$

Therefore, we confirm that the expected risk in the current stage can be upper bounded by the weighted empirical risk in the previous stage and the evolving discrepancy. We now complete the proof sketch and omitted details will be presented in Section B of Supplemental Materials. □

Remark 3. Theorem 1 exhibits that the generalization error on the current stage can be bounded by the weighted empirical risk in the previous stage and their evolving discrepancy. Note that we supply the previous data and evolving data with weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to handle distribution changes, and the weights can be set as $\mathbf{1}$ if no distribution changes occur. Therefore, Theorem 1 fulfills the gap between two consecutive batches in the feature space and distribution evolvable stream with generalization ability understandings.

Remark 4. In proving Theorem 1, it is crucial to exploit the σ-admissibility of loss functions (Bousquet & Elisseeff, 2002). The property holds for the quadratic loss function and most common loss functions with bounded hypothesis space. We present more details and verifications of the property in Section A of Supplemental Materials. By exploiting the σ-admissibility, we can eliminate the unknown labels in the evolving stage and further align hypothesis classes $\mathcal{G}$ and $\mathcal{H}$ with different feature spaces. After aligning hypothesis classes, we can thus use historical data in the previous stage even though they are from a different feature space.

To obtain a well-generalized classifier on the current data, Theorem 1 suggests choosing appropriate S_{P_α} and $S_{\widetilde{C}_{\boldsymbol{\beta}}}$ that minimize the evolving discrepancy on the right hand of (2). However, the definition of evolving discrepancy involves the labels of evolving data and current data, namely, the $y_{\widetilde{P}}(= y_{\widetilde{C}})$ and y_C, which are unavailable. To this end, we propose a variant $\mathrm{disc}'_E(S_P, S_C)$ by estimating the unknown labels by $\mathcal{G}$ and $\mathcal{H}$, defined as

$$\begin{aligned}
&\mathrm{disc}'_E(S_P, S_C) \\
=& \sup_{g,g'\in\mathcal{G},h,h'\in\mathcal{H}} \left\{\left|\widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P) - \widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, g')\right|\right. \\
&\left. +\sigma d_1(g, h, \boldsymbol{\beta}) + \left|\widehat{R}_{S_{\widetilde{C}_{\boldsymbol{\beta}}}}(h, y_g) - \widehat{R}_{S_C}(h, h')\right|\right\}.
\end{aligned} \tag{3}$$

Notice that the above definition does not require the label information of the data in the evolving stage. The following proposition demonstrates that the evolving discrepancy in (1) can be upper bounded by the variant (3).

Proposition 1. *For any hypothesis sets $\mathcal{G}$ and $\mathcal{H}$, the evolving discrepancy $\mathrm{disc}_E(S_P, S_C)$ is upper bounded by*

$$\mathrm{disc}'_E(S_P, S_C) + \sigma\big(d_1(g, f_{\widetilde{P}}, \boldsymbol{\beta}) + d_1(\mathcal{G}, f_{\widetilde{P}}, \boldsymbol{\beta}) + d_1(\mathcal{H}, f_C)\big),$$

where $d_1(\mathcal{G}, f_{\widetilde{P}}, \boldsymbol{\beta}) = \min_{g\in\mathcal{G}} \mathbb{E}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}[|g(\mathbf{x}) - f_{\widetilde{P}}(\mathbf{x})|]$ with $f_{\widetilde{P}}$ being the concept function on the evolving stage, and $d_1(\mathcal{H}, f_C)$ follows a similar definition.

Remark 5. The proposition can be proved by exploiting the property of σ-admissible loss functions again. Details are provided in Section B of Supplemental Materials. Proposition 1 shows that the evolving discrepancy can be upper bounded in terms of $\mathrm{disc}'_E(S_P, S_C)$ and other terms that cannot be optimized. The term $d_1(g, f_{\widetilde{C}}, \boldsymbol{\beta})$ measures the closeness of the learned classifier g and the unknown concept function $f_{\widetilde{C}}$. The remaining two terms measure the closeness of the hypothesis classes and concept functions, which reduce to zero when the hypothesis classes $\mathcal{G}$ and $\mathcal{H}$ contain the concept functions $f_{\widetilde{P}}$ and f_C, respectively.

Based on Proposition 1, we can now optimize the upper bound of evolving discrepancy and thereby design the learning algorithm by minimizing the right hand side of generalization error bound in (2).

4.2. Deep Neural Network Implementation

In this part, we propose the Evolving Discrepancy Minimization (EDM) algorithm implemented by the deep neural networks, which is derived from the generalization error bound in Theorem 1. We focus on the learning problem of two consecutive batches where the feature space and distribution evolution occurs, with the purpose of predicting labels of data in the current batch.

The generalization error analysis in Theorem 1 motivates the following optimization objective,

$$\min_{g\in\mathcal{G},h\in\mathcal{H}} \widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, y_P) + \mathrm{disc}'_E(S_P, S_C|g, h) \tag{4}$$

where the last term $\mathrm{disc}'_E(S_P, S_C|g, h)$ is

$$\begin{aligned}
&\sup_{g'\in\mathcal{G},h'\in\mathcal{H}} \left\{\left|\widehat{R}_{S_{P_{\boldsymbol{\alpha}}}}(g, f_P) - \widehat{R}_{S_{\widetilde{P}_{\boldsymbol{\beta}}}}(g, g')\right|\right. \\
&\qquad \left. +\sigma d_1(g, h, \boldsymbol{\beta}) + \left|\widehat{R}_{S_{\widetilde{C}_{\boldsymbol{\beta}}}}(h, y_g) - \widehat{R}_{S_C}(h, h')\right|\right\}.
\end{aligned}$$

Evidently, the above optimization problem can be regarded as a minimax game, where the min-player minimizes the generalization error while the max-player searches for the worst case of the evolving discrepancy. Therefore, we design an adversarial network to solve it and the framework

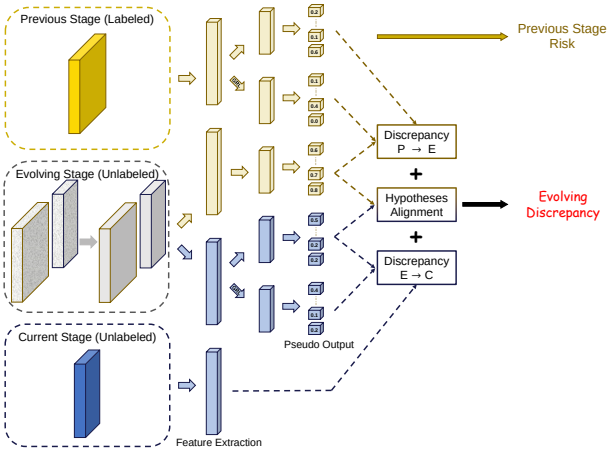


Figure 2. Framework of the Evolving Discrepancy Minimization (EDM) algorithm. We first alleviate the distribution change problem that occurs in the evolving stage, and then optimize the empirical risk and the evolving discrepancy.

is shown in Figure 2. With fixed weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ on the previous data and evolving data, we introduce the auxiliary classifiers to perform the maximum operation. Since the evolving discrepancy loss is not differentiable with respect to g and h, we minimize the evolving discrepancy loss through a gradient reversal layer (Ganin et al., 2016).

We employ the cross-entropy loss in our optimization framework, and modify it to avoid the exploding or vanishing of gradients in adversarial learning following the seminal work of Goodfellow et al. (2014). Specifically, we employ the standard cross-entropy loss for the min-player, while a slightly modified loss for the max-player. Denote by $\sigma(\cdot)$ the softmax function, for any $(\mathbf{x}, y)$, the cross-entropy loss is defined as

$$\ell(g(\mathbf{x}), y) = -\log[\sigma_y(g(\mathbf{x}))].$$

For the max-player, the modified cross-entropy loss is

$$\ell'(g(\mathbf{x}), y) = \log[1 - \sigma_y(g(\mathbf{x}))].$$

Therefore, in the adversarial networks, the objective of the max-players g' and h' can be formulated as,

$$\sup_{g' \in \mathcal{G}, h' \in \mathcal{H}} \mathrm{disc}'_{E1}(g') + \mathrm{disc}'_{E2}(h')$$

where $\mathrm{disc}'_{E1}(g')$ and $\mathrm{disc}'_{E2}(h')$ are defined as

$$\begin{aligned}\mathrm{disc}'_{E1}(g') = \sup_{g' \in \mathcal{G}} \Big\{ \Big| & \mathbb{E}_{S_P}[\boldsymbol{\alpha} \cdot \log(\sigma_{y_P}(g(\mathbf{x}_{S_P})))] \\ & + \mathbb{E}_{S_{\widetilde{P}}}[\boldsymbol{\beta} \cdot \log(1 - \sigma_{g'(\mathbf{x}_{\widetilde{P}})}(g(\mathbf{x}_{S_{\widetilde{P}}})))] \Big| \Big\};\\ \mathrm{disc}'_{E2}(h') = \sup_{h' \in \mathcal{H}} \Big\{ \Big| & \mathbb{E}_{S_{\widetilde{C}}}[\boldsymbol{\beta} \cdot \log(\sigma_{y_g}(h(\mathbf{x}_{S_{\widetilde{C}}})))] \\ & + \mathbb{E}_{S_C}[\log(1 - \sigma_{h'(\mathbf{x}_{S_C})}(h(\mathbf{x}_{S_C})))] \Big| \Big\}.\end{aligned}$$

By minimizing weighted empirical risk in the previous stage, together with the evolving discrepancy, we can therefore obtain a well-generalized classifier in the current stage. Since the optimization problem is not jointly convex in weights $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and classifiers g and h, we first find the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to alleviate the distribution change problem and then solve the minimax optimization problem. To summarize, a full implementation of the proposed EDM algorithm consists of the following two steps:

(1) learning the weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ on previous data and evolving data to alleviate distribution change problem, which can be done by using the smooth approximation algorithm of Cortes & Mohri (2014);
(2) solving the minimax optimization problem in (4) and obtain the well-generalized classifiers g and h for the evolving data and current data, respectively.

5. Experiment

In this section, we validate our EDM algorithm on synthetic data and real-world applications. Through empirical studies, we aim to answer the following three questions:

- whether the proposed evolving discrepancy implies the distance of distributions on the evolving feature spaces;
- whether the EDM algorithm shows superiority on learning with feature space and distribution evolvable streams, compared with other baseline methods;
- whether the EDM algorithm alleviates the distribution change problem and fulfills the gap between consecutive batches where the evolution occurs.

We study the first question in Section 5.1, and investigate the other two questions in Section 5.2.

5.1. Discrepancy Measure on Synthetic Data

In this part, we demonstrate that the evolving discrepancy can comprehensively measure the discrepancy of batches with evolvable feature spaces. We focus on two consecutive batches and consider a simplified scenario where only the feature space changes. Namely, we examine the accurateness of the proposed evolving discrepancy measure on two consecutive batches with different feature spaces.

When the feature spaces change, a natural idea is to learn a mapping between two feature spaces via the aligned data in the evolving stage (Hou et al., 2017). Intuitively, the mapping error should be small if these two batches are similar. We call the mapping error as the *mapping discrepancy* of two batches in the mapping-based methods, compared with our *evolving discrepancy*. We will show that the mapping scheme is not accurate, even without considering the issue of distribution change. By contrast, our proposed evolving

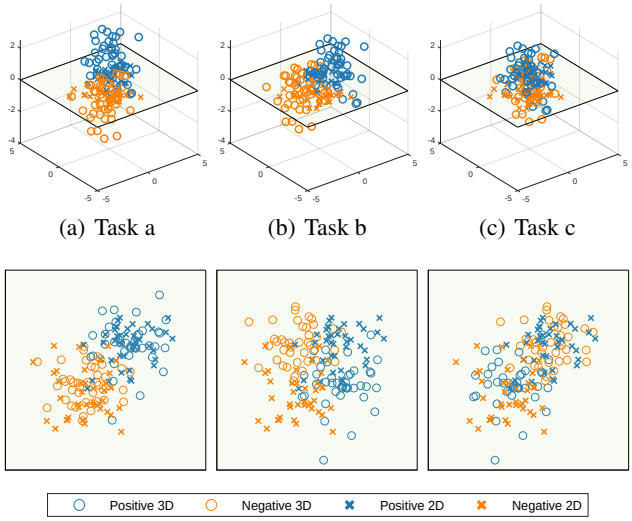


Figure 3. Previous data and the current data with different feature spaces in the 3D cube and their projections on the 2D surface of three synthetic FDESL tasks.

discrepancy offers a more accurate measure.

This empirical study is conducted on synthetic data. We sample the data of previous and current stages from 3-dim and 2-dim Gaussian distributions, respectively. The synthetic data of previous stage are generated from two class-conditional distributions, with each sample (x, y) generated from the standard 3-dim Gaussian distribution $\mathcal{N}_x$ by

$$\Pr[x|y=-1] = \mathcal{N}_x([-1,-1,-1]),$$
$$\Pr[x|y=1] = \mathcal{N}_x([1,1,1]).$$

We then generate three different batches as the current stage by 2-dim Gaussian distributions but with different mean values. Therefore, these three different data streams with previous and current stages form three FDESL tasks. For Task (a), we generate data of the current stage by

$$\Pr[x|y=-1] = \mathcal{N}_x([-1,-1]),$$
$$\Pr[x|y=1] = \mathcal{N}_x([1,1]).$$

For Task (b), we generate the current data with mean values $[-1, 1]$ for positive data and $[1, -1]$ for negative data. For Task (c), we set $[1, 1]$ and $[-1, -1]$ respectively. We plot the previous 3D data and current 2D data in Figure 3. The positive data are in blue, while the negative data are in orange. As the previous data are 3-dim, we also plot their projections in the 2D surface with the current data. Our proposal is to exploit the labeled previous data to make predictions on the unlabeled current data. As shown in Figure 3, for Task (a), the current 2D data is just the marginal distribution of the previous 3D data, which is intuitively similar and rather easy to learn. While for Task (b) and (c), the current data rather differ from the previous one.

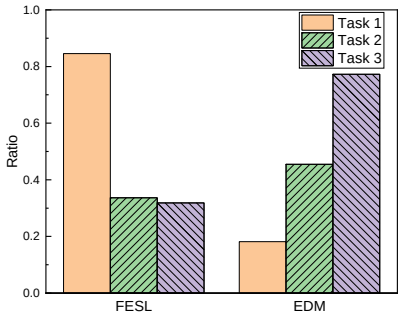


Figure 4. Relative discrepancy reported by FESL and EDM algorithms on three synthetic datasets. A lower ratio implies a higher similarity of consecutive batches returned by the algorithm.

For each task, we provide 100 previous 3D points and 80 current 2D points, and then randomly select 20 points as the evolving data. We perform the EDM algorithm and the FESL algorithm (Hou et al., 2017) on these three synthetic tasks, where both algorithms return the optimized discrepancy. All the results are normalized to $[0, 1]$. Figure 4 reports their relative discrepancies on these three tasks.

Overall, the EDM algorithm shows a much more accurate result. Specifically, Task (a) is intuitively the easiest case among three FDESL problems, because the current data is just the marginal distribution of the previous data. Figure 4 shows that the evolving discrepancy of Task (a) is significantly smaller than that of Task (b) and (c), which accords to the intuition. By contrast, the FESL algorithm reports the largest mapping error in Task (a). This synthetic study demonstrates that the EDM algorithm recovers the essential relationship and provides a more accurate characterization for the feature space evolvable problems.

5.2. Empirical Studies on Real-world Data

In this section, we first examine the EDM algorithm on various real-world applications to investigate its effectiveness in the FDESL problem. Then, we empirically verify the rationale of the evolving discrepancy and the usefulness of each component in the EDM algorithm on real-world datasets.

Global Performance Comparison

We examine the performance of the EDM algorithm on real-world scenarios. First, we conduct the empirical comparisons on the following two applications:

- RFID Dataset (Hou et al., 2017) is real-time data streams collected by the RFID technique. It contains the data collected by RFID aerial (feature) and the ground-truth location of the moving goods (label). Before the aerials expired, we will arrange new aerials beside the old ones to formulate evolving data. We split all the position index into 4 categories and thus generate their labels. As we have the time stamp and

Table 2. Performance comparisons on real-world applications. For each data stream, 10 evolutions are conducted, and the average accuracy as well as the standard deviation are presented. The best algorithm of each dataset is emphasized in bold.

Methods	RFID	A.Books	A.Movies	A.CDs
FESL	77.39 ± 2.5	70.53 ± 4.7	67.30 ± 3.6	61.79 ± 3.4
FESL+SW	82.57 ± 1.3	74.25 ± 2.6	68.66 ± 0.8	60.93 ± 1.9
FESL+FF	83.16 ± 2.0	75.92 ± 3.2	69.48 ± 1.1	63.57 ± 1.6
TSIW	91.34 ± 1.1	73.83 ± 2.1	72.61 ± 2.0	63.93 ± 0.7
EDM	$\mathbf{93.32 \pm 1.2}$	$\mathbf{77.97 \pm 5.2}$	$\mathbf{76.16 \pm 1.8}$	$\mathbf{69.47 \pm 2.5}$

the corresponding feature space of each coming data, we chronologically generate the feature space and distribution evolvable streams with the batch data size of 1000 and the evolving data size of 200.

- Amazon Dataset (McAuley et al., 2015) contains the product's quality (label) from 2006 to 2008 according to the ratings of its users (feature). We take three subsets of this dataset, which contain the data of Books, Movies, and CDs. As time goes on, some users sign out while the new users are signing up. We find some periods in which both old and new features exist to formulate the evolving data. We split the user ratings into 2 classes and thus generate the binary classification task. We also generate the FDESL tasks with the batch data size of 1000 and evolving data size of 200.

For implementations of the EDM algorithm, we set the main classifiers (min-player) and auxiliary classifiers (max-player) in the adversarial network as two 5-layer MLP with Tanh as activation functions. The model is trained by SGD with a learning rate of 0.004 and regularization weight decay 0.005. We provide omitted details of datasets and implementations in Section C of Supplemental Materials.

For these four FDESL tasks in real-world applications, we employ the FESL approach (Hou et al., 2017) as one of the baseline algorithms. Notice that the FESL method does not consider the distribution change in the streaming data, we thus further add additional rectifications for a fair comparison. Specifically, we apply the sliding window and forgetting factor mechanisms to the FESL method to alleviate the distribution change, and name them as FESL+SW and FESL+FF, respectively. For the FESL+SW method, we predict the unlabeled data only with the latest labeled data within the windows; for the FESL+FF method, we decrease the importance of previous data in an exponential rate.

Another baseline for the FDESL problem is the Two-Stage Importance Weighting (TSIW), where we train the first model on previous data with pre-trained weights $\boldsymbol{\alpha}$ and predict pseudo-labels of evolving data; then, we train a second model on evolving data with fixed weights $\boldsymbol{\beta}$. We also test this baseline in the global comparison.

Table 3. Summarization of the Reuters Multilingual Dataset.

Reuters dataset	English	French	German	Italian	Spanish
# dim after PCA	1,131	1,230	1,417	1,041	807
# samples	18,758	25,468	29,953	24,039	11,547

Table 4. Performance comparisons on the FDESL problem simulated by real-world datasets. For each streaming data, 10 evolutions are conducted, and the average accuracy as well as standard deviation are presented, and the best one is emphasized in bold.

Methods	EN-FR	FR-SP	GR-IT	IT-GR
FESL	78.51 ± 1.9	73.64 ± 2.6	75.12 ± 1.4	77.96 ± 0.9
FESL+SW	80.18 ± 1.0	73.70 ± 2.1	77.36 ± 3.3	77.90 ± 1.1
FESL+FF	79.21 ± 0.7	74.54 ± 1.5	76.85 ± 2.5	78.27 ± 1.9
TSIW	84.42 ± 1.8	79.43 ± 2.3	81.92 ± 4.4	82.30 ± 2.7
EDM	$\mathbf{86.74 \pm 0.7}$	$\mathbf{80.72 \pm 1.4}$	$\mathbf{85.40 \pm 3.9}$	$\mathbf{84.84 \pm 2.7}$

Table 2 reports the comparison results on these two real-world applications. The proposed EDM algorithm achieves the highest accuracy on all datasets, which indicates that our proposed algorithm successfully solves the FDESL problem. The proposed EDM algorithm shows its superiority over the mapping-based algorithms, as the evolving discrepancy provides a more accurate characterization for the feature space and distribution evolvable stream and we could directly minimize upper bounds of the expected risk. Furthermore, we discover that the EDM algorithm always outperforms the direct implementation (TSIW), because the evolving stage is usually small in the real-world scenario.

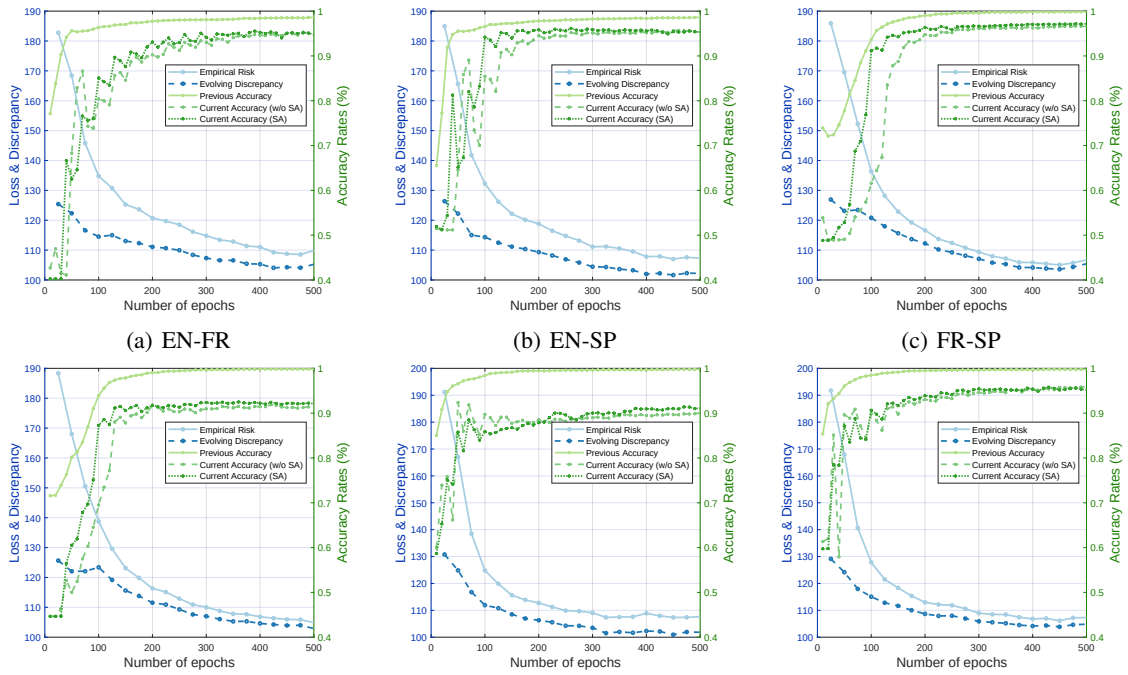
We further examine the performance of the EDM algorithm on more extensive scenarios, where the evolving stream is characterized by the textual information simulated by the Reuters Multilingual Dataset.

- Reuters multilingual dataset (Amini et al., 2009) contains about 11K articles from 6 classes in 5 languages so that we can simulate the evolving stage by various languages. As each document is translated into other languages, we treat that as the evolving data. All documents are represented by using the TF-IDF feature.

For the Reuters dataset, we perform Principal Component Analysis (PCA) based on the TF-IDF features with 60% energy preserved for each language and summarize the datasets in Table 3. The results on these simulated FDESL tasks are reported in Table 4. For these text streams, the EDM algorithm achieves the highest accuracy, which demonstrates the potential of the EDM algorithm to such real-world textual applications.

Local Effectiveness Study

In this part, we aim to demonstrate that the evolving discrepancy plays an essential role in the FDESL problem, and



(a) EN-FR (b) EN-SP (c) FR-SP

(d) FR-EN (e) SP-EN (f) SP-FR

Figure 5. Empirical risk, evolving discrepancy and accuracy on six FDESL tasks simulated by the cross-language data.

verify the efficacy of each component in the EDM algorithm.

This empirical study is conducted on the cross-language dataset (Ng et al., 2012). This classification dataset contains documents from Google with English, Chinese, and French pages, so that we can simulate the evolving batches from any two of these three languages as they share the different feature spaces. We additionally supplement the two consecutive batches by crawled data from Wikipedia to simulate the evolving data, as each article in Wikipedia has multiple language versions. Thus, the distribution of evolving data also differs from the previous and current data, which simulates the issue of distribution change.

Figure 5 reports the empirical risk, evolving discrepancy, and accuracy on the FDESL problem simulated by six pairs of cross-language data. Overall, the accuracy of current data increases with the decreasing of evolving discrepancy over all the six tasks, indicating that the minimization of the evolving discrepancy is of the essence. In the proposed EDM algorithm, we first learn the empirical weights on the previous and evolving stages by the Smooth Approximation technique (SA) and then solve the minimax optimization. As shown in Figure 5, the minimax optimization successfully obtains a well-generalized classifier, and SA helps to obtain a more satisfactory performance, which alleviates the distribution change issue in the FDESL problem. Therefore, the empirical results verify the rationale of proposed discrepancy measure and also indicate the effectiveness of each component in the EDM algorithm.

6. Conclusion

In this paper, we introduce the problem of learning with feature and distribution evolvable streams, which encompasses in a variety of real-world applications. Due to the simultaneous changes of both feature space and data distribution, it is challenging to design algorithms with sound theoretical guarantees, particularly with understandings of the generalization ability. To address this difficulty, we propose the *evolving discrepancy* to measure the discrepancy of consecutive data batches that might be of different feature spaces and data distributions. Based on the proposed discrepancy, we provide the generalization error analysis for the feature and distribution evolvable stream. The theory motivates the design of the proposed evolving discrepancy minimization algorithm, which is further implemented by deep neural networks. Empirical studies on synthetic data verify the rationale of our proposed evolving discrepancy, and extensive experiments on various real-world applications validate the effectiveness of our algorithm.

Acknowledgements

This research was supported by NSFC (61673201, 61921006) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

Authors thank Ming Jiang and Bo-Jian Hou for many insightful discussions. We are also grateful for the anonymous reviewers for their helpful comments.

References

Amini, M., Usunier, N., and Goutte, C. Learning from multiple partially observed views-an application to multilingual text categorization. In *Advances in Neural Information Processing Systems*, pp. 28–36, 2009.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144, 2007.

Beyazit, E., Alagurajah, J., and Wu, X. Online learning from data streams with varying feature spaces. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pp. 3232–3239, 2019.

Bifet, A., Gavaldà, R., Holmes, G., and Pfahringer, B. *Machine Learning for Data Streams: with Practical Examples in MOA*. MIT press, 2018.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

Cortes, C. and Mohri, M. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Cortes, C., Mohri, M., and Medina, A. M. Adaptation based on generalized discrepancy. *Journal of Machine Learning Research*, 20(1):1–30, 2019.

Dietterich, T. G. Steps toward robust artificial intelligence. *AI Magazine*, pp. 3–24, 2017.

Elwell, R. and Polikar, R. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:1–35, 2016.

Gomes, H. M., Barddal, J. P., Enembreck, F., and Bifet, A. A survey on ensemble learning for data stream classification. *ACM Computing Surveys*, 50(2):1–36, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

He, Y., Wu, B., Di Wu, E. B., Chen, S., and Wu, X. Online learning from capricious data streams: a generative approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2491–2497, 2019.

Hou, B.-J., Zhang, L., and Zhou, Z.-H. Learning with feature evolvable streams. In *Advances in Neural Information Processing Systems 30*, pp. 1417–1427, 2017.

Hou, C. and Zhou, Z.-H. One-pass learning with incremental and decremental features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40 (11):2776–2792, 2018.

Klinkenberg, R. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8 (3):281–300, 2004.

Kolter, J. Z. and Maloof, M. A. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 449–456, 2005.

Kuncheva, L. I. and Žliobaitė, I. On the window size for classification in changing environments. *Intelligent Data Analysis*, 13(6):861–872, 2009.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015.

Mohri, M. and Medina, A. M. New analysis and algorithm for learning with drifting distributions. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, pp. 124–138, 2012.

Ng, M. K., Wu, Q., and Ye, Y. Co-transfer learning via joint transition probability graph based method. In *Proceedings of the 1st international workshop on cross domain knowledge discovery in web and social network mining*, pp. 1–9. ACM, 2012.

Sugiyama, M. and Kawanabe, M. *Machine Learning in Non-stationary Environments: Introduction to Covariate Shift Adaptation*. MIT press, 2012.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, pp. 1433–1440, 2008.

Zhao, P., Wang, X., Xie, S., Guo, L., and Zhou, Z.-H. Distribution-free one-pass learning. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

Zhao, P., Cai, L.-W., and Zhou, Z.-H. Handling concept drift via model reuse. *Machine Learning*, 109(3):533–568, 2020.

Zhou, Z.-H. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.