Feature Selection in the Data Stream Based on Incremental Markov Boundary Learning

Xingyu Wu, Bingbing Jiang, Xiangyu Wang, Taiyu Ban, and Huanhuan Chen, *Senior Member, IEEE*

***Abstract*— Recent years have witnessed the proliferation of techniques for streaming data mining to meet the demands of many real-time systems, where high-dimensional streaming data are generated at high speed, increasing the burden on both hardware and software. Some feature selection algorithms for streaming data are proposed to tackle this issue. However, these algorithms do not consider the distribution shift due to nonstationary scenarios, leading to performance degradation when the underlying distribution changes in the data stream. To solve this problem, this article investigates feature selection in streaming data through incremental Markov boundary (MB) learning and proposes a novel algorithm. Different from existing algorithms focusing on prediction performance on off-line data, the MB is learned by analyzing conditional dependence/independence in data, which uncovers the underlying mechanism and is naturally more robust against the distribution shift. To learn MB in the data stream, the proposal transforms the learned information in previous data blocks to prior knowledge and employs them to assist MB discovery in current data blocks, where the likelihood of distribution shift and reliability of conditional independence test are monitored to avoid the negative impact from invalid prior information. Extensive experiments on synthetic and real-world datasets demonstrate the superiority of the proposed algorithm.**

***Index Terms*— Distribution shift, feature selection, Markov blanket, Markov boundary (MB), prior knowledge, streaming data.**

NOMENCLATURE

A. Variable Notations

$\mathbb{G}$	Directed acyclic graph (DAG).
$\mathbb{P}$	Joint probability distribution.
$\mathbb{P}_t$	Joint probability distribution of data stream in the tth state.
$\mathbb{D}$	Dataset, $\mathbb{D} = \{\mathbb{D}_1 \cup \mathbb{D}_2 \cup \cdots \cup \mathbb{D}_k \cup \cdots\}$.
$X, Y, \cdots$	Uppercase letters denote random variables.
$\boldsymbol{X}$	Set of all random variables, $\boldsymbol{X} = \{X_1 \cup X_2 \cup \cdots \cup X_n\}$.
$\boldsymbol{x}^i$	ith instances, $\boldsymbol{x}^i = \{{x_1}^i \cup {x_2}^i \cup \cdots \cup {x_n}^i\}$.
T	Target variable, $T \in \boldsymbol{X}$.
t^i	Value of target T in the ith instances.
$\mathbb{R}_X$	Value domain of X.
$\boldsymbol{Z} \cdots$	Uppercase bold letters denote variable subsets, $\boldsymbol{Z} \subset \boldsymbol{X}$.
α	Parameter set determining the mapping between $\boldsymbol{X}$ and MB.
β	Parameter set in the learner.
α_j^i	jth parameter in α after inputting the ith data block.
$\boldsymbol{X}_\alpha$	Subset of $\boldsymbol{X}$ determined by α.
$\boldsymbol{X}_{\hat{\alpha}}$	Subset of unselected features, i.e., $\boldsymbol{X} - \boldsymbol{X}_\alpha$.
λ_j	Likelihood that X_j is included in the MB.
μ_j	Relative possibility of the random event $X_j \in \boldsymbol{MB}$.
α_j^t	jth parameter in α after the tth iteration.
μ_j^t	Relative possibility of $X_j \in \boldsymbol{MB}$ after the tth iteration.
p	Prior knowledge term.
X^*	Selected features in an iteration.
σ	Scale parameter to determine the value of μ_j^t.
s	Average sample size on each degree of freedom in the CI test.
f_τ	Freedom degree of χ^2 statistic when X_τ is selected or removed.

B. Function Notations

$X \perp Y \vert \boldsymbol{Z}$	X and Y are conditionally dependent given $\boldsymbol{Z}$.
$X \perp\!\!\!\perp Y \vert \boldsymbol{Z}$	X and Y are conditionally independent given $\boldsymbol{Z}$.
$\mathbb{E}[\cdot]$	Mathematical expectation.
$I(X, Y \vert \boldsymbol{Z})$	Mutual information between X and Y conditioned on $\boldsymbol{Z}$.
$D_{\mathrm{KL}}[\mathbb{P}_1 \vert\vert \mathbb{P}_2]$	KL-divergence between distributions $\mathbb{P}_1$ and $\mathbb{P}_2$.
$H[\cdot]$	Information entropy.
$G^2(X, Y \vert \boldsymbol{Z})$	G^2 statistics between X and Y conditioned on $\boldsymbol{Z}$.
$\chi^2(f)$	χ^2 statistics with the degree of freedom f.
$\phi(\cdot)$	Optimization function.
$\vert \cdot \vert$	Number of items in a set.

Manuscript received 14 November 2021; revised 30 June 2022 and 5 November 2022; accepted 22 February 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0111700; in part by the National Nature Science Foundation of China under Grant 62137002, Grant 62176245, and Grant 62006065; in part by the Key Research and Development Program of Anhui Province under Grant 202104a05020011; in part by the Key Science and Technology Special Project of Anhui Province under Grant 202103a07020002; and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Huanhuan Chen.)*

Xingyu Wu, Taiyu Ban, and Huanhuan Chen are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: xingyuwu@mail.ustc.edu.cn; banty@mail.ustc.edu.cn; hchen@ustc.edu.cn).

Bingbing Jiang is with the School of Information Science and Engineering, Hangzhou Normal University, Hangzhou 311121, China (e-mail: jiangbb@hznu.edu.cn).

Xiangyu Wang is with the School of Data Science, University of Science and Technology of China, Hefei 230027, China (e-mail: sa312@mail.ustc.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3249767.

Digital Object Identifier 10.1109/TNNLS.2023.3249767

I. Introduction

MANY real-world applications are pouring data at an astonishing rate, where the high-speed generated instances bring many challenges in order to meet the demands of real-time analyses [1], [2]. For example, in many online systems (such as transportation systems [3], [4] and financial markets [5], [6], [7]), training data usually arrive in an infinite sequence. These high-dimensional streaming data significantly increase the computational burden on both storage memory and real-time data mining techniques [8], leading to performance degradation of learning algorithms. Previous research [8], [9] has indicated that the most discriminative information is carried by a subset of relevant features. This presents an interesting research topic: feature selection in the data stream.

Current feature selection algorithms for streaming data [11], [12], [13], [14] receive sequential data instances or data blocks as input and select a fixed number of features with a linear classification model. Unfortunately, these algorithms determine whether a feature should be selected according to its corresponding weight parameter in the learner, instead of analyzing the underlying mechanism $\mathbb{P}(T|\boldsymbol{X})$.[1] As a result, the predictability of selected features determined by specific learners and instances cannot be generalized to other situations, especially in streaming data with changeable underlying distributions, namely, distribution shift[2] [16], [17]. More concretely, three problems motivate this research.

1) Changes in $\mathbb{P}(T|\boldsymbol{X})$ [15], [18] make the information from previous data invalid, and the actual optimal feature subset changes correspondingly. Existing algorithms do not detect the concept shift but directly trust the information from previous data blocks, leading to the selected features only applicable in historical data but gradually invalid in the subsequent data stream.
2) When the feature distribution $\mathbb{P}(\boldsymbol{X})$ and target distribution $\mathbb{P}(T)$ change, but $\mathbb{P}(\boldsymbol{X}|T)$ remains stable, although the actual optimal feature subset does not change theoretically, the changed marginal distribution induces a different model learned for feature selection and, thereby, still has a negative impact on the selected feature subset, leading to unstable performance.
3) Another negative effect of ignoring the underlying mechanism is that existing algorithms have to predetermine the number of selected features, while, in the real-world online system with dynamically changing data streams, it is impractical and time-consuming to continuously adjust parameters to obtain the optimal scale of relevant features.

Given the deficiencies of existing methods, this article investigates feature selection in the data stream through incremental Markov boundary (MB)[3] learning [19] of the target, which has a prominent property: Given the MB of the target, all other features are conditionally independent of the target [10], as shown in Fig. 1(a). Hence, the MB discovery process determines the feature subset by analyzing the conditional dependence and independence in $\mathbb{P}(T|\boldsymbol{X})$ [20], [21], [22] and, thus, is naturally more robust against covariate shift and prior probability shift. Due to the revealed underlying distribution, MB-based algorithms can automatically determine the number of selected features [23], [24]. Nevertheless, to the best of our knowledge, few efforts have been made to MB discovery in streaming data. The main challenge is how to take the learned information on previous data blocks into account in the learning process on the current data block, as shown in Fig. 1(b). Specifically, unreliable results might be obtained if we simply pool the MB learned in different data blocks together since the possible distribution shift makes MB learned in previous data lose efficacy. While continuing to search the MB based on these early results puts conditional independence (CI) tests in adventure due to large conditioning sets, we have to extend on existing MB learning framework so that the historical information can be reasonably and spontaneously implemented into it, which is the theoretical difficulty in this article.

To address these problems, we first study the interface of external information in the traditional MB discovery scheme in Section II so that the knowledge in the previous data blocks can benefit the MB discovery in the current data block. Through the interface, the learned information in historical data can be taken as prior knowledge and incorporated into the MB discovery framework, as shown in Fig. 1(c). Based on this strategy, we propose an MB discovery algorithm for streaming data (MBSD) in Section III to meet two practical requirements for robust feature selection in data streams, that is, the historical data need not be stored, and the negative effects of distribution shift can be controlled. Specifically, MBSD is equipped with a prior term to consider the historical information, which could inherit effective knowledge from previous data blocks and then play a part in the subsequent MB discovery. Moreover, by monitoring the likelihood of concept shift and reliability of the CI test, MBSD adjusts the weight parameter of the prior term to control the negative influence of ineffective information from previous results. For the MB property vantage that is naturally immunized from the covariate shift and prior probability shift, MBSD is more robust against other methods in the data stream with distribution shift, which is validated in Section IV.

Compared with the state-of-the-art feature selection algorithms [11], [12], [13], [14] for the data stream, the proposed MBSD has at least three innovative practical benefits: 1) MBSD can automatically determine the number of selected features so that it can omit the extra step of tuning parameters to determine the size of the selected features; 2) MBSD selects features by mining the underlying mechanism behind instances, and thus, it can remain stable and superior under covariate shift and prior probability shift; and 3) by monitoring the concept drift to control the impact of the prior term, MBSD could perceive the emerging concept drift more timely, whose performance can also quickly recover to the level before the

[1] $\boldsymbol{X}$ and T denote the feature set and the target (class attribute).

[2] Distribution shift appears when the joint distributions in two data blocks are different. There are three types of shifts most commonly present: **covariate shift** [on $\mathbb{P}(\boldsymbol{X})$], **prior probability shift** [on $\mathbb{P}(T)$], and **concept shift** (on $\mathbb{P}(T|\boldsymbol{X})$ and $\mathbb{P}(\boldsymbol{X}|T)$, a.k.a., **concept drift** in the literature) [15].

[3] Refer to Definition 2 for a detailed concept.

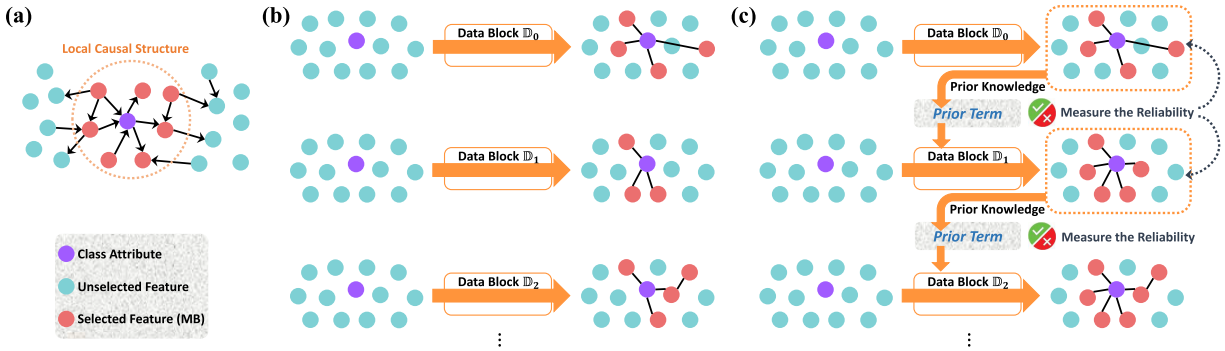


Fig. 1. Illustration of the idea in this article. (a) Definition of the MB, which satisfies that all other features should be independent of the target conditioned on the MB of the target. For example, in a faithful Causal network, the MB of a target includes its parents, children, and spouses [10], which demonstrates the local causal structure around the target. (b) Different distributions in different data blocks may induce distinct discovered MB, while existing MB learning schemes can neither directly consider the information in the previous data block nor provide an interface so that the information in the previous data block could influence the MB discovery in the current data block. (c) Proposal in this article provides an interface where the results from the previous data blocks are transformed into prior knowledge and employs them to assist MB discovery in the subsequent data blocks. The prior term is the port for obtaining prior knowledge from historical data in the MB discovery process, which could also measure the reliability of the prior knowledge so that incorrect results could have a chance to be corrected.

drift occurs under concept drift. Extensive empirical studies have demonstrated these superiorities.

The main contributions are summarized as follows.

1) We theoretically derive the form of prior knowledge in the MB discovery framework, based on which the external information from historical data could be integrated into the prior term in the MB search scheme. This contribution can also promote the usage of other information sources (e.g., domain knowledge) to extend the MB discovery method.
2) Based on the extended framework, a novel feature selection algorithm for streaming data is proposed, to take previous results as prior and control their impact by monitoring the concept drift and CI tests, which is robust in different types of data shift.

The remainder of this article is organized as follows. Section II derives the form of the prior term, and Section III introduces the usage of the prior term in the data stream and proposes a novel feature selection method MBSD for streaming data. The experimental results and analyses of the MBSD are presented in Section IV. Finally, Section V concludes this article and describes some possible future work directions.

II. Background

We first provide the commonly used notations of variables and functions in this article in the Nomenclature, respectively. Based on these notations, the basic definitions and related work are present in this section. In the following, the closely related research area, feature selection, MB learning, and streaming data feature selection are discussed, whose background and state-of-the-art algorithms are introduced.

A. Feature Selection

Feature selection is a commonly used dimensionality reduction technique [8], [9], [25], which could remove some redundant and irrelevant features to obtain a lower dimensional representation of data. Traditional feature selection methods can be roughly divided into three types, including filter, wrapper, and embedded approaches. Extensive methods have been proposed to consistently improve the accuracy and time efficiency of feature selection for common supervised learning, while recent years have witnessed the proliferation of feature selection for special purposes, such as imbalanced learning [26], evolutionary learning [27], [28], [29], [30], multilabel/multiview learning [31], [32], weakly supervised learning [33], [34], and so on. In this article, we focus on feature selection research in streaming data.

We first distinguish between two related topics, i.e., feature selection for streaming data [11] and for streaming features [35]. The research on streaming features assumes that the features arrive sequentially, but the number of training instances is fixed. While streaming data, a more natural scenario in real-world applications means that the training data arrive sequentially in form of data instances or data blocks, in which the values of all features are available.

The first feature selection algorithm for streaming data is proposed by Wang et al. [11], called the online feature selection (OFS) algorithm. OFS employs an online learner to choose a small and fixed number of features. Second-order OFS (SOFS) [12] improves the effectiveness by exploiting second-order information to compute the confidence weights, whose values determine the importance of each feature. The Truncation-based Adaptive Regularized Dual Averaging algorithm (B-ARDA) and the Adaptive Mirror Descent algorithm (B-AMD) are proposed by Zhai et al. [13], which consider the magnitude of feature values in the current predictor and their frequency in the history of predictions. Different from the aforementioned three algorithms taking each instance as input, the adaptive sparse CW (ASCW) algorithm [14] is an online-batch algorithm and takes data blocks as input, which could solve the class imbalance in OFS.

However, all of these algorithms fail to consider the data shift in the data stream, which is an important challenge for streaming data mining.

B. Data Shift in Data Stream

Data shift is a challenging issue in streaming data, where the joint probability distribution of different data windows is changeable. The formal definition is given as follows.

Definition 1 ((Data Shift) [18]): Data shift appears when the joint distributions in two data blocks are different, that is, when $\mathbb{P}_{\mathbb{D}_1}(\mathbf{X}, T) \neq \mathbb{P}_{\mathbb{D}_2}(\mathbf{X}, T)$.

Concretely, there are three types of data shift [18]: 1) covariate shift, where $\mathbb{P}_{\mathbb{D}_1}(T|\boldsymbol{X}) = \mathbb{P}_{\mathbb{D}_2}(T|\boldsymbol{X})$ but $\mathbb{P}_{\mathbb{D}_1}(\boldsymbol{X}) \neq \mathbb{P}_{\mathbb{D}_2}(\boldsymbol{X})$; 2) prior probability shift, where $\mathbb{P}_{\mathbb{D}_1}(T|\boldsymbol{X}) = \mathbb{P}_{\mathbb{D}_2}(T|\boldsymbol{X})$ but $\mathbb{P}_{\mathbb{D}_1}(T) \neq \mathbb{P}_{\mathbb{D}_2}(T)$; and 3) concept shift, where $\mathbb{P}_{\mathbb{D}_1}(T|\boldsymbol{X}) \neq \mathbb{P}_{\mathbb{D}_2}(T|\boldsymbol{X})$ but $\mathbb{P}_{\mathbb{D}_1}(\boldsymbol{X}) = \mathbb{P}_{\mathbb{D}_2}(\boldsymbol{X})$ for $\boldsymbol{X} \to Y$ problem. The two most common causes of data shift are: 1) sample selection bias and 2) nonstationary environments [18]. These causes more easily occur in the data stream when the training environment is changeable due to temporal or spatial changes [15]. To solve this problem, this article investigates the feature selection method in streaming data based on MB discovery. MB is naturally more robust against covariate shift and prior probability shift [36], and thus, only the concept shift needs to be considered in the algorithmic design. Some existing concept shift detection techniques [37], [38], [39], [40], [41] can be used to determine the parameter in our proposed algorithm, which is detailed in Section IV.

C. MB and MB Learning

We first provide the statistical concept of MB and introduce some classic MB learning methods. MB is defined based on conditional (in)dependence relationships [19].

Definition 2 (Markov Blanket and MB) [19]: The Markov blanket $\mathbf{Mb}$ of target T is a subset of $\mathbf{X}$ satisfying the condition: $\forall X \in \mathbf{X} - \mathbf{Mb}$, $X \perp T|\mathbf{Mb}$ in the joint probability distribution $\mathbb{P}$. The Markov boundary $\mathbf{MB}$ of T is the Markov blanket of T satisfying: $\forall \mathbf{Z} \subset \mathbf{MB}$, $\mathbf{Z}$ is not a Markov blanket of T.

According to the information theory, $I(\boldsymbol{MB}, T|\boldsymbol{Z}) = I(\boldsymbol{X}, T|\boldsymbol{Z})$ for $\forall \boldsymbol{Z}$, which indicates that the MB set carries all predictive information about the target T. Hence, for the feature selection task, the MB set of a class attribute has been proven to be the optimal feature subset under the faithfulness assumption [42]. Data sampled from standard Bayesian networks are usually used to test the effectiveness of an MB discovery algorithm since the MB of a target is its parents, children, and spouses in a faithful Bayesian network. Therefore, in the experiments of this article, we use data from standard Bayesian networks to validate the superiority of the proposed methods against existing MB discovery methods.

Since the unique MB of the class attribute could be directly used as the solution for the feature selection problem, MB has an extensive application prospect for feature selection [43], a.k.a., causal feature selection. Yu et al. theoretically provide a unified view of causal and noncausal feature selection methods and first fill in the gap in the research of the relation between the two types of methods [44]. Existing algorithms can be broadly grouped into two categories, i.e., simultaneous learning methods and divide-and-conquer learning methods. Simultaneous approaches use the concept in Definition 2 to learn MB, without distinguishing the parent–child variables and spouse variables, which are time-efficient but require the number of samples to be exponential to the size of the MB, such as IAMB [20] and FastIAMB [45]. Divide-and-conquer approaches discover the parent–child and spouse sets, respectively, which effectively improves the accuracy with a reasonable time cost, such as STMB [21], BAMB [46], and EEMB [47]. In this article, we extend the simultaneous learning framework with an added prior term to record the historical information in the data stream, and the MB can be obtained by simultaneous learning according to Definition 2

$$\boldsymbol{MB} = \underset{\mathbf{Z} \subset \boldsymbol{X}}{\arg\min}\ I(T, \boldsymbol{X} \mid \boldsymbol{Z}) \quad \text{with minimal } |\boldsymbol{Z}|. \tag{1}$$

To select relevant features, MB learning algorithms employ a stepwise process to solve the optimization problem in (1), including the forward and backward phases [48]. The forward phase adds the most possible feature X^* to the selected subset, while the backward phase removes the variable violating the MB concept from the selected subset. Let the discrete random variable vector α^t ($\alpha_i^t \in \{0, 1\}$) denotes the result obtained in the tth backward step, where $\alpha_i^t = 1$ means that the ith feature is selected. Then, in the tth forward step, the selected feature X^* should be found as follows:

$$X^* = \underset{X^* \in \boldsymbol{X} - \boldsymbol{X}_{\alpha^{t-1}}}{\arg\max}\ I(T, X^* | \boldsymbol{X}_{\alpha^{t-1}}). \tag{2}$$

Similarly, the deleted feature in the tth backward step should be found as follows:

$$X^* = \underset{X^* \in \boldsymbol{X}_{\alpha^{t-1}}}{\arg\min}\ I(T, X^* | \boldsymbol{X}_{\alpha^{t-1}} - \{X^*\}). \tag{3}$$

In this article, we design a novel method to implement the prior knowledge term into this framework so that the information in the previous data stream could be inherited to facilitate the feature selection task in current data blocks.

III. Construct the Prior Term for MB Discovery

Suppose that we want to approximate the true distribution $\mathbb{P}$ with an estimated distribution $\mathbb{P}_e$ from $\mathbb{D}_k$ through an MB discovery process and a classifier predicting class attribute T trained with features in the MB set. Then, let parameter set α determine the mapping between feature set $\boldsymbol{X}$ and the discovered MB, and β denotes the parameter set in the learner. Since the instances in $\mathbb{D}_k$ are seems to be independent, a joint likelihood, shown as follows, should be maximized in the standard way to obtain the best parameter set $\{\alpha, \beta\}$:

$$\mathcal{L}(\mathbb{D}_k, \alpha, \beta) = \mathbb{P}(\alpha, \beta) \prod_{i=1}^{|\mathbb{D}_k|} \mathbb{P}_e\left(t^i \mid \mathbf{x}^i, \alpha, \beta\right) \mathbb{P}_e\left(\mathbf{x}^i\right). \tag{4}$$

To simplify (4), we transform (4) as its log-likelihood

$$\log \mathcal{L}(\mathbb{D}_k, \alpha, \beta) = \sum_{i=1}^{|\mathbb{D}_k|} \log \mathbb{P}_e\left(t^i \mid \mathbf{x}^i, \alpha, \beta\right) + \sum_{i=1}^{|\mathbb{D}_k|} \log \mathbb{P}_e\left(\mathbf{x}^i\right) + \log \mathbb{P}(\alpha, \beta). \tag{5}$$

The second term in (5) entirely depends on the training samples, which is an irrelative term and could be removed.

Then, the problem can be converted to a minimizing function represented as follows:

$$\min_{\alpha,\beta} \underbrace{-\frac{1}{|\mathbb{D}_k|}\sum_{i=1}^{|\mathbb{D}_k|}\log \mathbb{P}_e\left(t^i \mid \mathbf{x}^i,\alpha,\beta\right)}_{\text{Term1}} \underbrace{-\frac{1}{|\mathbb{D}_k|}\log \mathbb{P}(\alpha,\beta)}_{\text{Term2}}. \quad (6)$$

To select the MB variables, (6) should be minimized to the optimal parameter set $\{\alpha, \beta\}$, in which Term1 and Term2 are analyzed in the following.

A. Analyze the Term1 in (6)

In fact, the first term can be decomposed to three explainable items by two introduced intermediate distributions $\mathbb{P}(t^i \mid \mathbf{x}^i)$ and $\mathbb{P}(t^i \mid \mathbf{x}^i, \alpha)$

$$\text{Term1} = \frac{1}{|\mathbb{D}_k|}\left(-\sum_{i=1}^{|\mathbb{D}_k|}\log\frac{\mathbb{P}_e\left(t^i \mid \mathbf{x}^i,\alpha,\beta\right)}{\mathbb{P}(t^i \mid \mathbf{x}^i,\alpha)} - \sum_{i=1}^{|\mathbb{D}_k|}\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i,\alpha\right)}{\mathbb{P}(t^i \mid \mathbf{x}^i)} - \sum_{i=1}^{|\mathbb{D}_k|}\log\mathbb{P}\left(t^i \mid \mathbf{x}^i\right)\right) \quad (7)$$

where $\mathbb{P}(t^i \mid \mathbf{x}^i)$ denotes the posterior probability of the target given the entire feature set and $\mathbb{P}(t^i \mid \mathbf{x}^i, \alpha)$ denotes the posterior probability of target given a feature subset selected with α. Therefore, the first term is the log-likelihood of the ratio between the estimated posterior distribution and the true posterior distribution given the feature subset determined by α, which evaluates the effectiveness of the learner based on the selected features. The closer its value is to 0, the closer the predicted distribution is to the real distribution. The second term is the log-likelihood of the ratio between the true posterior distribution given the entire feature set and the true posterior distribution given the feature subset determined by α, which evaluates the effectiveness of the feature selection process. The closer its value is to 0, the better performance the feature subset has. Through the decomposition in (7), the learning process and MB discovery process can be separated. Moreover, each term in (7) could be transformed into an explicable statistic according to the finite sample approximation [49]

$$\text{Term1} \approx \mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i,\alpha\right)}{\mathbb{P}_e(t^i \mid \mathbf{x}^i,\alpha,\beta)}\right] + \mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i\right)}{\mathbb{P}(t^i \mid \mathbf{x}^i,\alpha)}\right] + \mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{1}{\mathbb{P}(t^i \mid \mathbf{x}^i)}\right] \quad (8)$$

in which the expectation form is obtained by averaging over all instances in $\mathbb{D}_k$. In this way, each part of Term1 could be explained.

According to the definition of KL-divergence, we can denote the second term as the KL-divergence [50] between $\mathbb{P}(t^i \mid \mathbf{x}^i)$ and $\mathbb{P}(t^i \mid \mathbf{x}^i, \alpha)$, which could be further expanded based on the different values of $\boldsymbol{X}$ and T as follows:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i\right)}{\mathbb{P}(t^i \mid \mathbf{x}^i,\alpha)}\right] &= D_{\mathrm{KL}}\left[\mathbb{P}\left(t^i \mid \mathbf{x}^i\right)||\mathbb{P}\left(t^i \mid \mathbf{x}^i,\alpha\right)\right] \\ &= \sum_{\boldsymbol{\chi}\in\mathbb{R}_{\boldsymbol{X}},\tau\in\mathbb{R}_T} p(\boldsymbol{\chi},\tau)\log\frac{p(\tau \mid \boldsymbol{\chi})}{p(\tau \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})}\end{aligned} \quad (9)$$

where D_{KL} is the symbol of KL-divergence, and $\mathbb{R}_{\boldsymbol{X}}$ and $\mathbb{R}_T$ represent the value domain of $\boldsymbol{X}$ and T, respectively. $\boldsymbol{\chi}_{\boldsymbol{\alpha}}$ is a value domain slice of feature subset determined by a parameter α. Using $\boldsymbol{X}_\alpha$ and $\boldsymbol{X}_{\hat{\alpha}}$ to denote the selected and unselected variable subsets of $\boldsymbol{X} = \boldsymbol{X}_\alpha \cup \boldsymbol{X}_{\hat{\alpha}}$, then the second term in (8) is transformed to the mutual information [50] between T and $\boldsymbol{X}_{\hat{\alpha}}$ conditioned on $\boldsymbol{X}_\alpha$ by further adding an auxiliary item $p(\boldsymbol{\chi}_{\hat{\boldsymbol{\alpha}}} \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})$ to both the numerator and the denominator in (9)

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i\right)}{\mathbb{P}(t^i \mid \mathbf{x}^i,\alpha)}\right] \\ &= \sum_{\boldsymbol{\chi}\in\mathbb{R}_{\boldsymbol{X}},\tau\in\mathbb{R}_T} p(\boldsymbol{\chi},\tau)\log\frac{p(\tau \mid \boldsymbol{\chi})}{p(\tau \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})}\frac{p(\boldsymbol{\chi}_{\hat{\boldsymbol{\alpha}}} \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})}{p(\boldsymbol{\chi}_{\hat{\boldsymbol{\alpha}}} \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})} \\ &= \sum_{\boldsymbol{\chi}\in\mathbb{R}_{\boldsymbol{X}},\tau\in\mathbb{R}_T} p(\boldsymbol{\chi},\tau)\log\frac{p(\tau,\boldsymbol{\chi}_{\hat{\boldsymbol{\alpha}}} \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})}{p(\tau \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})p(\boldsymbol{\chi}_{\hat{\boldsymbol{\alpha}}} \mid \boldsymbol{\chi}_{\boldsymbol{\alpha}})} \\ &= I(T,\boldsymbol{X}_{\hat{\alpha}}|\boldsymbol{X}_\alpha).\end{aligned} \quad (10)$$

Therefore, given the optimal $\boldsymbol{X}_\alpha$, the other variables are conditionally independent of the target, which is consistent with the definition of MB.

Note that the third term in (8) is actually the information entropy [50] of T conditioned on $\boldsymbol{X}$ since

$$H(T \mid \boldsymbol{X}) = \mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{1}{\mathbb{P}(t^i \mid \mathbf{x}^i)}\right]. \quad (11)$$

The information entropy measures the uncertainty of the class attribute conditioned on the existing dataset. It is independent of the two parameters α and β so that it could be ignored when the optimal problem is solved.

Substitute (10) and (11) into (8); thus,

$$\text{Term1} \approx \mathbb{E}_{\boldsymbol{X},T}\left[\log\frac{\mathbb{P}\left(t^i \mid \mathbf{x}^i,\alpha\right)}{\mathbb{P}_e(t^i \mid \mathbf{x}^i,\alpha,\beta)}\right] + I(T,\boldsymbol{X}_{\hat{\alpha}}|\boldsymbol{X}_\alpha) + H(T \mid \boldsymbol{X}). \quad (12)$$

As for the first term in (12), it could be ignored when we assume that the MB discovery process and learning process are divisible, which is a basic assumption for filter approaches and is duly acceptable in this problem. When the optimal α is considered, there always exists a corresponding β making (8) term minimal since MB discovery algorithms are learner-independent.

B. Analyze the Term2 in (6)

According to the analysis above, we can conclude that Term1 is mainly related to the current data block $\mathbb{D}_k$, without consideration of the information from the previous data blocks. If there does not exist any effective prior knowledge, then the objective is to minimize the $I(T,\boldsymbol{X}_{\hat{\alpha}}|\boldsymbol{X}_\alpha)$ according to (12),

which is just what existing traditional batch-learning methods do. However, abundant prior knowledge from the previous data blocks could facilitate the mining task in streaming data, which is mainly included in Term2.

Since α and β are independent of each other, Term2 could be rewritten as $-(1/|\mathbb{D}_k|)\log\mathbb{P}(\alpha)\mathbb{P}(\beta)$, where the two distributions denote the prior knowledge of these two parameters, respectively. In the streaming scenarios, the prior knowledge could be considered as the function of the results from previous data blocks, i.e.,

$$\text{Term2} \approx f\left(\{\alpha^1, \alpha^2, \ldots, \alpha^{i-1}\}\right) + g\left(\{\beta^1, \beta^2, \ldots, \beta^{i-1}\}\right) \tag{13}$$

where f and g represent the function to calculate the prior term from results in the previous data block, and $\alpha^\sharp$ and $\beta^\sharp$ represent the results from the $\sharp$th data block. To consider the optimal solution of MB discovery, we can ignore the term $g(\{\beta^1, \beta^2, \ldots, \beta^{i-1}\})$. Since α is the parameter controlling which variables should be included by the MB set, any component α^i_j in α^i is a discrete random variable valued in $\{0, 1\}$, where $\alpha^i_j = 1$ means that the jth variable should be included in the discovered MB in the result of the ith data block.

The simplest way to consider the $f(\{\alpha^1, \alpha^2, \ldots, \alpha^{i-1}\})$ is that only the result from the last data block is considered. Herein, the Bernoulli binomial distribution is utilized to approximate the $\mathbb{P}(\alpha^i)$, i.e.,

$$\mathbb{P}(\alpha^i) = \prod_{X_j \in \boldsymbol{X}} p\left(\alpha^i_j\right) = \prod_{X_j \in \boldsymbol{X}} \lambda_j^{\alpha^{i-1}_j} (1-\lambda_j)^{1-\alpha^{i-1}_j} \tag{14}$$

where λ_j is the likelihood that X_j should be included in the discovered MB. Parameter μ_j is introduced to describe the log odds, indicating the relative possibility of the random event $X_j \in \boldsymbol{MB}$

$$\log\frac{\lambda_j}{1-\lambda_j} = \mu_j \tag{15}$$

where μ_j would influence the attitude and likelihood to select the corresponding variable X_j before algorithm execution, which could be predetermined by domain knowledge. Specifically, $\mu_j > 0$ ($\mu_j < 0$) represents that the prior knowledge believes that X_j is (is not) an MB variable, and $\mu_j = 0$ means that there is no preference on X_j. According to (15), λ_j and $1-\lambda_j$ could be denoted as follows:

$$\lambda_j = \frac{e^{\mu_j}}{1+e^{\mu_j}}, \quad 1-\lambda_j = \frac{1}{1+e^{\mu_j}}. \tag{16}$$

According to the analyses above, to obtain the optimal parameter α^i for MB discovery in data block $\mathbb{D}_k$, we simplify the objective function in (6) with only conditional mutual information $I(T, \boldsymbol{X}_{\hat{\alpha^i}}|\boldsymbol{X}_{\alpha^i})$ and the function of prior knowledge of parameter α^i, denoted as ϕ

$$\alpha^i = \arg\min_{\alpha^i} \phi = \arg\min_{\alpha^i} \left[I(T, \boldsymbol{X}_{\hat{\alpha^i}}|\boldsymbol{X}_{\alpha^i}) - \frac{1}{|\mathbb{D}_k|} \times \log\left(\frac{\prod_{X_j\in\boldsymbol{X}} e^{\mu_j \alpha^{i-1}_j}}{\prod_{X_j\in\boldsymbol{X}}(1+e^{\mu_j})}\right)\right]. \tag{17}$$

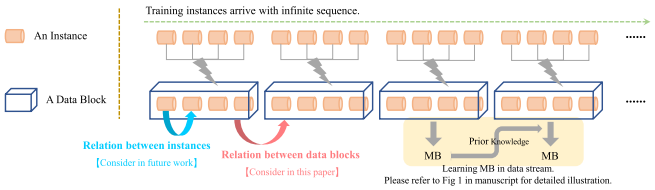


Fig. 2. Illustration to the main challenge of MB discovery in streaming data.

In Section IV, we show how to address the challenges in streaming data feature selection by incorporating prior knowledge into the feature selection process.

IV. MB Discovery in Streaming Data

Prior knowledge makes practical sense and can yield better results for MB discovery, especially in the data streams. In the following, we first introduce three practical benefits of prior terms and then propose the novel MBSD algorithm.

A. Main Challenge in Streaming Data

The goal of this article is to use MB to alleviate the negative impact of data shift, and the main challenge of MB learning in the data stream is that all existing MB learning techniques cannot be performed with one instance. Specifically, there are two types of strategies for learning MB with different techniques: 1) learning MB through CI tests and 2) learning MB through scoring the local Bayesian network around the target, both of which require a certain number of samples to complete. In other words, to learn MB in the data stream, we still need to collect a certain amount of instances to form a data block and learn MB on the data block, as shown in Fig. 2.

In this case, the relations among instances take two forms, i.e., the relation between instances (blue arrow in Fig. 2) and the relation between data blocks (red arrow in Fig. 2). As the first work to study MB learning in the data stream, the relation between data blocks is a more urgent problem to be considered since it undertakes the task of connecting training between different data blocks so that the system would not work if it is not considered. Specifically, we can neither simply union the results on different blocks nor directly trust the previous results due to possible concept drift. The solution, as an important contribution to this article, is that we propose a method to inherit the information from previous data blocks so that it can facilitate MB learning on the current data block. The derivations in Section III is to address the relationship between data blocks.

In the derivation, we assumed that the instances within a data block are independent. This simplification intent is that we temporarily ignore the impact of the relation between instances within each data block when we consider the relation between data blocks. In this way, the formula derivation can be simplified, but the disadvantage is that the instance relations within the data block are lost. Furthermore, the data block reduces the size of the instances used in each training. Considering the relation between data blocks is equivalent to considering the instance relation between different data blocks. Therefore, the usage of the data block has decreased

the instance scale in each training, which weakens the negative impact caused by the failure of the independence assumption between instances to a certain extent. Another issue with additional consideration of instance relation within a data block is that additional computational steps would increase the time complexity of the algorithm. Hence, we only focus on the relationship between data blocks in this article and take the relation between instances as an open problem.

B. Stepwise Optimization With Prior Term

As mentioned in Section II, the existing MB discovery scheme is a stepwise optimization process, where each step includes forward and backward phases [48]. The forward step adds the most possible feature X^* into the selected feature subset, as shown in (2), while the backward step deletes the feature violating MB concept from the selected feature subset, as shown in (3). In the following, we extend the scheme to implement a prior term so that the learned information in the previous data block cannot be reasonably taken into account in the learning process with the current data block. Let $\phi(\alpha^t)$ denote the optimization function

$$\phi(\alpha^t) = I(T, \boldsymbol{X}_{\hat{\alpha^t}}|\boldsymbol{X}_{\alpha^t}) - \frac{1}{|\mathbb{D}_k|}\log\left(\frac{\prod_{X_j\in\boldsymbol{X}} e^{\mu_j \alpha_j^{t-1}}}{\prod_{X_j\in\boldsymbol{X}}(1+e^{\mu_j})}\right) \tag{18}$$

where the result output in the backward phase in the tth step is denoted as α^t, and we have $\sum_i \alpha_i^t \leqslant \sum_i \alpha_i^{t-1} + 1$. Considering the prior knowledge, the α^{t-1} in the $(t-1)$th-step must satisfy the condition in (17), i.e., for $\forall\alpha'$, $\phi_{t-1}(\alpha^{t-1}) \leqslant \phi_{t-1}(\alpha')$ with $\sum_i \alpha'_i \leqslant \sum_i \alpha_i^{t-1}$. Therefore, in the tth step, minimizing $\phi(\alpha^t)$ is equivalent to solving

$$\begin{aligned} X^* &= \underset{X^*\in\boldsymbol{X}-\boldsymbol{X}_{\alpha^{t-1}}}{\arg\max}\ \phi(\alpha^{t-1}) - \phi(\alpha^t) \\ &= \underset{X^*\in\boldsymbol{X}-\boldsymbol{X}_{\alpha^{t-1}}}{\arg\max}\ I(T, X^*|\boldsymbol{X}_{\alpha^{t-1}}) - p \end{aligned} \tag{19}$$

where $\boldsymbol{X}_{\alpha^{t-1}}$ is the current selected MB set in the $(t-1)$th step (the last step), X^* is the selected variable in the forward phase in the tth step (the current step), and $I(T, X^*|\boldsymbol{X}_{\alpha^{t-1}})$ is obtained from $I(T, \boldsymbol{X}_{\hat{\alpha^{t-1}}}|\boldsymbol{X}_{\alpha^{t-1}}) - I(T, \boldsymbol{X}_{\hat{\alpha^t}}|\boldsymbol{X}_{\alpha^t}) = I(T, X^*|\boldsymbol{X}_{\alpha^{t-1}})$ according to the property of mutual information. p is the difference between the prior knowledge in the two steps, denoting as

$$\begin{aligned} p = &\frac{1}{|\mathbb{D}_{t-1}|}\log\left(\frac{\prod_{X_j\in\boldsymbol{X}} e^{\mu_j^{t-1}\alpha_j^{t-2}}}{\prod_{X_j\in\boldsymbol{X}}(1+e^{\mu_j^{t-1}})}\right) \\ &- \frac{1}{|\mathbb{D}_t|}\log\left(\frac{\prod_{X_j\in\boldsymbol{X}} e^{\mu_j^t\alpha_j^{t-1}}}{\prod_{X_j\in\boldsymbol{X}}(1+e^{\mu_j^t})}\right). \end{aligned} \tag{20}$$

In (20), the superscript added to μ_j denotes its corresponding iteration. In the latter of this section, we show the intervention technique for μ_j in each step.

When the data blocks in $(t-1)$th and tth steps are of the same size (including the case within the same data block), p could be simplified as follows:

$$\begin{aligned} p &= \frac{1}{|\mathbb{D}_t|}\sum_{X_j\in\boldsymbol{X}}\left(\mu_j^{t-1}\alpha_j^{t-2} - \mu_j^t\alpha_j^{t-1}\right) \\ &= \frac{1}{|\mathbb{D}_t|}\left(\mu_*^t - \sum_{X_j\in\boldsymbol{X}_{\alpha^{t-2}}-\boldsymbol{X}_{\alpha^{t-1}}}\mu_j^{t-1}\right) \end{aligned} \tag{21}$$

where $X_j \in \boldsymbol{X}_{\alpha^{t-1}} - \boldsymbol{X}_{\alpha^t}$ means the removed variables in the $(t-1)$th step.

Equally, the stepwise optimization in the tth backward phase is

$$\begin{aligned} X^* &= \underset{X^*\in\boldsymbol{X}_{\alpha^{t-1}}}{\arg\min}\ \phi(\alpha^{t-1}) - \phi(\alpha^t) \\ &= \underset{X^*\in\boldsymbol{X}_{\alpha^{t-1}}}{\arg\min}\ I(T, X^*|\boldsymbol{X}_{\alpha^{t-1}} - \{X^*\}) - p. \end{aligned} \tag{22}$$

Compared with the existing MB discovery scheme in batch data, (19) and (22) further take the information difference in prior terms between current and previous data blocks into consideration with p. This improvement could exploit the domain knowledge as well as the learned information in previous data to assist MB discovery in the current data block. As analyzed after (15), the influence from p could be adjusted by controlling the value of μ_j in each iteration.

C. Control Through the Prior Term

According to (19) and (21), when two candidates $X_i, X_j \in \boldsymbol{X} - \boldsymbol{X}_{\alpha^{t-1}}$ are compared in the forward phase of the tth step, if the prior knowledge is given higher confidence, then the values of μ and conditional mutual information should satisfy

$$\frac{I(T, X_i|\boldsymbol{X}_{\alpha^{t-1}}) - I(T, X_j|\boldsymbol{X}_{\alpha^{t-1}})}{\mu_i^t - \mu_j^t} < \frac{1}{|\mathbb{D}_t|}. \tag{23}$$

The backward phase has a similar condition in (23) with modified conditioning set $\boldsymbol{X}_{\alpha^{t-1}} - \{X^*\}$. Hence, the value of μ can be used to control the impact of some unreliable cases.

1) Monitor the Concept Shift and Weaken Its Influence: Since the data blocks that generate prior knowledge are not the same as the data block that executes the MB discovery algorithm, three commonly present kinds of distribution shifts might exist. As discussed in Section I, only concept shift changes the actual optimal feature subset. The covariate shift and prior probability shift can be ignored since MB discovery methods could naturally shield the influence from them. If we use the KL-divergence ratio to denote the degree of concept shift between data blocks, then μ_j of variable X_j should be inversely proportional to it, i.e.,

$$\mu_j^t \propto D_{\mathrm{KL}}^{-1}[\mathbb{P}_{t-1}(T \mid X)||\mathbb{P}_t(T \mid X)]. \tag{24}$$

Some concept shift detection techniques [37], [38], [39], [40] can be used to estimate the value of $D_{\mathrm{KL}}^{-1}[\mathbb{P}_{t-1}(T \mid X)||\mathbb{P}_t(T \mid X)]$. In this article, we use microcluster nearest neighbor (MC-NN) [51], [52] to cluster the samples, in which the split rate $\mathrm{Sr}[t]$ and the death rate $\mathrm{Rr}[t]$ in the tth step are monitored to

detect the concept shift [1], [53]. They are calculated as

$$\mathrm{Sr}[t] = \frac{1}{t}\sum_{i=1}^{n}(N_{\mathrm{Split}}[i] - N_{\mathrm{Split}}[i-1]) \tag{25}$$

$$\mathrm{Rr}[t] = \frac{1}{t}\sum_{i=1}^{n}(N_{\mathrm{Remove}}[i] - N_{\mathrm{Remove}}[i-1]) \tag{26}$$

where t is the serial number of the current data block (or time window), and $N_{\mathrm{Split}}[i]$ and $N_{\mathrm{Remove}}[i]$ denote the number of split and removal operations in ith data blocks. According to (24), μ_j^t should be proportional to the degree of concept drift

$$\mu_j^t \propto \frac{|\mathrm{Sr}[t] - \mathrm{Sr}[t-1]|}{\frac{1}{2}(\mathrm{Sr}[t] + \mathrm{Sr}[t-1])} \frac{|\mathrm{Rr}[t] - \mathrm{Rr}[t-1]|}{\frac{1}{2}(\mathrm{Rr}[t] + \mathrm{Rr}[t-1])} \tag{27}$$

where the two fractions denote the difference of Sr and Rr between two neighboring data blocks, which further reflects the degree of concept drift. By this consideration, the negative influence from prior knowledge is restrained if the concept shift is significant. Considering the zero value in the case without drift, a nonzero constant (e.g., 1) can be added to the value of $D_{\mathrm{KL}}^{-1}[\mathbb{P}_{t-1}(T \mid X)||\mathbb{P}_t(T \mid X)]$ to guarantee the impact of undermentioned factors on μ_j^t.

2) Weaken the Negative Influence From the Unreliable Result: The existing MB discovery scheme usually takes a nonzero threshold value to test the conditional mutual information [19], [54], which should be predetermined as a hyperparameter. Specifically, the conditional mutual information $I(X, Y|\boldsymbol{Z})$ could be measured by the CI test with a $G^2(X, Y|\boldsymbol{Z})$ statistic [10], which approaches the chi-squared distribution asymptotically

$$G^2(X, Y|\boldsymbol{Z}) \sim \chi^2(f) \tag{28}$$

where f denotes the appropriate degrees of freedom and is calculated as $f = (\mathbb{R}_X - 1)(\mathbb{R}_Y - 1)\prod_{Z\in\boldsymbol{Z}}\mathbb{R}_Z$ ($\mathbb{R}_*$ denotes the value domain of the corresponding variable $*$).

To improve the reliability of the CI test, more than s samples need to be arranged on each degree of freedom. Therefore, the sample scale of the entire data block should satisfy

$$\frac{|\mathbb{D}_t|}{(\mathbb{R}_X - 1)(\mathbb{R}_Y - 1)\prod_{Z\in\boldsymbol{Z}}\mathbb{R}_Z} > s. \tag{29}$$

The higher the value of s is, the higher the reliability of the corresponding CI test is. Due to this demand, the fraction between the entire data scale and the demanding sample scale can be used to measure the reliability of the prior term (learned information) from previous data blocks. Specifically, if a variable X_j is added into the current MB set or removed from the current MB set via calculating $G^2(X_j, T|\boldsymbol{Z})$ in data block $\mathbb{D}_t$, then the result could be employed as prior knowledge with a reliability parameter μ_j

$$\mu_j^t = \frac{|\mathbb{D}_t| D_{\mathrm{KL}}^{-1}[\mathbb{P}_t(T \mid X)||\mathbb{P}_{t+1}(T \mid X)]\sigma}{s(\mathbb{R}_{X_j} - 1)(\mathbb{R}_T - 1)\prod_{Z\in\boldsymbol{Z}}\mathbb{R}_Z} \tag{30}$$

in which μ_j is influenced by three parts: 1) the reliability of the CI test; 2) the KL-divergence between the conditional probability distribution in two data blocks; and 3) a scale parameter without any other practical meaning. For the first part, the more reliable the CI test is, the higher μ_j, which further has a more significant influence on the MB discovery through determining the value of λ_j according to (16).

3) Control the Scale of Selected Features: As we know, MB-based feature selection does not require the number of selected features to be predetermined. However, they might lose their flexibility and practicality in some cases due to their rigidity. This issue could be assuaged by controlling σ in μ_j^t. Taking p in (19) and (22) as the threshold value for the CI test in (28), the negative value of $I(T, X^*|\boldsymbol{X}_{\alpha^{t-1}} - \{X^*\}) - p$ means that the corresponding X^* is independent of the target conditioned on current MB set. Thus, X^* is not an MB variable according to the definition of MB and should be removed from $\boldsymbol{X}_{\alpha^{t-1}}$. Assume that the features are reordered according to the selected order, the last feature added to MB is X_τ, and the last tested feature is $X_{\tau+1}$; then, we have

$$\chi_a^2(f_\tau) < \mu_\tau^t - \sum_{j=1}^{\tau-1}\mu_j^{t-1}, \quad \chi_a^2(f_{\tau+1}) > \mu_{\tau+1}^{t+1} - \sum_{j=1}^{\tau}\mu_j^t \tag{31}$$

where a denotes the confidence coefficient. The corresponding value of μ after the τth selected feature could be set according to the $\chi_a^2(f_\tau)$ and $\chi_a^2(f_{\tau+1})$ by calculations in (31), which could be achieved by adjusting the σ of μ_j for $j > \tau$ to limit the size of the selected feature set to around τ. Note that this adjustment is not necessary in most cases. Only when the expected performance could not be achieved under the automatically determined MB set scale, the scale can be appropriately relaxed or tightened near the automatically determined scale.

D. MB Discovery Algorithm in Streaming Data

Based on the theoretical analyses above, we present an MBSD in Algorithm 1. MBSD takes a continuous stream of data blocks as input and outputs the learned MB in the current data block (line 1). Two variable sets ***LastMB*** and ***CMB*** should be used to keep the results in the last data block and the current data block, respectively, which are initialized as empty sets (line 2). MBSD consists of two phases: Phase 1 (lines 5–9) acquires prior knowledge from the results of the last data block, and Phase 2 (lines 10–20) learns the MB in the current data block.

Phase 1 (Lines 5–9): Phase 1 calculates and updates the parameters used in the computation of the prior term. First, line 5 estimates the KL-divergence between the conditional probability distribution in two adjacent data blocks, which could be used in parameter μ_j to measure the concept shift. Here, the MC-NN is used to cluster the samples, in which the split rate (25) and the death rate (26) are monitored to measure the concept shift with calculation in (27). Subsequently, lines 7 and 8 update parameters μ_j and λ_j, respectively, for each variable $X_j \in \boldsymbol{X}$, so that the prior term in (20) and (21) could be updated in Phase 2.

Phase 2 (Lines 10–20): Based on Phase 1, Phase 2 adopts a forward step (lines 11–13) and a backward step (lines 14–19) to learn the MB set with help of prior knowledge from previews data blocks and domain knowledge. Both of these steps test features by analyzing conditional dependence and independence in data, which could avoid the negative influence

Algorithm 1 MBSD

1: **Input:** Streaming data $\mathbb{D} = \{\mathbb{D}_1 \cup \mathbb{D}_2 \cup \cdots \cup \mathbb{D}_k \cup \cdots\}$, Target T and features set $\boldsymbol{X}$; Scale parameter σ.
2: **Initialization:** The MB set learned in last data block $\boldsymbol{LastMB} \leftarrow \varnothing$; The MB set to be learned in current data block $\boldsymbol{CMB} \leftarrow \varnothing$;
3: **repeat**
4: Input a data block $\mathbb{D}_k$ from streaming data $\mathbb{D}$.
*{Phase 1: Acquire prior knowledge from **LastMB**.}*
5: Estimate $D_{KL}[\mathbb{P}_{i-1}(T \mid \boldsymbol{X}) || \mathbb{P}_i(T \mid \boldsymbol{X})]$.
6: **for** each $X_j \in \boldsymbol{X}$ **do**
7: Update μ_j via (30)
8: Calculate λ_j via (16) and ***LastMB***.
9: **end for**
{Phase 2: Learn the MB in $\mathbb{D}_k$.}
10: **repeat**
11: $X^* = \underset{X \in \boldsymbol{X} - \boldsymbol{CMB}}{\arg\max} [G^2(X, T|\boldsymbol{CMB}) - p]$ via (20)
12: $\boldsymbol{CMB} \leftarrow \boldsymbol{CMB} \cup \{X^*\}$
13: Update μ_{X^*}, λ_{X^*} based on CI test in line 11
14: **for** each $X_j \in \boldsymbol{CMB}$ **do**
15: **if** $G^2(X_j, T|\boldsymbol{CMB} - \{X_j\}) - p < 0$
16: $\boldsymbol{CMB} \leftarrow \boldsymbol{CMB} - \{X_j\}$
17: Update μ_{X_j}, λ_{X_j} based on CI test in line 15
18: **end if**
19: **end for**
20: **until** the ***CMB*** does not change.
21: $\boldsymbol{LastMB} \leftarrow \boldsymbol{CMB}$, $\boldsymbol{CMB} \leftarrow \varnothing$.
22: **until** no data blocks are input.
23: **Output:** The learned MB set ***LastMB***.

of covariate shift and prior probability shift. Instead of directly using the results from preview data blocks, MBSD transforms this information to prior term p and controls their influence on the current results by a reliability parameter. The purpose of these assignments is that MBSD could not only effectively use the information from previous data blocks via prior term but also avoid the possible influence of concept shift between different data blocks via reliability parameters. Specifically, line 11 chooses the most proper variable and adds it to the current MB ***CMB*** in line 12. Then, line 14 retests each candidate in ***CMB***, and line 16 removes the false MB variables. The size of samples participating in all the above calculations is recorded, to compute the reliability in (30) of corresponding results, which is used to update μ_{X_j} and λ_{X_j} when X_j is selected (line 13) or deleted (line 17). Furthermore, to optimize the performance of MBSD when executing, the data structure pipeline machine proposed in [22] is used to store the instances that could improve the reliability of the prior term.

E. Discussion of MBSD

We analyze the time complexity of MBSD on each data block, denoted as $\mathbb{D}_k$ in the following. MBSD consists of two phases: Phase 1 (lines 5–9) and Phase 2 (lines 10–20). The computational cost of Phase 1 in MBSD is relatively lower, $O(|\mathbb{D}_k|)$ in line 5, and $O(|\boldsymbol{X}|)$ in lines 6–9. Hence, the time complexity of MBSD is determined by Phase 2. Similar to other MB-based feature selection algorithms, the efficiency performance based on CI tests is measured in the number of association calculations or CI tests executed. In Phase 2, the number of iterations is $|\boldsymbol{X} - \boldsymbol{CMB}|$ according to line 11, and lines 14–19 perform $|\boldsymbol{CMB}|$ computations for each variable that enters ***CMB***. Since $|\boldsymbol{CMB}| = O(\boldsymbol{MB})$, the number of CI tests in MBSD is $O(|\boldsymbol{MB}||\boldsymbol{X}|)$. Each CI test $G^2(X, Y|\boldsymbol{Z})$ performs $\max\{\mathbb{R}_X\mathbb{R}_Y\mathbb{R}_{\boldsymbol{Z}}, |\mathbb{D}_k|c_m\}$ calculations, where c_m denotes the maximum size of conditioning set manually predetermined.

Next, we first explain what feature selection algorithms are good and effective in streaming data. Then, we explain how MBSD meets these requirements. Feature selection in the data stream is affected by distribution changes in the underlying system. The optimal feature subset needs to be determined according to the underlying mechanism $\mathbb{P}(T|\boldsymbol{X})$. Therefore, the selected feature subset should remain stable if the current distribution is stable (i.e., without concept drift), and in this case, an algorithm that exhibits stability is a good algorithm. Conversely, if there exists concept drift in the data stream, then the true optimal feature subset should also change with the occurrence of concept drift. Correspondingly, the feature subset selected by a good algorithm should also change with the passage of instances.

MBSD meets the aforementioned requirements for a good algorithm. Specifically, when concept drift does not occur, MBSD is more stable than other algorithms since MBSD focuses on the underlying mechanism $\mathbb{P}(T|\boldsymbol{X})$ and selects features by analyzing the conditional (in)dependence relationship between variables. Moreover, although the information is incomplete at each timestamp, MBSD takes the learned information in historical data as prior knowledge and incorporates them into the current MB discovery process. Therefore, without concept drift, the effective information extracted from historical data will continue to accumulate, making the distribution information obtained by MBSD closer to the real distribution of the underlying system. On the other hand, when concept drift occurs, MBSD perceives the distribution changes through a concept drift monitor, and the magnitude of this change directly influences the confidence of the results calculated on previous data blocks according to (24). Hence, MBSD remains stable under stable distributions and quickly adapts to the shifted distributions under concept drift.

V. Experiments

We first validate the performance of MB discovery in the streaming data sampled from the standard Bayesian network in Section V-A, where the MBSD is compared with four MB discovery algorithms and its variant without prior term. In Section V-B, MBSD is compared with four state-of-the-art feature selection algorithms on high-dimensional real-world datasets to evaluate the feature selection performance in the streaming data. In Section V-C, we aim to demonstrate the robustness of MBSD against distribution shift, where the feature selection task is conducted on synthetic streaming data with different types of shifts.

A. Task 1: MB Discovery in Streaming Data

1) Experimental Settings: Since there are no algorithms for MB discovery in the data stream, we deploy existing MB

TABLE I
DETAILS OF FIVE DATASETS FOR MB DISCOVERY

Data set	Alarm	Child	Gene	Link	Pigs
#Variables	37	20	801	724	441
#Edges	46	25	972	1125	592

TABLE II
F_1-SCORE AND log TIME OF MBSD AND THE COMPARING STATE-OF-THE-ART MB DISCOVERY ALGORITHMS ON STREAMING DATA

Algorithm	Metric	Alarm	Child	Gene	Link	Pigs
FastIAMB	F_1	0.471	0.580	0.597	0.265	0.739
	$\log Time$	0.04	0.04	0.42	0.19	0.22
STMB	F_1	0.464	0.548	0.072	0.079	0.100
	$\log Time$	0.17	0.16	1.83	34.22	4.44
BAMB	F_1	0.688	0.713	0.602	0.232	0.735
	$\log Time$	0.34	0.33	2.63	268.18	5.17
EEMB	F_1	0.695	0.777	0.623	0.256	0.766
	$\log Time$	0.18	0.18	1.50	8.40	3.76
MBSD	F_1	**0.756**	**0.854**	**0.670**	**0.440**	**0.801**
	$\log Time$	0.21	0.14	2.98	2.05	1.82
	Variation of F_1	↑ 8.8%	↑ 9.9%	↑ 7.5%	↑ 66.0%	↑ 4.6%

TABLE III
COMPARISON OF MBSD AND ITS VARIANT WITHOUT PRIOR TERM

Algorithm	Metric	Alarm	Child	Gene	Link	Pigs
MBSD	F_1	**0.756**	**0.854**	**0.670**	**0.440**	**0.801**
	$\log Time$	0.21	0.14	2.98	2.05	1.82
MBSD-p	F_1	0.659	0.705	0.634	0.262	0.757
	$\log Time$	0.05	0.06	0.76	0.52	0.42

discovery algorithms to search the MB set in the current data block based on the results (MBs) learned in the previous data blocks. We choose four state-of-the-art MB learning algorithms, including FastIAMB [45], STMB [21], BAMB [46], and EEMB [47]. The parameter of the G^2-test [19] in MB discovery algorithms is set to 0.05. The F_1-score is calculated to measure the performance of different MB learning algorithms, which considers the Precision (the number of discovered true positives divided by the total number of discovered variables) and the Recall (the number of discovered true positives divided by the total number of true positives) simultaneously as

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{32}$$

where Precision and Recall are calculated as

$$\text{Precision} = \frac{|\boldsymbol{MB}_S \cap \boldsymbol{MB}|}{|\boldsymbol{MB}_S|}, \quad \text{Recall} = \frac{|\boldsymbol{MB}_S \cap \boldsymbol{MB}|}{|\boldsymbol{MB}|} \tag{33}$$

where $\boldsymbol{MB}$ denotes the true MB and $\boldsymbol{MB}_S$ denotes the searched MB, output by any comparing method.

Five datasets, sampled from standard Bayesian networks [19], are used to evaluate the performance of MB discovery, whose statistical information is provided in Table I. The main reason to use these data is that the MB of each variable in the Bayesian network is exactly known, i.e., the parents, children, and spouses of the variable, which could be directly read out from the directed acyclic graph. Each dataset has 5000 instances, divided into ten data blocks. Each comparing algorithm learns the MBs of all variables in each data block, and the F_1-score and the logarithmic CPU time are recorded in Tables II and III to compare the accuracy and time efficiency of these algorithms. The experiment is conducted with Inter i5-8500 3.00-GHz CPU and 16-GB memory.

2) Superiority Against the State of the Art: "Variation of F_1" in the last row of Table II demonstrates the proportion of performance improvement of MBSD compared with the best baseline. From Table II, we can observe that MBSD consistently achieves better accuracy than state-of-the-art MB discovery algorithms. Since these comparing algorithms are not designed to solve data streams, they cannot use the abundant information in previous data blocks. However, MBSD could transform the results in previous data blocks to prior knowledge and employ it to assist the MB discovery in the current data block. Moreover, MBSD is more time-efficient than other algorithms except for FastIAMB since MBSD is designed based on simultaneous learning methods, which could meet the demands for the efficiency of real-time streaming data. Compared with algorithms other than FastIAMB, the time-efficiency superiority of MBSD is expanded with the increase in data scale, which demonstrates that, although MBSD consumes extra time to process the calculation of the prior term, the growth rate of the execution time is linear. Compared with Fast-IAMB, the extra time consumed by MBSD is worth since it achieves significantly better performance. Considering the performance of both accuracy and efficiency, MBSD is more practical than others.

3) Effectiveness of Prior Term: To validate the effectiveness of the prior term, we compare MBSD with its variant, denoted as MBSD-p in Table III, whose prior term and related steps are removed. As shown in Table III, MBSD significantly improves the accuracy of MB discovery with a reasonable time cost, which demonstrates the effectiveness of the prior term.

B. Task 2: Feature Selection in Streaming Data

1) Experimental Settings: To demonstrate the performance of the MBSD for feature selection in real-world applications, in this section, MBSD is evaluated on 12 datasets in different scales, which are taken from diverse application domains. Six of them are commonly used in data stream learning, which includes millions of instances so that they can be used to simulate the real-world online environment. Another added benefit is that four of the streaming datasets have known types of concept drift, and we can use these datasets to study the effectiveness of comparing algorithms against the effects of concept drift. However, these datasets always contain small-scale feature sets with few redundant or irrelevant features, incapable to differentiate the state-of-the-art feature selection algorithms. Therefore, these datasets are additionally introduced 50 redundant features and 50 irrelevant features. Moreover, the other six datasets, commonly used in batch learning, are also employed to evaluate these compared algorithms in the streaming setting. These datasets contain tens of thousands of features, with more redundant and irrelevant features. Table IV provides the standard statistics (i.e., the number of features, instances, and relevant features) and data information (i.e., the description of prediction tasks and drift types) of these datasets.

Four state-of-the-art feature selection algorithms for streaming data are compared, including OFS [11], SOFS [12], ARDA [13], and AMD [13]. Each dataset is divided into

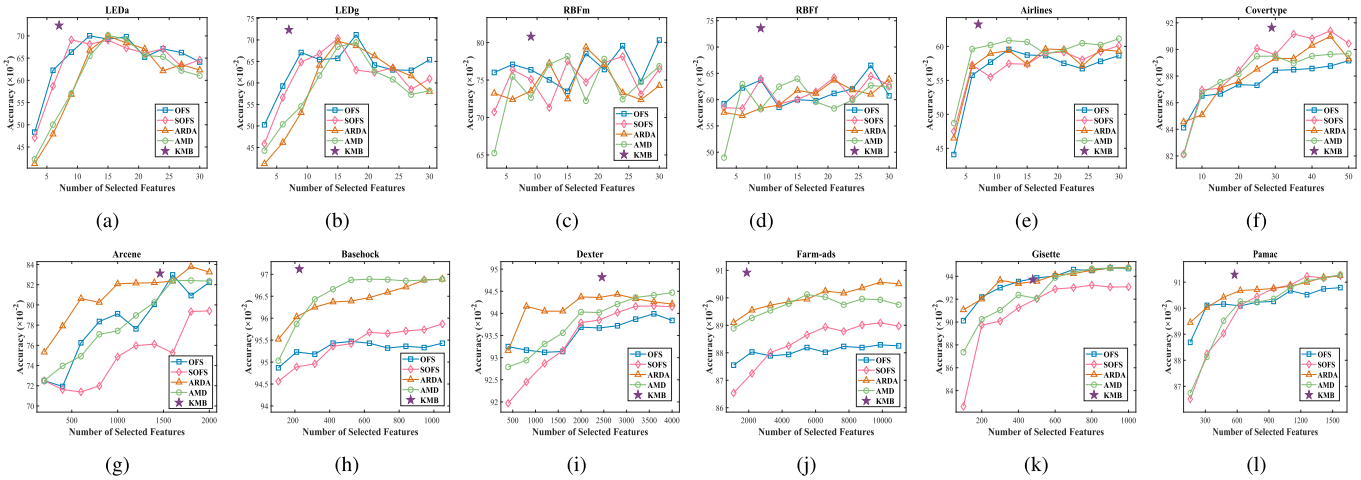


Fig. 3. Classification accuracy achieved with Perceptron on 12 streaming datasets. (a) LED$_a$. (b) LED$_g$. (c) RBF$_m$. (d) RBF$_f$. (e) Airlines. (f) Covertype. (g) Arcene. (h) Basehock. (i) Dexter. (j) Farm-ads. (k) Gisette. (l) Pamac.

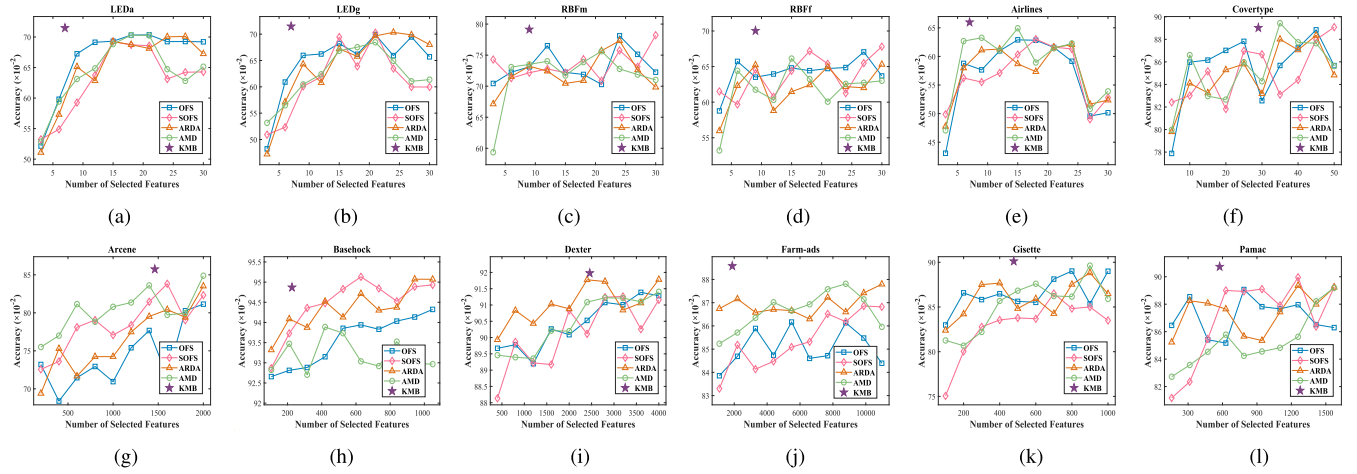


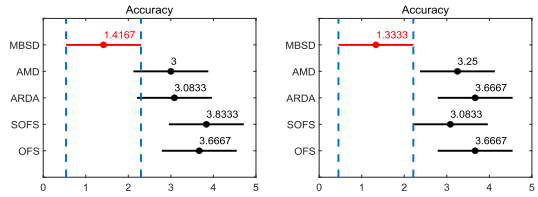
Fig. 4. Classification accuracy achieved with SVM on 12 streaming datasets. (a) LED$_a$. (b) LED$_g$. (c) RBF$_m$. (d) RBF$_f$. (e) Airlines. (f) Covertype. (g) Arcene. (h) Basehock. (i) Dexter. (j) Farm-ads. (k) Gisette. (l) Pamac.

TABLE IV
DETAILS OF THE EXPERIMENTAL DATASETS FOR FEATURE SELECTION

Data set	Prediction Task	#Features	#Relevant Features	#Instances	Drift Type
LED$_a$	Synthetic	24+100	$\leqslant 7$	1,000,000	Abrupt
LED$_g$	Synthetic	24+100	$\leqslant 7$	1,000,000	Gradual
RBF$_m$	Synthetic	10+100	$\leqslant 10$	1,000,000	Moderate Incremental
RBF$_f$	Synthetic	10+100	$\leqslant 10$	1,000,000	Fast Incremental
Airlines	Delayed Flight	7+100	$\leqslant 7$	539,383	Unknown
Covertype	Forest Covertype	54+100	$\leqslant 54$	581,012	Unknown
Arcene	Cancer Identification	10,000	Unknown	200	Unknown
Basehock	News Classification	4,862	Unknown	1,993	Unknown
Dexter	Corporate Acquisitions	20,000	Unknown	600	Unknown
Farm-ads	Ad Approvement	54,877	Unknown	4,143	Unknown
Gisette	Digit Recognition	5,000	Unknown	7,000	Unknown
Pamac	Text Classification	7,510	Unknown	1,945	Unknown

several data blocks. In each data block, MBSD and these comparing methods are utilized to select the relevant features, and then, two classifiers (perceptron and SVM) are trained with these selected features. Each experiment is repeated ten times with different training and test data, and we report the average performances (classification accuracy) on the testing set. The regularization parameters for all algorithms are searched from $\{0.01, 0.1, 0.3, \ldots, 0.9, 1\}$ by grid search. In addition, since MBSD measures the importance of features by uncovering the underlying mechanisms rather than calculating the perdition performance, MBSD does not need to predetermine the number of selected features, while, for other comparing algorithms, we gradually increase the number of the selected features until the performances tend to be stable.

2) Superiority Against the State of the Art: Figs. 3 and 4 show the average classification performance curves with respect to the number of selected features. We can conclude that MBSD significantly outperforms other methods under the same feature scale in 11 out of 12 datasets, which demonstrates that the feature subsets selected by MBSD are more effective due to the uncovered underlying mechanism. In Gisette with Perceptron, MBSD also achieves very competitive performances that are similar to the best results in these comparing algorithms. Compared with the best results of each comparing algorithm, MBSD achieves better performance in ten out of 12 datasets with both classifiers. Overall, the observed superiority of MBSD demonstrates that the consideration of distribution shift is valuable in the feature selection of data streams. Compared with existing algorithms only focusing on prediction performance on off-line data, MBSD could resist the concept shift and select more predictive and informative features by introducing a prior term to the traditional framework, whose classification performance is noticeably improved as shown in these experiments. In addition, the superiority of LED$_a$, LED$_g$, RBF$_m$, and RBF$_f$ just demonstrates that MBSD could to some extent overcome the negative impact of concept drift and achieve better performance. We also provide the number and proportion of features selected by MBSD in



(a) (b)

Fig. 5. Friedman test for MBSD and other comparative algorithms. The dots (numerical values) and bars denote the average ranks and the critical differences, respectively, and the methods that have nonoverlapped bars are significantly inferior to the proposed approach. (a) With perceptron. (b) With SVM.

TABLE V
NUMBER AND PROPORTION OF FEATURES SELECTED BY MBSD

Dataset	LED_a	LED_g	RBF_m	RBF_f	Airlines	Covertype
Number of features	124	124	110	110	107	154
Number of selected features	7	7	9	9	7	29
Percentage of selected features	5.6%	5.6%	8.2%	8.2%	6.5%	18.8%
Dataset	Arcene	Basehock	Dexter	Farm-ads	Gisette	Pamac
Number of features	10,000	4,862	20,000	54,877	5,000	7,510
Number of selected features	209	1,129	2,459	1,883	478	549
Percentage of selected features	2.1%	4.3%	12.3%	3.4%	9.6%	7.3%

Table V. We observe that the proportion of features selected by MBSD does not exceed 20% in all datasets, and in ten of them, the proportion of features selected by MBSD is less than 10%. Hence, MBSD can reduce the number of features and simultaneously improve accuracy.

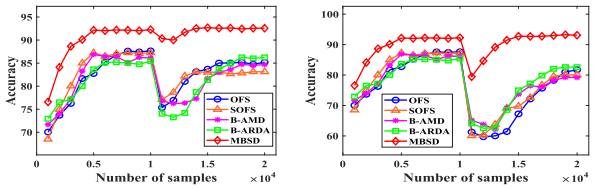
To further demonstrate the superiority of MBSD, we utilize the Friedman test [55] combined with a post-hoc test to make comparisons of five methods over 12 datasets. The performance of two algorithms is significantly different if their average ranks on all datasets differ by at least the critical difference CD, calculated as

$$\text{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{34}$$

where k is the number of algorithms and N is the number of datasets. The significance level α is set to 5%, and the critical value $q_\alpha = 2.728$. Fig. 5 shows the Friedman test results on the 12 datasets. We observe that the differences between MBSD and other algorithms are significant (greater than CD on classifier SVM and close to CD on classifier Perceptron), which indicates that the proposed MBSD ranks the highest among the five methods and is significantly different from most comparative methods.

C. Robustness Against Distribution Shift in Feature Selection

1) Experimental Settings: We conduct feature selection experiments on synthetic datasets to evaluate the performance variation of MBSD under distribution shift. The location of underlying distribution change in simulated data is artificially set. The synthetic distributions contain 50 relevant features and 300 noisy features. The datasets are sampled from the synthetic distribution with the simulation method presented in [56] and consist of 20 000 samples divided into 20 data blocks. In each dataset, we install a gradual distribution shift centered around the 10 000th samples. Therefore, we can observe the performance change of these comparing algorithms between $[1000, 10\,000]$ and $[11\,000, 20\,000]$ sample sections. The experiments examine the capacity of different algorithms against the different types of shifts. Specifically,



(a) (b)

Fig. 6. Classification accuracy achieved with features selected by MBSD and four state-of-the-art feature selection algorithms on two synthetic datasets. (a) Synthetic data with covariate and prior probability shift. (b) Synthetic data with concept shift.

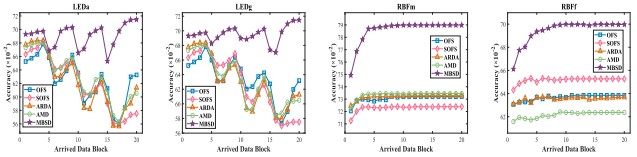
the feature distribution $\mathbb{P}(\boldsymbol{X})$ and the target distribution $\mathbb{P}(T)$ in the first dataset are changed to simulate the covariate shift and prior probability shift. The underlying mechanism (relationships between features and target) $\mathbb{P}(T|\boldsymbol{X})$ in the second dataset is changed to simulate the concept shift. The aforementioned four feature selection algorithms for streaming data are compared. In each data block, the MBSD and other compared algorithms are employed to select features, based on which the classifiers are trained to measure the predictability of selected features. The selected features are fixed as 50 for these compared algorithms, and other experimental settings are the same as Section V-B.

2) Robustness Against Distribution Shift: Fig. 6 provides the performance variation curves with respect to the arriving instances. We can conclude from Fig. 6 that the following holds.

1) Covariate and prior probability shift almost have no negative influence on MBSD but lead to the performance degradation of the compared four algorithms. The main reason is that these two types of shifts do not change the underlying mechanism $\mathbb{P}(T|\boldsymbol{X})$, which is just what the MB discovery process learns. Therefore, MBSD could select a more accurate feature subset than others.
2) Concept shift has a higher influence on these algorithms than the other two types. Since concept shift changes the distribution $\mathbb{P}(T|\boldsymbol{X})$, the knowledge from previous data blocks is invalid. In this case, MBSD could discover the change by calculating the KL-divergence between distributions in two data blocks, and then, the reliability parameter in MBSD limits the influence of prior knowledge from previous data blocks. Therefore, the performance of MBSD drops to a point similar to the starting point (1000 instances) due to insufficient samples. However, the four comparing algorithms are misguided by the learned information from previous blocks, leading to worse performances.

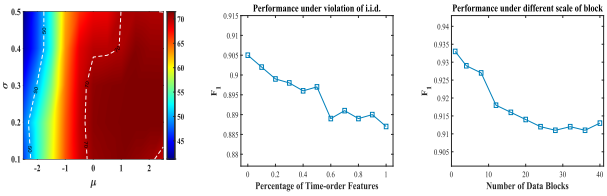
D. Influence of Different Types of Drift

In this section, we study the impact of different types of concept drift on MBSD and other comparing algorithms. The four synthetic datasets with different characteristics in Table IV are utilized to show how MBSD performs in each of these scenarios, which includes abrupt, gradual, moderate incremental, and fast incremental drifts. Specifically, LED_a and LED_g simulate drifts by swapping relevant features with irrelevant features. There are three drifts each with an amplitude of 50 000 instances and centered at the 250 000, 500 000,



(a) (b) (c) (d)

Fig. 7. Performance variation under different types of drifts. (a) LED_a. (b) LED_g. (c) RBF_m. (d) RBF_f.



(a) (b) (c)

Fig. 8. Parameter and assumption analyses. (a) Parameter analysis. (b) i.i.d. assumption. (c) Scale of the block.

and 750 000 instances, respectively. RBF creates centroids at random positions as labels and new instances around one centroid selected at random, whose values are set according to a random direction chosen to offset the centroid. To simulate incremental drifts, centroids move at a continuous rate, effectively causing new instances that ought to belong from one centroid to another with a different class. RBF_f and RBF_m simulate a fast/moderate incremental drift via setting a speed of change to 10^{-3} and 10^{-4}. Each dataset is separated into 20 blocks and randomly sampled for training and test sets. SVM is utilized to measure the predictability of the selected features.

To observe the impact of different types of drift, the number of selected features for each comparing algorithm is set to the actual number of relevant features in these synthetic data, i.e., 7 and 10 in LED and RBF, respectively. Fig. 7 shows the predictability of the selected features when data blocks sequentially arrive. We can conclude that the selected features of MBSD show better predictability compared with the state of the art. When abrupt and gradual drifts occur, the performance of MBSD can recover to that before the drift occurs, while other comparison algorithms continue to suffer from the drift although negative effects can be mitigated to some extent with the arrival of subsequent data blocks. In datasets LED_a and LED_g, MBSD detects the occurrence of drift earlier and adjusts the credibility of previous data blocks, thereby stopping performance degradation earlier. Other algorithms take a longer time to recover. In datasets RBF_f and RBF_m, other comparing algorithms suffer from the drift and fail to exploit the information in current data blocks to improve their performance, while MBSD could employ this knowledge and further select more predictive features.

E. Analyses About Parameter and Assumption

We provide the parameter analysis for MBSD in this section. As shown in Section IV, two parameters may be predetermined, i.e., the initial value of μ and the scale parameter σ. In this experiment, we adjust the initial μ value of all relevant features and σ in the LED dataset and observe the performance variation of MBSD. From Fig. 8(a), we conclude that, when $\mu \in [0, 2]$, the performance of MBSD tends to be stable and is better than other value ranges, demonstrating that nonnegative value should be set for relevant features. When $\mu \in [1.5, 1.6]$ and $\sigma \in [0.2, 0.3]$, MBSD can achieve the best performance. Suggestively, the initial value of μ could be determined by the prior knowledge and set to 0 without preference in the prior knowledge. Moreover, σ should not be set too large or too small; moderate values can ensure the accuracy of MBSD. σ usually be adjusted according to the scale of datasets and CI tests, and has a weaker impact on the performance than the sign (positive/negative) of μ.

To further observe the influence of the independent identically distributed (i.i.d.) assumption in each data block, we conduct an experiment on synthetic data, with 25 relevant features, 75 redundant features, and 5000 instances. We adjust the percentage of time-order features to observe the performance variation in the case violating the i.i.d. assumption. Under different settings, we randomly select a certain proportion of features from relevant features and redundant features as time-order features, and their values are affected by the values of the previous instances. The result is shown in Fig. 8(b), which demonstrates that the performance of MBSD will be slightly affected when the proportion of temporal features increases, but it can maintain good performance on the whole.

To observe the impact of the number of samples in each data block, we try to divide the simulation data of 10 000 samples into different numbers of data blocks. Thus, the total number of instances remains unchanged, and the number of samples in each data block is different. The performance variation of MBSD is shown in Fig. 8(c). From Fig. 8(c), we can conclude that aggregation of samples in fewer data blocks is conducive to the improvement of algorithm performance, which is consistent with our common sense because using more samples for one training can better reflect the underlying distribution. In addition, the loss in the information transmission process could also be avoided. However, when data are split into more data blocks, the impact on algorithm performance is limited because MBSD significantly reduces information loss by inheriting the information of the previous data block.

VI. Conclusion

This article investigates the feature selection problem in the data stream. We discuss different types of distribution shifts and their impact on feature selection and then propose a novel algorithm MBSD to improve the robustness against distribution shifts. Different from existing algorithms focusing on the prediction performance on off-line data, MBSD mines the underlying mechanism via analyzing conditional dependence and independence in data to learn the MB of the class attribute. Hence, it is naturally more robust against the covariate shift and the prior probability shift. To consider the concept shift, MBSD transforms the abundant information in previous data blocks to prior knowledge, employs them to assist MB discovery in later data blocks, and then uses the likelihood of concept shift to measure the reliability of a prior term, which further improves the robustness.

As the first research for MB learning in the data stream, MBSD is a novel and promising algorithm. Nonetheless,

MBSD still has some limitations to be addressed. First, the evaluation protocols and datasets could be not representative, constrained by either sample or feature sizes. Mainly because the commonly used streaming datasets always contain small-scale feature sets with few redundant or irrelevant features, while the commonly used high-dimensional datasets have limited sample sizes and cannot be used to simulate the data flow, next, the derivation from (7) to (8) is based on sample average approximation [49], which requires sufficient samples. However, it cannot be satisfied in real-world applications. Furthermore, to simplify the analyses, we ignore the impact of the instance relations within each data block when the relation between data blocks is considered. As mentioned in Section IV-A, instances in data streams for online learning research are not independent, while it is undoubtedly challenging to consider the instance dependence within the same data block when learning MB, which will shake the foundation of existing MB learning techniques. A possible future direction is to propose a new CI test method or Bayesian network scoring method, which should naturally consider the correlation between instances as learning MB. Finally, MBSD extends the simultaneous learning framework with a prior term, whose idea is also possible to employ to improve the divide-and-conquer framework. This research may further improve the performance of MB-based feature selection on the data stream.

REFERENCES

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.

[2] B. Krawczyk, L. L. Minkub, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, Sep. 2017.

[3] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1393–1402, Sep. 2013.

[4] K. Manchella, A. K. Umrawal, and V. Aggarwal, "Flexpool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2035–2047, Apr. 2021.

[5] L. Wu, Z. Li, H. Zhao, Q. Liu, and E. Chen, "Estimating fund-raising performance for start-up projects from a market graph perspective," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108204.

[6] L. Wu, Z. Li, H. Zhao, Z. Pan, Q. Liu, and E. Chen, "Estimating early fundraising performance of innovations via graph-based market environment model," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6396–6403.

[7] L. Wu, H. Wang, E. Chen, Z. Li, H. Zhao, and J. Ma, "Preference enhanced social influence modeling for network-aware cascade prediction," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 3–19.

[8] J. Li et al., "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Jan. 2018.

[9] B. Jiang, C. Li, M. D. Rijke, X. Yao, and H. Chen, "Probabilistic feature selection and classification vector machine," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 2, pp. 1–27, Apr. 2019.

[10] P. Spirtes et al., *Causation, Prediction, and Search*. Cambridge, MA, USA: MIT Press, 2000.

[11] J. Wang, P. Zhao, S. C. H. Hoi, and R. Jin, "Online feature selection and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 698–710, Mar. 2014.

[12] Y. Wu, S. C. H. Hoi, T. Mei, and N. Yu, "Large-scale online feature selection for ultra-high dimensional sparse data," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 4, pp. 1–22, Nov. 2017.

[13] T. Zhai, H. Wang, F. Koriche, and Y. Gao, "Online feature selection by adaptive sub-gradient methods," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Dubin, Ireland. New York, NY, USA: Springer, 2018, pp. 430–446.

[14] Y. Liu, Y. Yan, L. Chen, Y. Han, and Y. Yang, "Adaptive sparse confidence-weighted learning for online feature selection," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 4408–4415.

[15] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[16] I. Žliobaitė, "Learning under concept drift: An overview," 2010, *arXiv:1010.4784*.

[17] S. Mehta, "Concept drift in streaming data classification: Algorithms, platforms and issues," *Proc. Comput. Sci.*, vol. 122, pp. 804–811, Jan. 2017.

[18] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 521–530, 2012.

[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1998.

[20] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov, "Algorithms for large scale Markov blanket discovery," in *Proc. Florida Artif. Intell. Res. Soc. Conf.*, 2003, pp. 376–380.

[21] T. Gao and Q. Ji, "Efficient Markov blanket discovery and its application," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1169–1179, May 2017.

[22] X. Wu, B. Jiang, K. Yu, C. Miao, and H. Chen, "Accurate Markov boundary discovery for causal feature selection," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4983–4996, Dec. 2020.

[23] X. Wu, Z. Tao, B. Jiang, T. Wu, X. Wang, and H. Chen, "Domain knowledge-enhanced variable selection for biomedical data analysis," *Inf. Sci.*, vol. 606, pp. 469–488, Aug. 2022.

[24] X. Wu, B. Jiang, Y. Zhong, and H. Chen, "Tolerant Markov boundary discovery for feature selection," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 2261–2264.

[25] C. Li and H. Chen, "Sparse Bayesian approach for feature selection," in *Proc. IEEE Symp. Comput. Intell. Big Data (CIBD)*, Dec. 2014, pp. 1–7.

[26] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

[27] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[28] B. H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, and M. Zhang, "Multiple reference points-based decomposition for multiobjective feature selection in classification: Static and dynamic mechanisms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 170–184, Feb. 2020.

[29] S. Han, K. Zhu, M. Zhou, and X. Cai, "Information-utilization-method-assisted multimodal multiobjective optimization and application to credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 856–869, Aug. 2021.

[30] M. Ghahramani, Y. Qiao, M. Zhou, A. O. Hagan, and J. Sweeney, "AI-based modeling and data-driven evaluation for smart manufacturing processes," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1026–1037, Jul. 2020.

[31] B. Jiang, J. Xiang, X. Wu, W. He, L. Hong, and W. Sheng, "Robust adaptive-weighting multi-view classification," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 3117–3121.

[32] X. Wu, B. Jiang, K. Yu, H. Chen, and C. Miao, "Multi-label causal feature selection," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6430–6437.

[33] Y. Wang, Z. Zhang, and Y. Lin, "Multi-cluster feature selection based on isometric mapping," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 3, pp. 570–572, Mar. 2022.

[34] B. Jiang, X. Wu, K. Yu, and H. Chen, "Joint semi-supervised feature selection and classification through Bayesian approach," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3983–3990.

[35] D. Wu, Y. He, X. Luo, and M. Zhou, "A latent factor analysis-based approach to online sparse streaming feature selection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 11, pp. 6744–6758, Nov. 2022, doi: 10.1109/TSMC.2021.3096065.

[36] X. Wu, B. Jiang, K. Yu, and H. Chen, "Separation and recovery Markov boundary discovery and its application in EEG-based emotion recognition," *Inf. Sci.*, vol. 571, pp. 262–278, Sep. 2021.

[37] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 4, pp. 619–633, Apr. 2012.

[38] S. Wang, L. L. Minku, D. Ghezzi, D. Caltabiano, P. Tino, and X. Yao, "Concept drift detection for online class imbalance learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–10.
[39] M. Harel, S. Mannor, R. El-Yaniv, and K. Crammer, "Concept drift detection through resampling," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1009–1017.
[40] Z. Yang, S. Al-Dahidi, P. Baraldi, E. Zio, and L. Montelatici, "A novel concept drift detection method for incremental learning in nonstationary environments," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 309–320, Jan. 2020.
[41] X. Wang, Q. Kang, M. Zhou, L. Pan, and A. Abusorrah, "Multiscale drift detection test to enable fast learning in nonstationary environments," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3483–3495, Jul. 2021.
[42] K. Yu, L. Liu, J. Li, W. Ding, and T. D. Le, "Multi-source causal feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2240–2256, Sep. 2020.
[43] X. Wu, B. Jiang, Y. Zhong, and H. Chen, "Multi-target Markov boundary discovery: Theory, algorithm, and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 18, 2022, doi: 10.1109/TPAMI.2022.3199784.
[44] K. Yu, L. Liu, and J. Li, "A unified view of causal and non-causal feature selection," *ACM Trans. Knowl. Discovery Data*, vol. 15, no. 4, pp. 1–46, Aug. 2021.
[45] S. Yaramakala and D. Margaritis, "Speculative Markov blanket discovery for optimal feature selection," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, pp. 809–812.
[46] Z. Ling, K. Yu, H. Wang, L. Liu, W. Ding, and X. Wu, "BAMB: A balanced Markov blanket discovery approach to feature selection," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, pp. 1–25, 2019.
[47] H. Wang, Z. Ling, K. Yu, and X. Wu, "Towards efficient and effective discovery of Markov blankets for feature selection," *Inf. Sci.*, vol. 509, pp. 227–242, Jan. 2020.
[48] K. Yu et al., "Causality-based feature selection: Methods and evaluations," 2019, *arXiv:1911.07147*.
[49] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM J. Optim.*, vol. 12, no. 2, pp. 479–502, 2002.
[50] S. Kullback, *Information Theory and Statistics*. Chelmsford, MA, USA: Courier Corporation, 1997.
[51] M. Tennant, F. Stahl, O. Rana, and J. B. Gomes, "Scalable real-time classification of data streams with concept drift," *Future Gener. Comput. Syst.*, vol. 75, pp. 187–199, Oct. 2017.
[52] C. C. Aggarwal, S. Y. Philip, J. Han, and J. Wang, "A framework for clustering evolving data streams," in *Proc. VLDB Conf.* Amsterdam, The Netherlands: Elsevier, 2003, pp. 81–92.
[53] Y. Song, J. Lu, H. Lu, and G. Zhang, "Fuzzy clustering-based adaptive regression for drifting data streams," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 3, pp. 544–557, Mar. 2020.
[54] J.-P. Pellet and A. Elisseeff, "Using Markov blankets for causal structure learning," *Mach. Learn. Res.*, vol. 9, no. 7, pp. 1295–1342, Jun. 2008.
[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[56] A. Statnikov and C. F. Aliferis, "TIED: An artificially simulated dataset with multiple Markov boundaries," in *Proc. Workshop Conf. Causality: Objectives Assessment*, vol. 6, 2010, pp. 249–256.

Xingyu Wu received the B.Sc. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China.

He has published some scientific papers in prestigious journals and conferences. His research interests include causal learning, causal inference, causal feature selection, and Markov boundary.

Dr. Wu has served as a PC Member of AAAI Conference on Artificial Intelligence (AAAI) and Conference on Empirical Methods in Natural Language Processing (EMNLP). He has served as a Reviewer for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE/CAA JOURNAL OF AUTOMATICA SINICA.

Bingbing Jiang received the B.Sc. degree from the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2014, and the Ph.D. degree from the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China, in 2019.

He is currently an Assistant Professor with Hangzhou Normal University, Hangzhou, China. He has published more than ten scientific papers, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE (TETCI), AAAI Conference on Artificial Intelligence (AAAI) and ACM International Conference on Information and Knowledge Management (CIKM). His research interests include Bayesian learning, feature selection, semisupervised learning, and multiview learning.

Dr. Jiang has served as a Reviewer for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), TNNLS, TKDE, IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and TETCI.

Xiangyu Wang received the B.Sc. degree from Donghua University, Shanghai, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Data Science, University of Science and Technology of China, Hefei, China.

His research interests include machine learning and causal learning.

Taiyu Ban received the B.Sc. degree in computer science and technology from the School of the Gifted Young, University of Science and Technology of China, Hefei, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology.

His current research interests include machine learning and knowledge engineering.

Huanhuan Chen (Senior Member, IEEE) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2004, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2008.

He is currently a Full Professor with the School of Computer Science and Technology, USTC. His research interests include neural networks, Bayesian inference, and evolutionary computation.

Dr. Chen received the 2015 International Neural Network Society Young Investigator Award, the 2012 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (bestowed in 2011 and only one paper in 2009), and the 2009 British Computer Society Distinguished Dissertations Award. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.