This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

Datum–wise Inference in Structured Environments

Sachini Piyoni Ekanayake, *Student Member, IEEE*, Daphney–Stavroula Zois, *Member, IEEE*

***Abstract*—In various application domains (e.g., health, psychology), experts use Bayesian networks to represent relationships among variables. However, these variables are not in practice directly observable, but can be instead inferred via noisy but costly features. Herein, we study the problem of datum–wise feature selection and classification in the case where the label of each data instance is described by a known Bayesian network, and features are available at a cost. The goal is to accurately classify each data instance, while keeping the feature acquisition cost minimum. To this end, we first propose a forward pass algorithm that sequentially acquires features to infer the label of each variable in the Bayesian network. During this process, the proposed algorithm uses both the acquired features and the Bayesian network relationships. In an effort to improve classification accuracy, we also devise a backward pass algorithm, which exploits Bayesian network relationships along with evidence. We discuss the computational complexity of both algorithms and experimentally assess their performance on 11 datasets. We observe that the forward pass algorithm achieves higher accuracy using a small fraction of features compared to state–of–the–art, while the backward pass algorithm enhances accuracy without acquiring additional features.**

***Impact Statement*—In traditional supervised classification, each data instance is associated with a single label (e.g., cat). However, in many real–world applications (e.g., medical diagnosis, insurance recommendation), a data instance is described by a set of related labels (e.g., physical activity and emotion, driving quality and accident severity). At the same time, access to all features is prohibitive due to cost, invasiveness, or limited resources. This work addresses the above challenges by proposing a methodology and two algorithms to perform accurate classification, while minimizing the total feature acquisition cost. The proposed methodology has the additional benefit of tailoring classification decisions to each individual data instance, not only resulting in up to 19.68% improvement in accuracy, but also decreasing by up to 88.35% the average number of acquired features. Thereby, it enables resource–efficient and accurate reasoning in non–traditional machine learning environments with a wide variety of applications including medical diagnosis, education, economics, environmental science, and transportation.**

***Index Terms*—Bayesian networks, classification, instance–wise acquisition, noisy features, sequential acquisition.**

I. Introduction

OVER the past few decades, Bayesian networks have received considerable attention finding applications in many domains (e.g., medical diagnosis [2], behavioral analysis [3], insurance recommendation [4]). There are two main reasons that explains their prevalence. First, they facilitate knowledge representation since they employ directed acyclic graphs (DAGs) to visually describe relationships between variables using nodes and edges [5]. For instance, the cancer Bayesian network [6] consists of five nodes, i.e., "*pollution*", "*smoker*", "*cancer*", "*X–ray*", and "*dyspnoea*", representing the factors that potentially contribute to the probability of having cancer. Second, Bayesian networks can also be used for reasoning in a domain of interest. For example, it is possible to identify causes of road accidents via backward analysis [7].

In supervised machine learning, Bayesian networks have been typically used to represent class–feature dependencies, where the goal is to classify a data instance in one out of N classes [8]. For example, Naive Bayes and its extensions (e.g., Tree–Augmented Naive Bayes) can be graphically represented using Bayesian networks. In this context, a single Bayesian network node represents the classification variable, while the rest represent noisy features (see Fig. 1(a)). On the other hand, Bayesian networks have been employed to describe relationships between multiple classification variables [6], [9]. In this case, the objective is to infer their values by exploiting the associated relationships (see Fig. 1(b)). For instance, a recommendation system is proposed in [4] to predict insurance products for customers. Note that such classification variables are assumed to be *fully observable*. This is typically not the case in many real–world applications, where classification variables are observed via noisy features (see Fig. 1(c)). For example, the emotion and personality characteristics of an individual, which are only observable through noisy galvanic skin response, electrocardiogram, and electroencephalogram data [10], are related and thus, can be represented by a Bayesian network with two classification variables.

In many real–world applications (e.g., medical diagnosis, planetary imaging), features are acquired at a cost that captures the relevant effort needed to access them. At the same time, using different features for classification has a different effect on the accuracy of the resulting prediction (e.g., in the medical domain, tests may be intrusive, uncomfortable and/or costly, but may impact accurate and timely diagnosis). As a result, feature selection in this context has received considerable attention [11]–[16]. Depending on the stage that feature selection takes place, relevant methods can be roughly categorized either as *streaming* [11], [12], or *dynamic instance–wise* [13]–[16]. The former methods assume that feature selection takes place during *training*, where features arrive one at a time or in batches, and the *same* selected feature subset is used for classification during testing. In contrast, the latter methods

Manuscript received (date to be filled by Editor). This material is based upon work supported by the National Science Foundation under Grants ECCS–1737443 & CNS–1942330.

S. P. Ekanayake and D.–S. Zois are with the Department of Electrical and Computer Engineering, University at Albany, State University of New York, Albany, NY (email: sekanayake@albany.edu, dzois@albany.edu).

Parts of this paper have previously appeared in [1]. This material significantly extends our previous work by introducing *forward* and *backward pass* algorithms followed by a complexity analysis. Moreover, a detailed set of experiments is conducted that includes practical considerations, sensitivity analysis, and comparison with state–of–the–art baselines.

The Associate Editor coordinating the review of this manuscript and approving it for publication was (name to be filled by Editor).

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

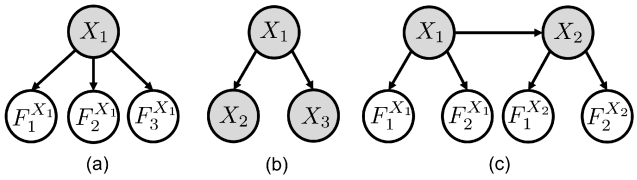


Fig. 1: Graphical illustration of: (a) a Bayesian network of a single classification variable X_1 observed via features $F_1^{X_1}, F_2^{X_1}, F_3^{X_1}$, (b) a Bayesian network of three observable variables X_1, X_2, X_3, and (c) a Bayesian network of two classification variables X_1, X_2 observed via features $F_1^{X_1}, F_2^{X_1}$ and $F_1^{X_2}, F_2^{X_2}$, respectively.

perform dynamic feature selection during *testing*, essentially using *different features* to classify each data instance. To the best of our knowledge, *all* such works consider supervised classification of a *single* variable observed via noisy features.

In this work, we consider *supervised classification* of *structured data instances*, i.e., the label of each data instance corresponds to a set of classification variables with the relationships between them captured by a *known Bayesian network*. We particularly focus on the case where features are not freely available, but instead are sequentially acquired at a cost one at a time during *testing*. The objective is two–fold: *accurately classify each data instance while keeping the total feature acquisition cost minimum*. In summary, our contributions are:

1) We present a novel variant of the dynamic instance–wise feature selection problem [13], where multiple interrelated variables need to be classified.
2) To address this problem, we propose an algorithm that assigns a label to each variable in the Bayesian network by acquiring the appropriate optimum number of features per instance. The algorithm also propagates the labels through the Bayesian network in a *forward pass*, incorporating their effect in the inference process.
3) We also present a *backward pass* algorithm that incorporates evidence into the inference process, thus improving accuracy.
4) We analyze the computational complexity of both algorithms, and experimentally assess their performance on 11 real–world datasets. We also compare them with the state–of–the–art, showing their effectiveness and generalizability on a variety of applications.

The remainder of this paper is organized as follows. Section II summarizes relevant prior work. Section III describes the problem of classifying structured data instances and provides the optimum solution. Sections IV and V describe the proposed forward and backward pass algorithms that exploit the optimum solution and the Bayesian network relationships to perform feature selection and classification. Section VI presents detailed experiments that assess the performance of the proposed algorithms and discuss relevant findings. Finally, Section VII concludes the paper and briefly describes future research directions. For reproducibility purposes, the source code of the proposed algorithms will become available upon acceptance of this manuscript.

II. Related Work

In this section, we briefly review the most relevant literature. For clarity, consider a supervised classification setting with variables $X \triangleq \{X_1, X_2, \ldots, X_n\}$ and features $F \triangleq \{F_1, F_2, \ldots, F_K\}$. Here, we use n to denote the number of categorical variables, and K to represent the number of features. The goal is to infer the value of X using F, where each $X_i \in X$ takes one out of $N_i, i = 1, \ldots, n$, possible values. Depending on the values that n and N_i take, we have the following cases: (i) *binary classification*: $n = 1$ and $N_1 = 2$, (ii) *multi–class classification*: $n = 1$ and $N_1 > 2$, (iii) *multi–label classification*: $n > 1$ and $N_i = 2, i = 1, \ldots, n$, and (iv) *multi–dimensional classification*: $n > 1$ and $N_i > 2, i = 1, \ldots, n$. The problem we study in this paper is closely related to multi–dimensional classification (MDC).

Historically, MDC problems have been addressed by the *independent classifier* (IC) and the *power–set classifier* (PC) approaches [17]. The former approaches independently learn the value of each variable in X using a set of standard multi–class classifiers (e.g., multi–label music classification via the Binary Relevance method [18]), the results of which are then combined to obtain the final result. Even though the IC approach is computationally efficient, it fails to capture the relationships between variables in X resulting in a significant loss in accuracy. On the other hand, the latter approaches convert the original MDC problem into a single multi–class classification problem by considering a single vector variable $[X_1, X_2, \ldots, X_n]$. As a result, they succeed in directly accounting for the relationships between variables in X at the expense of computational complexity, which increases as the number of variables and their values grow [19]. In this context, the proposed approach combines the best of the above two extremes. Namely, it indeed classifies each variable separately, but at the same time, it exploits the structure of the Bayesian network that describes the relationships between the variables by propagating classification decisions through the network. The proposed approach has the additional flexibility of dynamically selecting which features to use, not only boosting accuracy but also saving on resources.

In an effort to reduce computational complexity while ensuring that relationships between variables are appropriately accounted for, various other MDC approaches have been proposed. These can be roughly categorized into *classifier–chains* [19], *variable partitioning* [20], [21], and *probabilistic graphical model* methods [22]–[24]. Next, we briefly review the most important relevant works.

Inspired by multi–label classification [25], the classifier chains (CC) approach extends the IC approach by considering relationships between variables in X captured by a chain. During classification, the value learned for each variable in X is used as an additional source of information for the next variable in the chain. Since the fixed variable ordering in the chain considerably affects accuracy, ensembles of classifier chains (ECC) have been proposed [25], where each member in the ensemble is trained with a random chain ordering and on a random subset of the training dataset. The Bayesian chain classifier (BCC) approach [19] can be considered as

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

an extension of CC, where the goal is to improve accuracy without incurring additional computational complexity by considering meaningful variable orderings in the chain. To this end, BCC first learns relationships from data in the form of a Bayesian network, thus restricting the possible variable orderings in the chain. In the second step, a chain classifier is built such that the chain ordering is consistent with the previously learned Bayesian network structure. Similar to our work, the above methods account for relationships between variables in X through the use of an appropriate mechanism (e.g., chain, Bayesian network). However, in contrast to our work, classification decisions are based on the entire feature set F for all variables in X, not on the most cost–efficient and "informative" features per variable. At the same time, all such methods require access to a base classifier (e.g., Naive Bayes [19]), the performance of which affects the overall accuracy achieved. Contrary to this, the proposed approach jointly optimizes feature selection and classification, hence selecting the classification decision that gives the best accuracy when using a specifically selected feature subset.

In an effort to achieve comparable accuracy but avoid the large computational complexity of the PC approach, variable partitioning has been proposed [20], [21]. The main idea is to partition variables in X into groups (also known as super–classes [20]) based on relevant conditional dependence information, and use the PC approach on top of these groups. Recently, Jia et al. [21] proposed a two–step grouping approach that involves first computing relevant counting statistics from an unseen data instance's k nearest neighbors (kNN) in the training dataset, and then performing maximum *a posteriori* (MAP) inference based on these statistics for each possible pair of class spaces. Similar to our approach, the above line of work accounts for relationships between variables in X. Yet, relationship information is used to reduce the size of the classification space by carrying out classification in subsets of X. At the same time, such approaches use a base classifier and perform classification using the whole feature set F. In contrast, the proposed approach explicitly models variable relationships in terms of a Bayesian network and propagates classification decisions over this structure. At the same time, it explicitly determines both the optimum number of features and optimum classification strategy for each variable in X. Thus, unlike [20], [21], there is no need to employ a base classifier and use the whole feature set F, which affect performance and hinder explainability of the classification decisions.

Finally, in *multi–dimensional Bayesian network classification* (MBC) [22]–[24], relationships between variables in X are modeled using a Bayesian network model (see [24] for a comprehensive survey). Such approaches learn the underlying unknown Bayesian network structure between variables in X and features in F, and then perform inference to compute the values of variables in X. Since the Bayesian network can be split into three main subgraphs (i.e., class, feature, bridge) with potentially different structures, different MBC families (e.g., *tree–tree*) have been designed. Irrespective of the structures, however, the associated computational complexity of these approaches is high. Furthermore, it has been experimentally shown that BCC achieves better accuracy than them. In an effort to limit computational complexity, Bielza et al. [22] theoretically formulate the notion of decomposable *class–bridge* MBC, where maximal connected components that do not share children are identified. A variable ordering approach is also proposed to efficiently navigate through all possible variable combinations, but no overall algorithmic solution is provided. A conditional tree–structured Bayesian network (CTBN) is proposed in [23] as an alternative way of addressing MBC. In this context, variable relationships are modeled as a directed tree, where the set of all features is treated as a common parent for all classification variables. Classification is then conducted based on MAP estimation using exact inference. The main difference between prior work on MBC and our proposed approach is that the former methods use the entire feature set F to perform classification, while the latter approach dynamically selects the optimum number of features per classification variable in X to balance accuracy with the cost of feature acquisition. The benefits of such an approach are: (1) improved accuracy using fewer features, (2) being able to reason about the classification outcomes, and (3) lower computational complexity than performing MBC.

III. Problem Description & Solution

In this section, we describe the problem of classifying structured data instances and provide the basis of the proposed approach discussed in Section IV. Specifically, we start by introducing the details of the problem and stating our assumptions. Next, we define an optimization problem for each variable in the Bayesian network, where the goal is to limit the number of features used for classification accounting for the effect on the classification performance. Finally, we summarize the solution to the optimization problem.

A. Description

We consider a *supervised classification* setting, in which each data instance is associated with n categorical variables $X_1, X_2, \ldots, X_n$, each of which can take multiple values. These variables are related and thus described by a Bayesian network $\mathcal{G} = (X, E)$. We note that $X \triangleq \{X_1, X_2, \ldots, X_n\}$ and the set E represents relationships between variables in X as directed edges (e.g., $X_\ell \rightarrow X_m$ indicates that the associated variables are related). According to the chain rule [5], the joint probability distribution P over the random variables in X is:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa_{X_i}^{\mathcal{G}}), \tag{1}$$

where $P(X_i|Pa_{X_i}^{\mathcal{G}})$ denotes the conditional probability of variable X_i given its parents $Pa_{X_i}^{\mathcal{G}}$ in graph $\mathcal{G}$. Furthermore, we have access to labeled training data, i.e., each data instance is represented by a feature vector described by $F \triangleq \{F_1^{X_1}, \ldots, F_{K_1}^{X_1}, F_1^{X_2}, \ldots, F_{K_2}^{X_2}, \ldots, F_1^{X_n}, \ldots, F_{K_n}^{X_n}\}$, where $F_k^{X_i}, k = 1, 2, \ldots, K_i$, denotes the kth feature associated with variable X_i, and its label is an appropriate n–dimensional class vector described by X. At the same time, during testing, features in F are not readily available (e.g., due to large feature space or cost of feature acquisition), but

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

become available one at a time based on a *fixed ordering*. In this context, the goal is to learn functions that dynamically select which features in F to acquire, and use the acquired features to assign a vector of values to the variables in X. The ideal functions will of course limit the number of features acquired per variable to restrict the total feature acquisition cost, while maintaining accuracy within acceptable levels. To simplify the problem of classifying structured data instances, we adopt the following three crucial assumptions:

(A1) We assume that the Bayesian network structure is known and given.

(A2) We assume that the features $F_k^{X_i}, k = 1, 2, \ldots, K_i$, associated with variable $X_i \in X$ are conditionally independent given X_i.

(A3) We assume that features $F_k^{X_i}, k = 1, 2, \ldots, K_i$, are ordered based on a certain performance measure (c.f. Section VI-B), and such ordering is given for each variable $X_i \in X$. As a side note, it is possible that features are ordered differently for each variable $X_i \in X$.

B. Optimization Problem Formulation

For each categorical variable $X_i, i = 1, 2, \ldots, n$, in the Bayesian network $\mathcal{G}$, we define two random variables, R_i and D_{R_i}. We use random variable $R_i \in \{0, 1, \ldots, K_i\}$ to denote the last feature acquired from the ordered set $F^{X_i} \triangleq \{F_1^{X_i}, \ldots, F_{K_i}^{X_i}\}$ before proceeding with a classification decision for categorical variable X_i. Furthermore, we use random variable D_{R_i} to denote the classification decision for categorical variable X_i. Since X_i takes one out of N_i possible values, $D_{R_i} \in \{1, 2, \ldots, N_i\}$. To acquire each feature $F_k^{X^i}$, we expense cost $e_k^i, k = 1, 2, \ldots, K_i$. At the same time, we use the term M_{lm}^i to capture the cost of classifying X_i as $C_l^{X_i}, l = 1, 2, \ldots, N_i$, when its true class is $C_m^{X_i}, m = 1, 2, \ldots, N_i$. In line with our goal in Section III-A, we define the cost function below:

$$J(R_i, D_{R_i}) = \mathbb{E}\left[\sum_{k=1}^{R_i} e_k^i\right] + \sum_{l=1}^{N_i}\sum_{m=1}^{N_i} M_{lm}^i P(D_{R_i} = l, C_m^{X_i}), \quad (2)$$

where $P(D_{R_i} = l, C_m^{X_i})$ denotes the probability of assigning class $C_l^{X_i}$ to categorical variable X_i, even though its true class is $C_m^{X_i}$. The first expression in Eq. (2) represents the total cost of acquiring R_i features to classify X_i. On the other hand, the second expression in Eq. (2) represents the cost of the classification decision D_{R_i}. Thus, the goal is to minimize the cost function in Eq. (2) with respect to both R_i and D_{R_i}.

C. Optimum Solution

We solve the optimization problem in Section III-B in two steps. In the first step, we determine the optimum decision $D_{R_i}^*$ for fixed given R_i features. In the second step, we determine the optimum R_i^* features by minimizing $J(R_i)$, which is the reduced cost function resulting from the first step.

We begin by considering the *a posteriori* probability vector $\pi_k \triangleq [\pi_k^1, \pi_k^2, \ldots, \pi_k^{N_i}]^T$. Each term $\pi_k^m \triangleq P(C_m^{X_i} | F_1^{X_i}, \ldots, F_k^{X_i})$ represents the *a posteriori* probability of the mth class, $m = 1, 2, \ldots, N_i$, when k out of K_i features associated with categorical variable X_i have been acquired. We assume that initially, $\pi_0 \triangleq [p_1, p_2, \ldots, p_{N_i}]^T$, where $p_m \triangleq P(C_m^{X_i}), m = 1, 2, \ldots, N_i$. The *a posteriori* probability π_k^m can be recursively updated as more features are sequentially acquired using Bayes' rule:

$$\pi_k^m = \frac{P(F_k^{X_i} | C_m^{X_i}) \pi_{k-1}^m}{P(F_k^{X_i} | C_1^{X_i}) \pi_{k-1}^1 + \ldots + P(F_k^{X_i} | C_{N_i}^{X_i}) \pi_{k-1}^{N_i}}. \quad (3)$$

Furthermore, we can rewrite the second term of Eq. (2) in terms of the *a posteriori* probability and the indicator function $\mathbb{1}_A$ (i.e., $\mathbb{1}_A \triangleq 1$ when event A occurs, and 0 otherwise) as:

$$J(R_i, D_{R_i}) = \mathbb{E}\left[\sum_{k=1}^{R_i} e_k^i + \sum_{l=1}^{N_i}\sum_{m=1}^{N_i} M_{lm}^i \pi_{R_i}^m \mathbb{1}_{D_{\{R_i = l\}}}\right]. \quad (4)$$

Starting from Eq. (4), we can show that the optimum classification strategy $D_{R_i}^*$ for any fixed and given feature selection strategy R_i has the following form [13]:

$$D_{R_i}^* = \underset{1 \le l \le N_i}{\operatorname{argmin}} \left[(\mathbf{M}_l^i)^T \pi_{R_i}\right], \quad (5)$$

where $\mathbf{M}_l^i \triangleq [M_{1l}^i, M_{2l}^i, \ldots, M_{N_i l}^i]^T$. Thus, we rewrite the cost function in Eq. (4) as follows:

$$J(R_i) = \mathbb{E}\left[\sum_{k=1}^{R_i} e_k^i + g(\pi_{R_i})\right], \quad (6)$$

where $g(\pi_{R_i}) \triangleq \min_{1 \le l \le N_i}[(\mathbf{M}_l^i)^T \pi_{R_i}]$. To determine the optimum feature selection strategy R_i^*, we minimize the cost function in Eq. (6) using dynamic programming [13] as:

$$L_k(\pi_k) = \min\left[g(\pi_k), \tilde{L}_k(\pi_k)\right], k = 0, \ldots, K_i - 1, \quad (7)$$

where

$$\tilde{L}_k(\pi_k) = e_{k+1}^i + \sum_{F_{k+1}^{X_i}} L_{k+1}(\pi_{k+1}) \Delta_{k+1}^T(F_{k+1}^{X_i}) \pi_k, \quad (8)$$

with $\Delta_k(F_k^{X_i}) \triangleq [P(F_k^{X_i} | C_1^{X_i}), \ldots, P(F_k^{X_i} | C_{N_i}^{X_i})]^T$ and $L_{K_i}(\pi_{K_i}) = g(\pi_{K_i})$. We observe that there are $K_i + 1$ stages for the resulting dynamic programming equations, since there are K_i features in total for each X_i.

IV. Proposed Approach

In this section, we propose an algorithm that exploits the optimum solution in Section III to classify all $X_i, i = 1, 2, \ldots, n$, in the Bayesian network $\mathcal{G}$ using the least number of features per categorical variable.

A. Instancewise Structured Environments Classification Algorithm

We start by explaining the main idea behind the proposed approach, referred to as Instance–wise Structured Environments Classification (ISEC) algorithm. Specifically, ISEC initializes the *a posteriori* probability vector π_0 for each categorical variable $X_i, i = 1, 2, \ldots, n$, in the Bayesian network $\mathcal{G}$. Next, ISEC uses Eqs. (7) and (8) to determine the optimum number of features needed to reach an accurate classification decision for each categorical variable in the Bayesian network $\mathcal{G}$. Finally, ISEC uses the optimum number

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

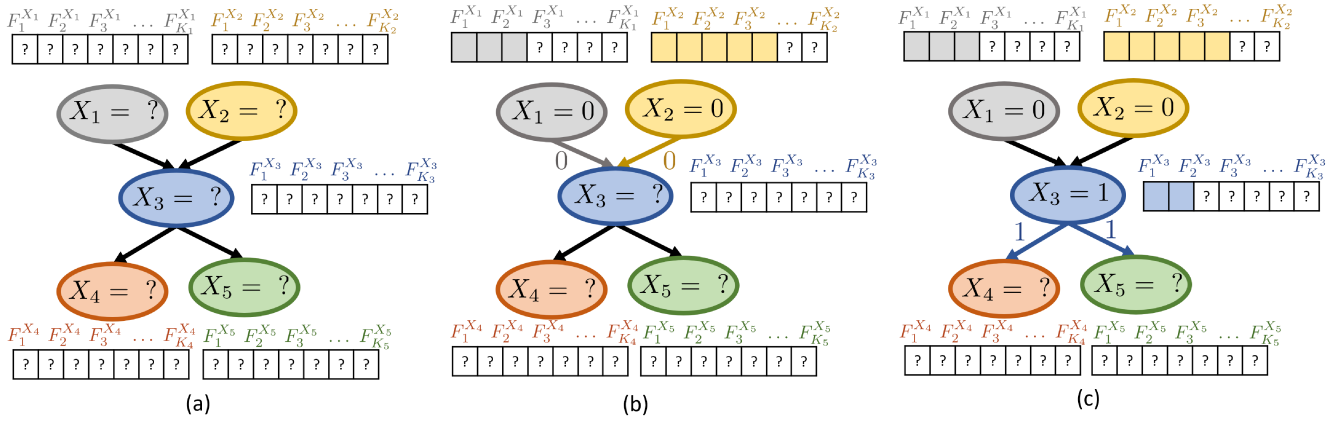


Fig. 2: Illustration of ISEC's operation during testing. (a) Original Bayesian network, (b) feature selection and classification for variables of indegree 0, (c) feature selection and classification for subset of variables of indegree greater than 0. Acquired features at each round of ISEC are highlighted. ISEC classifies X_1, X_2, and X_3 using $3, 5$, and 2 features, respectively.

Algorithm 1 ISEC

Input: test dataset D_{test}, numerical solutions $\{\mathcal{M}_i, \mathcal{M}_i^r\}$, topological ordering $O_{\mathcal{G}}$, indegree 0 nodes X^0, indegree > 0 nodes X^+
Output: classification decisions Val_X
for each data instance s in D_{test} **do**
  for each node $X_i \in O_{\mathcal{G}}$ **do**
    if node $X_i \in X^0$ **then**
      $\mathcal{M} = \mathcal{M}_i$
    else
      $\alpha = Pa^{\mathcal{G}}_{X_i}$
      $pred = Val_X(s, \alpha)$
      Let $pred$ be the r^{th} value combination of α
      $\mathcal{M} = \mathcal{M}_i^r$
    end if
    for each feature $F_k^{X_i}$ **do**
      Acquire feature and update π_k using Eq. (3)
      Get stopping cost $g^i(\pi_k)$ & continuing cost $\tilde{L}^i_k(\pi_k)$ from $\mathcal{M}$
      if $g^i(\pi_k) \leq \tilde{L}^i_k(\pi_k)$ **then**
        Find optimum decision $D^*_{R_i}$ using Eq. (5)
        Set $Val_X[s, i] = D^*_{R_i}$
        Terminate feature acquisition process
      end if
    end for
  end for
end for
Return: Val_X

of features along with the optimum classification strategy in Eq. (5) and the Bayesian network structure to classify each categorical variable $X_i, i = 1, 2, \ldots, n$. We discuss the training and testing phases of ISEC next.

During training of ISEC, we numerically solve Eqs. (5), (7), and (8) for each $X_i, i = 1, 2, \ldots, n$, in the Bayesian network $\mathcal{G}$. Specifically, for each categorical variable X_i, we generate a $K_i \times d$ matrix by quantizing the interval $[0, 1]$ such that $\sum_{m=1}^{N_i} \pi_k^m = 1$, and evaluate the above equations. We denote the resulting numerical solutions associated with categorical variables of indegree 0 as $\mathcal{M}_i, i = 1, 2, \ldots, z$, while we use the notation $\mathcal{M}_i^r, i = z+1, \ldots, n, r = 1, \ldots, c$, to represent the numerical solutions associated with categorical variables of indegree > 0. Here, z denotes the total number of categorical variables with indegree 0 and c represents the total number of value combinations of parent nodes $\text{Pa}^{\mathcal{G}}_{X_p}$.

During testing, ISEC starts the feature selection and classification process from categorical variables of indegree 0. Specifically, it uses the numerical solutions $\mathcal{M}_i$ to dynamically select features for each $X_i, i = 1, 2, \ldots, z$, *separately*. If the cost of continuing the feature selection process is less than the cost of reaching a classification decision, ISEC keeps acquiring more features and updates the *a posteriori* probability vector accordingly using Eq. (3). It repeats the process until either all features associated with a categorical variable X_i are acquired or if it decides that a subset of features is sufficient for reaching an accurate classification decision. In that case, ISEC uses the numerical solutions $\mathcal{M}_i$ to assign a classification decision to each categorical variable $X_i, i = 1, 2, \ldots, z$, of indegree 0. Next, ISEC moves on to categorical variables $X_i, i = z+1, \ldots, n$, of indegree > 0 for which it has already classified their parent variables. ISEC uses the numerical solutions $\mathcal{M}_i^r, r = 1, \ldots, c$, and these classification decisions to dynamically select features for each such categorical variable until it reaches a classification decision. It repeats this process until all categorical variables in the Bayesian network $\mathcal{G}$ have been assigned a classification decision. Algorithm 1 outlines this process, while Fig. 2 illustrates ISEC on a Bayesian network $\mathcal{G}$ of five binary variables.

B. Complexity Analysis

We discuss the computational complexity of ISEC during testing next. We observe that acquiring a single feature value requires $\mathcal{O}(1)$, and updating the *a posteriori probability vector* using Eq. (3) requires $\mathcal{O}(N_i)$ since it involves the dot product of two N_i–dimensional vectors. Selecting between continuing or terminating the feature acquisition process in Eq. (7) using pre–computed numerical solutions requires $\mathcal{O}(1)$. In the worst

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

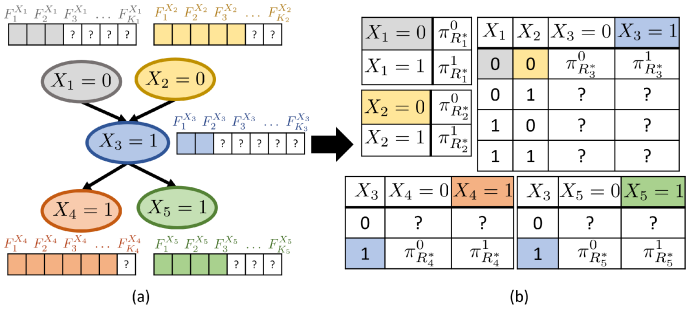


Fig. 3: (a) Graphical representation of ISEC's classification decisions and number of acquired features, (b) CPD tables for data instance s after running ISEC.

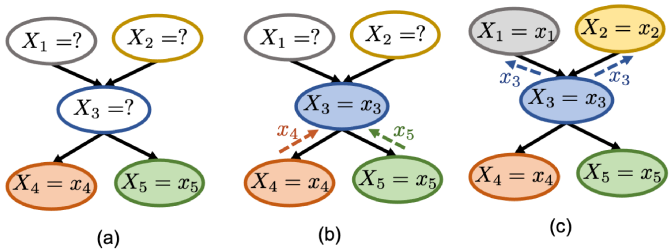


Fig. 4: Illustration of IBSEC's operation. (a) Original Bayesian network with evidence for X_4 and X_5, (b) Classification of X_3, (c) Classification of X_1 and X_2.

case, ISEC carries out these comparisons for all K_i features. Thus, the total computational complexity is $\mathcal{O}(K_i N_i)$. Finally, ISEC carries out the classification process via Eq. (5) requiring $\mathcal{O}(N_i^2)$ computational complexity. Thus, the total computational complexity of ISEC for feature acquisition and classification of a single categorical variable X_i is $\mathcal{O}(K_i N_i + {N_i}^2)$. Since there are in total n such categorical variables, the time complexity of ISEC during testing is $\mathcal{O}(n(K_i N_i + N_i^2))$.

V. The Effect of Evidence

In this section, we propose to improve ISEC's classification decisions through backward inference. Specifically, assuming that the values of categorical variables of outdegree 0 are given as evidence, the goal is to determine the values of the remaining categorical variables in the Bayesian network $\mathcal{G}$.

A. Instance–wise Backward Structured Environments Classification Algorithm

As discussed in Section IV, ISEC employs the optimum classification strategy $D^*_{R_i}$ in conjunction with the optimum number R_i^* of features to classify each categorical variable $X_i, i = 1, 2, \ldots, n$, in the Bayesian network $\mathcal{G}$. At that time, the *a posteriori* probability $\pi_{R_i^*} \triangleq P(X_i | F_i^*, Pa^{\mathcal{G}}_{X_i})$, where $F_i^* = \{F_1^{X_i}, \ldots, F_{R_i^*}^{X_i}\}$, has been computed only for the values of parents propagated through the Bayesian network $\mathcal{G}$. However, in order to perform backward inference, we need to have access to the complete posterior probability distribution (CPD) table for each data instance s incorporating both the dependence of each categorical variable on its parents and the optimum number R_i^* of features acquired for each variable $X_i, i = 1, 2, \ldots, n$. We can reconstruct these CPD tables for all parents values combinations by running ISEC. Specifically, we observe that ISEC has determined the CPD tables for all categorical variables of indegree 0. Thus, during training, we only need to reconstruct the CPD tables for categorical variables of indegree > 0 by running ISEC for all the relevant parents values combinations. Fig. 3 illustrates ISEC's hypothetical classification decisions $D^*_{R_i}$ using R_i^* features for the Bayesian network $\mathcal{G}$ of Fig. 2 and the missing values of the CPD table for a data instance s. We observe that the CPD tables for X_1 and X_2 have already been determined by ISEC. Thus, we proceed with determining the CPD table of X_3 by running ISEC for the remaining X_1 and X_2 value combinations i.e., $\{(0,1), (1,0), (1,1)\}$. Similarly, we determine the CPD tables of X_4 and X_5.

Algorithm 2 IBSEC

Input: CPD tables $\mathcal{C}_{\mathcal{G}}$, outdegree 0 nodes X^0_-, evidence for outdegree 0 nodes $\xi^{x^0_-}$, reverse topological order $O^-_{\mathcal{G}}$
for each data instance s in D_{test} **do**
  for each node $X_i \in O^-_{\mathcal{G}}$ **do**
    if node $X_i \in X^0_-$ **then**
      $X_i = \xi^{x^0_-}$
    else
      Obtain $P(X_i, Ch^{\mathcal{G}}_{X_i}, F_i^*)$ & $P(Ch^{\mathcal{G}}_{X_i}, F_i^*)$ using $\mathcal{C}_{\mathcal{G}}$
      $P(X_i | Ch^{\mathcal{G}}_{X_i}, F_i^*) = \frac{P(X_i, Ch^{\mathcal{G}}_{X_i}, F_i^*)}{P(Ch^{\mathcal{G}}_{X_i}, F_i^*)}$
      $X_i = \text{argmax}_x P(X_i = x | Ch^{\mathcal{G}}_{X_i}, F_i^*)$
    end if
  end for
end for
Return: classification decisions for all X_i (except X^0_-)

During testing, we start from the categorical variables of outdegree 0, the values of which are given as evidence. The proposed approach, referred to as Instance–wise Backward Structured Environments Classification (IBSEC) algorithm, performs backward inference by using the evidence values along with the CPD tables determined during training. It carries out MAP inference to find the most likely assignment for the parent categorical variables given the evidence values through variable elimination [5]. IBSEC repeats this process for the remaining categorical variables in the Bayesian network $\mathcal{G}$ by visiting them in reverse topological ordering of $\mathcal{G}$. Fig. 4 illustrates IBSEC on the Bayesian network of Fig. 2, while Algorithm 2 outlines this process[1].

B. Complexity Analysis

Next, we discuss the computational complexity of IBSEC during testing. Since MAP inference is carried out during variable elimination, the complexity of computing $P(X_i, Ch^{\mathcal{G}}_{X_i}, F_i^*)$ is $\mathcal{O}(nW_{\text{max}})$ [5]. Here, W_{max} is the maximum number of entries in a factor[2] of the Bayesian network

[1]The children of a random variable X_i are denoted as $Ch^{\mathcal{G}}_{X_i}$.

[2]The joint distribution is represented as a product of factors, where each factor is a conditional probability of the form $P(X_i | Pa^{\mathcal{G}}_{X_i})$ [5].

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

TABLE I: Datasets description (number of instances (S), number of variables (n), number of classes per each variable (N_i), number of features (K_i)).

Dataset	S	n	N_i	K_i	Domain
Edm	154	2	3	16	Machines
Voice	3136	2	2 to 4	19	Voice
Jura	359	2	4 to 5	9	Geography
Song	785	3	3	98	Music
Flare	323	3	2 to 4	10	Solar flares
Student	649	3	2	30	Education
Emotion	593	6	2	72	Music
Child	1000	3	2	17	Medical
Hepar2	1000	3	2	67	Medical
Sachs	1000	2	3	9	Biology
Insurance	1000	2	3 to 4	25	Insurance

$\mathcal{G}$. The complexity of computing the probability $P(Ch^{\mathcal{G}}_{X_i}, F_i^*)$ of evidence is $O(N_i)$, since it involves variable elimination over a single variable X_i. Since N_i divisions are carried out, the total complexity of computing $P(X_i|Ch^{\mathcal{G}}_{X_i}, F_i^*)$ is $\mathcal{O}(nW_{\max} + 2N_i)$. Since X_i takes N_i values, carrying out MAP inference via variable elimination incurs computational complexity $\mathcal{O}(nW_{\max}+3N_i)$. Since there are $\Gamma \triangleq n-\beta$ nodes, where β is the number of outdegree 0 nodes in $\mathcal{G}$, the time complexity of IBSEC during testing is $\mathcal{O}(\Gamma(nW_{\max} + 3N_i))$.

VI. Experimental Results

In this section, we conduct a number of experiments on a variety of datasets from different domains to assess the performance of ISEC and IBSEC and illustrate their operation. All experiments are conducted on a PC with Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz with 16 GB memory. All reported results are five–fold cross validated.

A. Datasets

Table I presents the 11 datasets used in our experiments including a description of their characteristics and the domain they represent. Edm [26], Voice [27], Jura [28], Song [29], Solar–flare (Flare) [30] and Emotion [18] are typical MDC datasets[3]. Following the standard approach in MDC literature [24], we also employ forward sampling [5] to generate four additional datasets of $1,000$ instances[4] each based on the Child [31], Hepar2 [32], Sachs [33], and Insurance [34] Bayesian networks. In each case, we randomly split the nodes between classification variables and features, keeping the number of classification variables low (i.e., 2 or 3). Finally, we preprocess the Student performance dataset (Student) [35] such that the 3 classification variables G_1, G_2, G_3 representing three period grades are binary, i.e., we set $G_i = 1, i \in \{1, 2, 3\}$, if the corresponding student score is $\geqslant 11$.

B. Training

We have assumed that the Bayesian network structure that represents relationships between variables in X is known and given (see Section III-A). Thus, we employ the well–known Chow–Liu algorithm [36] to learn the underlying Bayesian network structure for those datasets that we do not have access to such information. For the Student dataset, we construct the relevant Bayesian network by first carrying out correlation–based analysis. Specifically, we observed that variable G_3 exhibits a strong correlation with variables G_1 and G_2 [35]. Next, we take into account the immediate effect of cause variables to generate directed edges [9]. Thus, the resulting Bayesian network includes a set $X \triangleq \{G_1, G_2, G_3\}$ of 3 nodes with two directed edges $E \triangleq \{(G_1, G_3), (G_2, G_3)\}$ (see also Fig. 7). The variable relationships for the Insurance and Sachs datasets are obtained from the original Bayesian network.

During training, we estimate the prior probabilities $P(X_i)$ and conditional probabilities $P(F_k^{X_i}|C_m^{X_i}), k = 1, 2, \ldots, K_i, m = 1, 2, \ldots, N_i$. Specifically, $\hat{P}(F_k^{X_i}|C_m^{X_i}) = \frac{S_{k,m}+1}{S_m+B}$, where $S_{k,m}$ represents the number of instances that are in class $C_m^{X_i}$ and feature $F_k^{X_i}$ takes a specific value, S_m represents the total number of instances in class $C_m^{X_i}$, and B represents the number of bins considered. Similarly, $\hat{P}(X_i) = \frac{S_m+1}{\sum_{m=1}^{N_i} S_m+N_i}$. For each variable $X_i, i = 1, 2, \ldots, N_i$, we compute the sum of type I and II errors and scale the result by the feature cost e_k^i of the kth feature. We use this performance indicator to order features for each variable X_i separately, such that low cost and accurate features appear earlier in the order. Finally, since there are no restrictions imposed by the datasets, we assume that each variable has access to the same set of features.

C. Performance Metrics

In this work, we use *mean accuracy*, *global accuracy*, and *average number of features* acquired as performance metrics. We define mean accuracy and global accuracy as:

$$\text{Mean Accuracy (MA)} \triangleq \frac{1}{n \times S} \sum_{i=1}^{n} \sum_{s=1}^{S} \mathbb{1}_{x_{is}=\hat{x}_{is}}, \tag{9}$$

$$\text{Global Accuracy (GA)} \triangleq \frac{1}{S} \sum_{s=1}^{S} \mathbb{1}_{\mathbf{x_s}=\hat{\mathbf{x}}_\mathbf{s}}, \tag{10}$$

where n represents the total number of variables in the Bayesian network, S represents the total number of instances and we use $\hat{(\cdot)}$ to indicate predicted values. The former performance metric represents the effect of separately predicting the values of the variables in the Bayesian network, while the latter represents the effect of joint prediction.

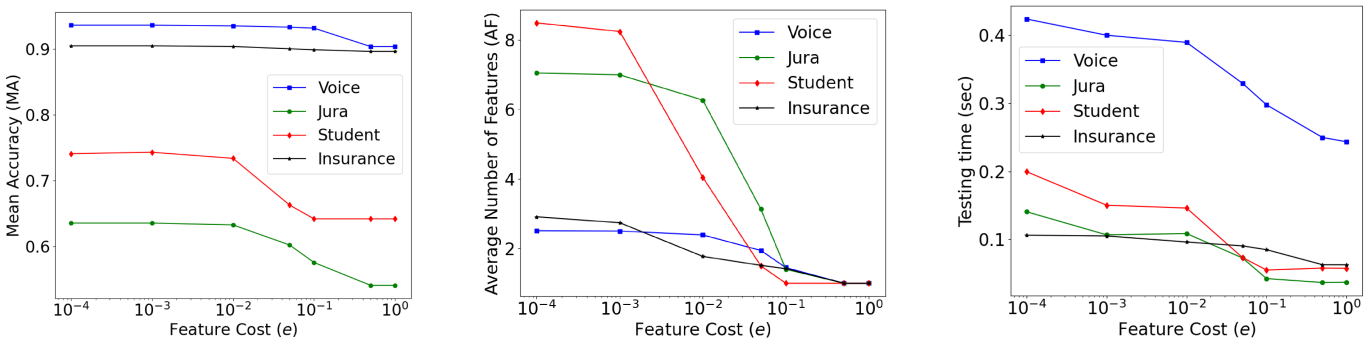
D. Sensitivity Analysis

In this subsection, we assess the effect of the feature cost e_k^i and the number B of bins on the performance of ISEC. Specifically, Fig. 5 illustrates MA, average number of features, and testing time of ISEC as the feature cost e_k^i varies. For simplicity, we assume that all features incur the same cost, i.e., $e_k^i = e, \forall k, i$, and misclassification costs are $M_{lm}^i = 1, \forall l \neq m, M_{ll}^i = 0, l, m = 1, \ldots, N_i$. We observe that different feature costs yield different accuracy, and acquiring more features typically results in better accuracy on the premise that such features are informative. However,

[3]The preprocessed versions of Voice, Jura, and Song [29] are used herein.
[4]Accuracy values stabilize at around $1,000$ data instances.

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616



(a) (b) (c)

Fig. 5: Variation of (a) MA, (b) average number of features, and (c) testing time of ISEC as a function of feature cost $e \in \{0.0001, 0.001, 0.01, 0.05, 0.1, 0.5, 1.0\}$ for Voice, Jura, Student and Insurance datasets.

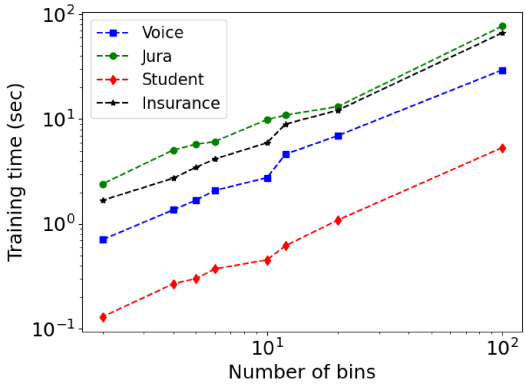


Fig. 6: Variation of training time (sec) of ISEC as a function of number $B \in \{2, 4, 5, 6, 10, 12, 20, 100\}$ of bins for Voice, Jura, Student and Insurance datasets.

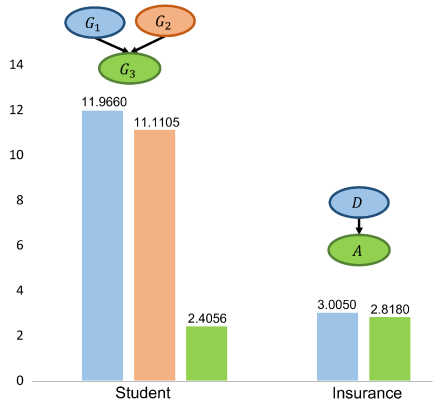


Fig. 7: Average number of features acquired per variable in Student and Insurance (D: Driving Quality, A: Accident) datasets.

acquiring more features comes at a cost, and also results in an increase in testing time. Even though not included herein, GA exhibits similar trends to MA, yet has lower values since even the misclassification of a single variable is considered an error (see also Eq. (10)). Training time is not affected by the selection of feature cost, since training involves numerically solving Eqs. (5), (7) and (8). Unless otherwise specified, results are reported for $e_k^i = e = 0.0001$, since this value achieves a relatively good trade–off between the two accuracy metrics, the average number of features, and testing time.

To understand the effect of the number B of bins on the performance of ISEC, we compute the values of MA, GA, and average number of acquired features for $B \in \{2, 4, 5, 6, 10, 12, 20, 100\}$. We observe that the values of these performance metrics do not change significantly for different values of B. However, as expected, the training time linearly increases when we increase B (see Fig. 6). To keep the training time manageable while also achieving good accuracy results, from here onwards, we set $B = 10$ for all datasets except from Emotion, Flare, and Jure where $B = 20$.

E. The effect of Bayesian Network and Feature Importance

In this subsection, we assess the effect of the Bayesian network on the average number of features acquired for each variable X_i, as well as the effect of feature importance on the classification outcome. Fig. 7 illustrates the average number of features per variable for the Student and Insurance datasets. We observe that the average number of features acquired to classify variables G_1 and G_2 does not significantly differ. However, classifying variable G_3 requires on average significantly less number of features, which is expected since the final grade (G_3) of a student depends heavily on their intermediate grades (G_1, G_2). In contrast, we observe that for the Insurance dataset, the average number of features acquired to classify variables D and A does not significantly differ, as both of them are classified using a small number of features (≈ 3). Still, A requires on average less number of features than D as the classification decision of D affects the classification decision of A. We underscore that our analysis indicates that combining parents' classification decisions with the smart acquisition of features results in accurate classification decisions using very few number of features on average for the majority of data instances (e.g., $6, 3$, and 1 features on average for classifying G_1, G_2, and G_3, respectively). We observe similar trends for the remaining datasets.

Figs. 8 and 9 illustrate the features (and their frequency) acquired by ISEC during testing and used to classify the variables in the Insurance and Student datasets. We observe that in both cases ISEC selects intuitive features. For example,

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

the *driving skills* and *driving history* are most often used to classify the driving quality of an individual (Fig. 8(a)), while the *car damage* is more often used for assessing if an accident happened (Fig. 8(c)). We observe similar trends in the case of the Student dataset (Figs. 9(a), (c), (e)). For instance, feature *school* is most often selected to classify all variables. Intuitively, a student's school may be a good rough indicator of a student's grade, since in good schools students also tend to have higher grades. Next, we consider the effect of feature costs on the classification decisions by introducing different costs for different features based on the difficulty of collecting them in practice. For instance, considering the Insurance dataset, *age* is relatively easy to acquire, contrary to *driving skills* [34]. We assign feature costs as $e^i_k = d \times 0.0001, \forall k$, where $d \in \{1, 2, 3\}$ with the corresponding values indicating that it is easy (green), medium (yellow), and difficult (red) to acquire the corresponding feature, respectively. Our results indicate that the MA in the Student dataset goes down by 0.49% using 0.70% fewer features on average compared to constant feature costs. On the other hand, for the Insurance dataset, MA increases by 0.66% using 23.77% more features on average, which illustrates that accuracy is robust for different feature costs. We also observe that ISEC tends to acquire more low–cost features (see Figs. 9(b), (d) and (f) for example) so at to preserve accuracy. Nonetheless, if features are quite informative (e.g., *driving skills* in Fig. 8(b)), ISEC still acquires them but later in the feature acquisition process.

F. Comparison with Baselines

In this subsection, we illustrate the performance of ISEC on the datasets in Table I with respect to the metrics introduced in Section VI-C. Further, ISEC is compared with the following baseline algorithms: 1) Independent Classifier with Naive Bayes (IC–NB), SVM (IC–SVM) and ETANA [13] (IC–ETANA) as base classifiers, 2) Powerset Classifier with Naive Bayes (PC–NB) and SVM (PC–SVM) as base classifiers, 3) BCC [19], which uses a Bayesian network to determine variable relationships and chaining order, and 4) MD–KNN [21], which considers pair–wise variable dependencies and performs classification using kNN counting statistics. IC and PC with NB and SVM as base classifiers are widely used in MDC literature, while BCC and MD–KNN are recently proposed algorithms that outperform prior MDC methods. We underscore that none of these baselines dynamically selects features for classification; thus, we use ETANA [13], which performs dynamic feature selection for classification of a single variable, as a base classifier for the independent classifier approach. We set parameter k of MD–KNN to 10 as used in [21], and parameters e, B, η of ETANA to $0.0001, 10, 10$, as used in [13], respectively. We use LIBSVM with the linear kernel for MD–KNN, IC–SVM, and PC–SVM, as suggested in [21], and Naive Bayes as the base classifier for BCC [19]. All baselines' codes have been provided by their authors or are publicly available. In addition, for fair comparison, we evaluate all algorithms using the same performance metrics. Table II provides our experimental results and we discuss our main observations next.

As expected, MA is higher than GA for almost all datasets. This is because GA represents the result of predicting all variables' values together as a single variable in each instance, unlike MA, which evaluates individual variables separately (see Eqs. (9) and (10)). Therefore, a classification error of a single variable is considered a misclassified instance in terms of GA, irrespective of the correct classification of other variables, unlike MA. Furthermore, ISEC outperforms nearly all baselines with respect to MA (improvement between 5.97% and 19.68%) and GA (improvement between 2.10% and 32.00%), since it not only uses the most informative features per classification variable to infer its value, but also takes advantage of the parent–child relationships in the Bayesian network. PC based algorithms (e.g., PC–SVM) perform competitively with ISEC with respect to GA. We suspect that this is due to the fact that they translate the original problem into a single multi–class classification problem promoting the joint assignment of all the variables in the Bayesian network. However, ISEC achieves high MA and GA, while acquiring the least number of features on average (feature reduction is between 1.75% and 88.35%) compared to all the baselines. In fact, it performs better than IC–ETANA, which also performs dynamic instance–wise feature selection on a single classification variable, suggesting that exploiting parent–child relationships in the Bayesian network can benefit both accuracy and the feature acquisition process.

To validate the statistical significance of the results presented in Table II, we conduct the Friedman test that is typically used to compare the performance of classifiers over multiple datasets [37]. We include the average ranks in Table II, and we note that the p–values for GA, MA, and AF are 4.16×10^{-3}, 7.74×10^{-4}, and 4.53×10^{-11}, respectively. Thus, we conclude that there is statistical difference in the performance of ISEC and the baselines.

G. ISEC versus IBSEC

In this subsection, we compare ISEC with IBSEC using the datasets in Table I. Specifically, assuming that the true values of the classification variables of outdegree 0 are provided in the form of evidence, we run IBSEC on all datasets and assess its performance. Since all datasets except Student and Emotion have only one ancestor, we report only mean accuracy values since global and mean accuracy coincide in this case. Fig. 10 illustrates the mean accuracy of ISEC versus the mean accuracy of IBSEC, where each point $(\text{MA}_{\text{ISEC}}, \text{MA}_{\text{IBSEC}})$ is associated with a specific dataset. We include the 45° line to enable easy interpretation of the results. We observe that the majority of points lie over this line. This suggests that the inclusion of evidence significantly improves mean accuracy. Note that the average number of features acquired is not affected, since IBSEC just uses the Bayesian network structure to improve the classification decisions of the variables in the network.

VII. Conclusions and Future Work

In this work, a datum–wise inference methodology is proposed for structured data instances by balancing classification accuracy and cost of acquired features. Specifically, a

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

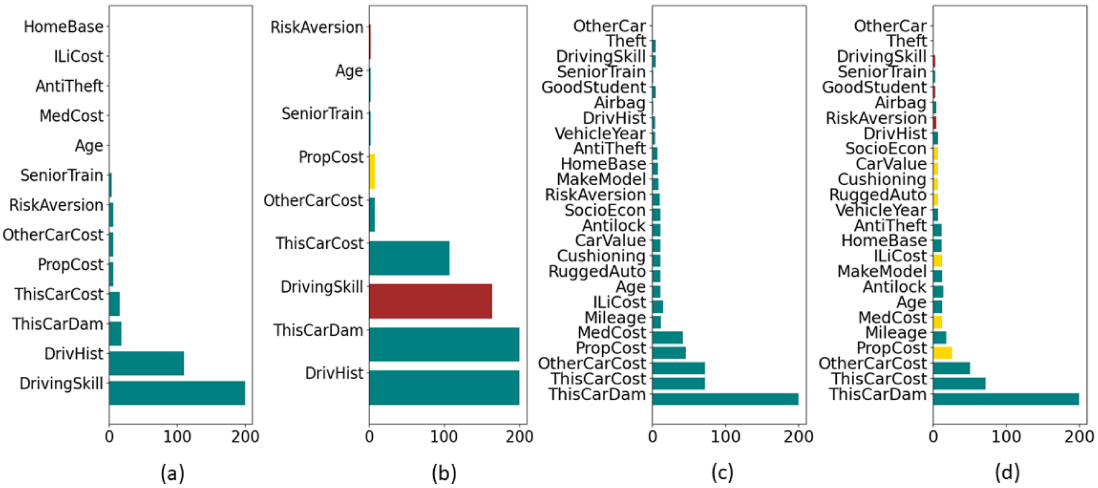


Fig. 8: Features (and their frequency) selected by ISEC during testing for the Insurance dataset ($X \triangleq \{Driving\ Quality, Accident\}$). Features are illustrated in descending order (Y–axis). (a), (b) Feature acquisition for variable *Driving Quality* under same and different feature costs, (c), (d) Feature acquisition for variable *Accident* under same and different feature costs.

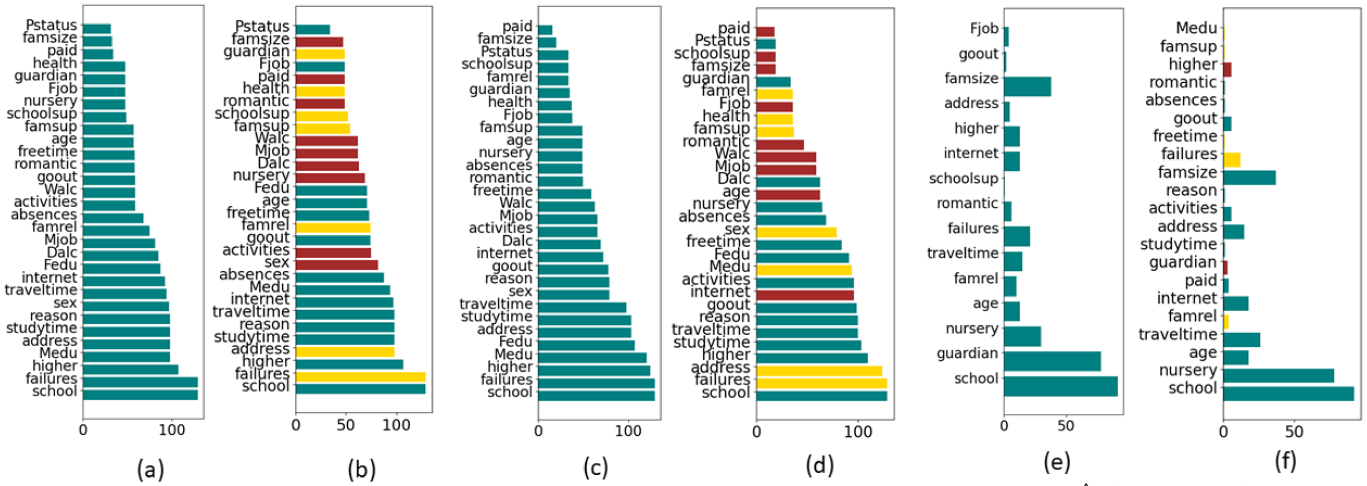


Fig. 9: Features (and their frequency) selected by ISEC during testing for the Student dataset ($X \triangleq \{G_1, G_2, G_3\}$). Features are illustrated in descending order (Y–axis). (a), (b) Feature acquisition for variable G_1 under same and different feature costs, (c), (d) Feature acquisition for variable G_2 under same and different feature costs, (e), (f) Feature acquisition for variable G_3 under same and different feature costs.

forward pass algorithm is designed that assigns a label to each classification variable in the Bayesian network by selecting the appropriate optimal number of informative features. The proposed algorithm propagates the resulting classification decisions through the Bayesian network, incorporating their effect in the inference process. Furthermore, a backward pass algorithm is designed that improves classification accuracy by incorporating evidence without requiring additional acquisition of features. The experimental analysis indicates that the proposed algorithms not only improve classification accuracy by wisely acquiring features, but also shed light into the effect of different features on classification decisions.

A limitation of the proposed methodology is its high training time, which is a direct outcome of numerically solving the relevant dynamic programming equations. Consequently, ISEC does not scale to large datasets. As part of our current efforts, we are looking into characterizing the structural properties of the associated cost functions, which along with existing methods can help to significantly decrease the training time. At the same time, we are considering the design of greedy approaches that may be suboptimal but scale well to large datasets. To improve classification accuracy even more, we are also considering extending the proposed methodology to employ more powerful classifiers (e.g., neural networks) than the proposed variable feature classification strategy.

Motivated by applications in medical diagnosis and behavioral analysis among others, we plan to consider time–varying inference of the values of the categorical variables in the Bayesian networks. In contrast to our proposed framework that focuses on supervised learning over static data assuming conditional feature independence, in these cases, semi–supervised learning over time–varying correlated features is more appropriate. To address such cases, we can consider dynamic Bayesian networks to formally describe time–series

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

TABLE II: Comparison of global accuracy (GA), mean accuracy (MA), and the average number of features (AF). The highest and the second highest accuracy values are bolded and gray–shaded, and gray–shaded, respectively. The smallest and the second smallest AF values are bolded and gray–shaded, and gray–shaded, respectively.

Dataset	Metric	ISEC	IC–NB	IC–ETANA	PC–NB	BCC	MD–KNN	IC–SVM	PC–SVM
Edm	GA	**0.5905**	0.3890	0.4668	0.5443	0.3905	0.3864	0.3578	0.4483
	MA	**0.7401**	0.6491	0.6500	0.7101	0.6952	0.6209	0.6755	0.7013
	AF	**5.8654**	16.0000	8.6333	16.0000	16.0000	16.0000	16.0000	16.0000
Voice	GA	**0.8753**	0.6897	0.8224	0.6824	0.2735	0.8359	0.7663	0.7220
	MA	**0.9364**	0.8243	0.8748	0.8343	0.5210	0.9142	0.8780	0.8514
	AF	2.5127	19.0000	**2.2719**	19.0000	19.0000	19.0000	19.0000	19.0000
Jura	GA	**0.4402**	0.3036	0.3481	0.4010	0.1588	0.2591	0.2562	0.2393
	MA	**0.6352**	0.5405	0.5845	0.6016	0.4764	0.4889	0.5307	0.4830
	AF	**7.0517**	9.0000	8.2394	9.0000	9.0000	9.0000	9.0000	9.0000
Song	GA	0.3299	0.2114	0.2509	0.2611	0.3082	**0.4229**	0.3471	0.3548
	MA	0.7134	0.6012	0.6709	0.6360	0.6802	**0.7565**	0.6728	0.6724
	AF	**16.3172**	98.0000	16.6072	98.0000	98.0000	98.0000	98.0000	98.0000
Flare	GA	0.8173	0.0277	0.7800	0.0463	**0.8204**	0.7802	0.8202	0.8202
	MA	0.9205	0.2194	0.8906	0.5736	**0.9226**	0.9035	0.9225	0.9225
	AF	**1.3040**	10.0000	7.0573	10.0000	10.0000	10.0000	10.0000	10.0000
Student	GA	**0.6099**	0.5742	0.5914	0.0815	0.5469	0.5208	0.5334	0.5021
	MA	**0.7409**	0.7227	0.5529	0.5418	0.6522	0.6546	0.6560	0.6084
	AF	**8.4940**	30.0000	14.9458	30.0000	30.0000	30.0000	30.0000	30.0000
Emotion	GA	0.3121	0.1820	0.2378	0.2731	0.0000	0.1164	0.2631	**0.3203**
	MA	0.7783	0.7391	0.7641	0.7700	0.6885	0.7026	**0.7934**	0.7718
	AF	**8.5983**	72.0000	15.3432	72.0000	72.0000	72.0000	72.0000	72.0000
Child	GA	**0.5620**	0.5509	0.5350	0.4800	0.3910	0.5098	0.3909	0.3909
	MA	**0.8197**	0.8156	0.8069	0.7783	0.7106	0.7799	0.7106	0.7106
	AF	**4.4293**	17.0000	5.8147	17.0000	17.0000	17.0000	17.0000	17.0000
Hepar2	GA	0.4200	0.0900	0.4170	0.0350	0.4180	0.4150	**0.4230**	0.4150
	MA	0.7807	0.4260	0.7757	0.4193	0.7813	0.7792	**0.7813**	0.7747
	AF	**12.6213**	67.0000	31.9470	67.0000	67.0000	67.0000	67.0000	67.0000
Sachs	GA	**0.7920**	0.7770	0.6000	0.3000	**0.7920**	0.7880	**0.7920**	**0.7920**
	MA	**0.8420**	0.8345	0.7250	0.5765	**0.8420**	0.8399	**0.8420**	**0.8420**
	AF	**1.8575**	9.0000	8.4295	9.0000	9.0000	9.0000	9.0000	9.0000
Insurance	GA	**0.8270**	0.6920	0.8100	0.6150	0.4320	0.6062	0.7240	0.7310
	MA	**0.9050**	0.8350	0.9030	0.7840	0.5870	0.7841	0.8540	0.8520
	AF	**2.9115**	25.0000	5.4565	25.0000	25.0000	25.0000	25.0000	25.0000
Avg. rank	GA	1.86	5.55	4.09	5.64	5.32	4.95	4.32	4.27
	MA	1.77	5.64	4.91	5.82	5.09	4.73	3.41	4.64
	AF	1.09	5.50	1.91	5.50	5.50	5.50	5.50	5.50

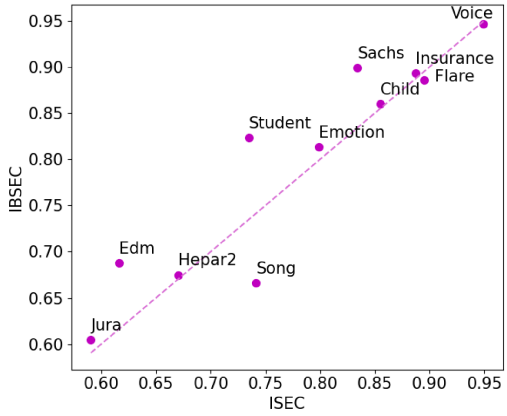


Fig. 10: Mean accuracy (MA) of ISEC and IBSEC. Dashed line represents the 45° line.

data along with spatio–temporal features extracted through neural network architectures, similar to [38]. Additionally, we can drop the conditional feature independence assumption and instead exploit the arrangement–based method used by [39] to capture pair–wise feature correlations. Finally, we plan to use an optimal neighboring assignment–based approach as in [40] to uncover relationships between training instances, thus extending our framework to semi–supervised settings.

REFERENCES

[1] S. P. Ekanayake, Y. W. Liyanage, and D.-S. Zois, "Dynamic feature selection for classification in structured environments," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*. IEEE, pp. 140–144.

[2] E. Kyrimi, S. McLachlan, K. Dube, M. R. Neves, A. Fahmi, and N. Fenton, "A comprehensive scoping review of Bayesian networks in healthcare: Past, present and future," *Artificial Intelligence in Medicine*, p. 102108, 2021.

[3] E. Nazerfard and D. J. Cook, "Using Bayesian Networks for Daily Activity Prediction," in *AAAI workshop: plan, activity, and intent recognition*. Citeseer, 2013.

[4] M. Qazi, G. M. Fung, K. J. Meissner, and E. R. Fontes, "An insurance recommendation system using Bayesian networks," in *Proceedings of the Eleventh ACM conference on recommender systems*, 2017, pp. 274–278.

[5] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[6] K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*. CRC press, 2010.

[7] E. Castillo, Z. Grande, E. Mora, X. Xu, and H. K. Lo, "Proactive, backward analysis and learning in road probabilistic Bayesian network models," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 820–835, 2017.

[8] C. Bielza and P. Larranaga, "Discrete Bayesian network classifiers: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–43, 2014.

[9] D. Heckerman, "A tutorial on learning with Bayesian networks," *Innovations in Bayesian networks*, pp. 33–82, 2008.

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2023.3271616

[10] R. Subramanian, J. Wache, M. K. Abadi, R. L. Vieriu, S. Winkler, and N. Sebe, "Ascertain: Emotion and personality recognition using commercial sensors," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 147–160, 2016.
[11] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, "A survey on online feature selection with streaming features," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 479–493, 2018.
[12] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: A survey," *Applied Computing and Informatics*, 2020.
[13] Y. W. Liyanage, D.-S. Zois, and C. Chelmis, "Dynamic instance-wise joint feature selection and classification," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 169–184, 2021.
[14] ——, "Dynamic instance-wise classification in correlated feature spaces," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 6, pp. 537–548, 2021.
[15] A. Ghosh and A. Lan, "Difa: Differentiable feature acquisition," 2023.
[16] Y. Li and J. Oliva, "Active feature acquisition with generative surrogate models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6450–6459.
[17] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.
[18] K. Trohidis, G. Tsoumakas, G. Kalliris, I. P. Vlahavas *et al.*, "Multi-label classification of music into emotions." in *ISMIR*, vol. 8, 2008, pp. 325–330. [Online]. Available: https://www.openml.org/d/41465
[19] J. C. Zaragoza, E. Sucar, E. Morales, C. Bielza, and P. Larranaga, "Bayesian chain classifiers for multidimensional classification," in *Twenty-second international joint conference on artificial intelligence*, 2011.
[20] J. Read, C. Bielza, and P. Larrañaga, "Multi-dimensional classification with super-classes," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 7, pp. 1720–1733, 2013.
[21] B.-B. Jia and M.-L. Zhang, "Md-knn: An instance-based approach for multi-dimensional classification," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 126–133.
[22] C. Bielza, G. Li, and P. Larranaga, "Multi-dimensional classification with Bayesian networks," *International Journal of Approximate Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.
[23] I. Batal, C. Hong, and M. Hauskrecht, "An efficient probabilistic framework for multi-dimensional classification," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2417–2422.
[24] S. Gil-Begue, C. Bielza, and P. Larrañaga, "Multi-dimensional Bayesian network classifiers: A survey," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 519–559, 2021.
[25] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.
[26] A. Karalič and I. Bratko, "First order regression," *Machine learning*, vol. 26, no. 2, pp. 147–176, 1997. [Online]. Available: https://www.openml.org/d/41552
[27] K. Becker, "Identifying the gender of a voice using machine learning," May 2021. [Online]. Available: http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-using-machine-learning/
[28] P. Goovaerts *et al.*, *Geostatistics for natural resources evaluation*. Oxford University Press on Demand, 1997.
[29] M.-L. Zhang, "Datasets for multi-dimensional classification." [Online]. Available: http://palm.seu.edu.cn/zhangml/Resources.htm#data
[30] "solar-flare." [Online]. Available: https://www.openml.org/d/40686
[31] D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, and R. G. Cowell, "Bayesian analysis in expert systems," *Statistical science*, pp. 219–247, 1993.
[32] A. Onisko, M. J. Druzdzel, and H. Wasyluk, "A probabilistic causal model for diagnosis of liver disorders," in *Proceedings of the Seventh International Symposium on Intelligent Information Systems (IIS98)*, 1998, p. 379.
[33] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, 2005.
[34] J. Binder, D. Koller, S. Russell, and K. Kanazawa, "Adaptive probabilistic networks with hidden variables," *Machine Learning*, vol. 29, no. 2, pp. 213–244, 1997.
[35] P. Cortez and A. M. G. Silva, "Using data mining to predict secondary school student performance," 2008. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Student+Performance
[36] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.
[37] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[38] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, "Making sense of spatio-temporal preserving representations for eeg-based human intention recognition," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3033–3044, 2019.
[39] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 5, pp. 1747–1756, 2019.
[40] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE transactions on cybernetics*, vol. 48, no. 2, pp. 648–660, 2017.

Sachini Piyoni Ekanayake received the B.Sc. degree in electrical and electronic engineering from the University of Peradeniya, Sri Lanka, in 2017. Currently, she is working toward the Ph.D. degree in electrical and computer engineering at the University at Albany, State University of New York, USA. Her research interests include machine learning and statistical signal processing.

Daphney-Stavroula Zois received the B.S. degree in computer engineering and informatics from the University of Patras, Patras, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA. Previous appointments include the University of Illinois, Urbana-Champaign, IL, USA. She is an Associate Professor in the Department of Electrical and Computer Engineering, University at Albany, State University of New York, Albany, NY, USA. She received the Viterbi Deans and Myronis Graduate Fellowships, the NSF CAREER award, and a Google AI for Social Good "Impact Scholars" award. She has served and is serving as Co-Chair, TPC member or reviewer in international conferences and journals, such as AAAI, ICASSP, ICLR, NeurIPS, IEEE Transactions on Signal Processing, IEEE Transactions on Information Theory, IEEE Transactions on Artificial Intelligence, and IEEE Transactions on Neural Networks and Learning Systems. Her research interests include decision making under uncertainty, machine learning, detection & estimation theory, intelligent systems design, and signal processing.

Authorized licensed use limited to: UNIVERSITY AT ALBANY SUNY. Downloaded on August 31,2023 at 01:37:00 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.