Dynamic Instance-Wise Classification in Correlated Feature Spaces

Yasitha Warahena Liyanage, *Student Member, IEEE*, Daphney-Stavroula Zois, *Member, IEEE*, and Charalampos Chelmis, *Member, IEEE*

***Abstract*—In a typical supervised machine learning setting, the predictions on all test instances are based on a common subset of features discovered during model training. However, using a different subset of features that is most informative for each test instance individually may improve not only the prediction accuracy but also the overall interpretability of the model. At the same time, feature selection methods for classification have been known to be the most effective when many features are irrelevant and/or uncorrelated. In fact, feature selection ignoring correlations between features can lead to poor classification performance. In this work, a Bayesian network is utilized to model feature dependencies. Using the dependence network, a new method is proposed that sequentially selects the best feature to evaluate for each test instance individually and stops the selection process to make a prediction once it determines that no further improvement can be achieved with respect to classification accuracy. The optimum number of features to acquire and the optimum classification strategy are derived for each test instance. The theoretical properties of the optimum solution are analyzed, and a new algorithm is proposed that takes advantage of these properties to implement a robust and scalable solution for high-dimensional settings. The effectiveness, generalizability, and scalability of the proposed method are illustrated on a variety of real-world datasets from diverse application domains.**

***Impact statement*— The ability to rationalize which features to use to classify each data instance is of paramount importance in a wide range of application domains, including but not limited to medicine, criminal justice, and cybersecurity. Correlations between features, and the need for variable selection at the same stage as classification, in such application domains present additional challenges to machine learning related to classification accuracy and computationally intractability. The proposed framework presents, to the best of our knowledge, the first practical solution that balances between classification accuracy and sparsity at the instance level, by dynamically choosing the most informative features, relative to each instance, from a set of potentially *correlated* features, for classifying each *individual* instance. The proposed framework achieves reductions up to 82% in the average number of features used by state-of-the-art methods without sacrificing accuracy and is robust for up to 10% of missing features. Broad positive societal implications include: fast, accurate, and cost-efficient inference in complex dynamic settings; and ease of interpretation of and trust in machine learning outcomes by domain experts (e.g., doctors and lawyers).**

***Index Terms*—Bayesian network, correlated features, costly features, datumwise feature selection and classification, sequential feature selection.**

I. Introduction

A WIDE range of applications, including but not limited to medicine and robotics, demand practical solutions that can perform feature selection and classification jointly in a dynamic setting for each data instance individually. For example, consider a scenario, where a doctor is called to provide a medical diagnosis to a patient. The doctor's diagnosis may often be time critical (e.g., in emergencies) and/or depend on numerous costly medical tests (out of which features are to be extracted), some of which cost thousands of dollars [1]. At the same time, a *different* set of tests may be appropriate for *each* individual patient (i.e., data instance). For instance, Kao *et al.* [2] have shown that relevant features for predicting heart failure may differ across patient subgroups. Finally, considering dependencies between medical tests (and corresponding features) is equally important [3], since individual features may seem irrelevant with the class when examined independently, but when combined may improve classification accuracy and at the same time enhance interpretability of the final decision. Similarly, in the domain of robotics, an autonomous vehicle can control the view of its environment (e.g., change position and modify sensors parameters) to inspect and classify an object of interest [4]. In this context, it may be important to select which sensors to use or what kind of measurements to take, while at the same time ensuring that an object in the field of view can be accurately classified.

In this article, the problem of selecting *which features* to use to *classify each test instance* as features *sequentially* arrive one at a time is explored. The current work extends the prior work [5], which formalized this problem with the simplifying assumption that features are conditionally independent given the class variable. Specifically, building upon [5], herein, features ensure consistency: correlation versus dependence is efficiently and effectively modeled using a Bayesian network. In general, handling feature dependencies can lead to computationally intractable solutions [6]. Therefore, a novel feature ordering is presented, such that each selected feature contains the maximum possible new information about the class variable with respect to the already evaluated feature set. The optimum solution is

Manuscript received February 5, 2021; revised May 20, 2021 and June 28, 2021; accepted July 30, 2021. Date of publication September 8, 2021; date of current version December 13, 2021. This work was supported by the National Science Foundation under Grant ECCS-1737443 and Grant CNS-1942330. This paper was recommended for publication by Associate Editor G. Nelson DeSouza upon evaluation of the reviewers' comments. *(Corresponding author: Daphney-Stavroula Zois.)*

Yasitha Warahena Liyanage and Daphney-Stavroula Zois are with the Department of Electrical and Computer Engineering, University at Albany, Albany, NY 12222 USA (e-mail: yliyanage@albany.edu; dzois@albany.edu).

Charalampos Chelmis is with the Department of Computer Science, University at Albany, Albany, NY 12222 USA (e-mail: cchelmis@albany.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAI.2021.3109858.

Digital Object Identifier 10.1109/TAI.2021.3109858

also derived for this more general formulation, and important structural properties are analyzed that facilitate the design of a scalable method. The proposed method is evaluated and compared with prior work on a variety of real-world datasets.

Next, the unique contributions of this work are summarized as follows: 1) the optimum stopping feature (i.e., the feature at which the sequential evaluation process terminates) and the optimum classification strategy are mathematically derived for each data instance individually without imposing any assumptions on feature dependencies; 2) the structure of the optimum solution is theoretically analyzed; 3) an efficient implementation of the optimum solution is introduced that considers feature correlations to guide the feature selection process; and 4) the effectiveness, generalizability, and scalability of the proposed method are evaluated using 11 publicly available datasets. To facilitate reproducibility, the source code of the proposed method is available at:[1]

II. Related Work

In this section, the most relevant prior work on feature selection and classification is summarized.

To accommodate large or unknown feature spaces during *model training*, streaming feature selection methods [7] are designed to handle features arriving sequentially over time. Existing work on this area can be roughly categorized into two directions depending on the availability of prior information about the feature space [8]–[11] or not [12], [13]. In general, various threshold-based approaches have been proposed, where a newly arriving feature is selected if a constraint is satisfied (e.g., predefined threshold [8], dynamically varying threshold [9], conditional independence via G^2-test [10], or Fishers Z-test [11]). In [12], features are selected if they exhibit high correlation with the class variable and low correlation with already selected features, while the boundary region of the decision is kept as small as possible. Zhou *et al.* [13] extends [12] by considering the size of the neighborhood of each data instance. Motivated by real-world applications, where training instances arrive sequentially or access to the full training dataset is not feasible, Wu *et al.* [14] jointly train a linear model and acquire a sparse representation of the feature space *global* to the entire dataset during the training process. All the above works use the *same fixed* set of features to classify *all instances* during *testing*. In sharp contrast to the above lines of work, in the proposed setting, both the training instances and the full feature space are available during the training process, and the goal is to *dynamically* select and use for classification a *different* subset of features for *each instance* during the *testing process*. As a result, each testing instance is classified using *different variable features*. This instance-wise property is demonstrated in Section VI-G.

Recent studies have shown that relevant features may differ across data instances, for example, in predicting heart failure for different patients [2]. At the same time, as complex machine learning models become more prevalent, the need to interpret their results becomes critical. Hence, instance-wise feature selection [15]–[17] tries to identify a small number of relevant features that explain/predict the output of a machine learning model during testing. In [15], a single neural network is learned to identify the top k features that explain a pretrained model based on mutual information. Yoon *et al.* [16] adopt an actor–critic architecture with three neural networks, bypassing the need for backpropagation to outperform [15]. To avoid the high computational cost of the above methods, Xiao and Wang [17] limit the number of possible relevant feature subsets to K, and model this constraint with a mixture of K deep neural networks. The sensitivity magnitude of the model is then used to select the most relevant features. These methods work in a *static* setting, since all feature values of a test instance must be first revealed. Such methods do not scale for large feature spaces, since the search space grows exponentially with the number of features. The method proposed herein can be used for model interpretability and as such is related to these methods. However, unlike that line of work, the proposed method is *dynamic*, in the sense that features arrive sequentially one at a time during testing, and the goal is to *jointly* select features and classify each data instance in this regime. Additionally, the number of features used for each instance is neither fixed nor predefined; it is *optimally* derived by the proposed framework. Finally, the proposed method scales well, with large feature spaces being able to accommodate more than 1 million features.

Similar to this work, classification with costly features [18], [19] considers costs associated with feature evaluation and misclassification, and the goal is to limit the number of features used for classification per data instance during testing. Such methods, however, define the problem *globally* with respect to the training dataset, namely, by introducing a penalization term to limit the number of features used for classification in a standard empirical loss minimization problem. In that sense, even though such approaches end up sequentially evaluating different features per instance before they classify it, the resulting classification function is *globally* learned with respect to the *dataset*. This modified problem is shown to be equivalent to a deterministic Markov decision process (MDP) formulation and solved by a linear [18] or nonlinear approximation [19] of the associated Q-function. The size of the state space of the MDP grows exponentially with the dimension of the feature space, making these methods impractical for high-dimensional settings. The proposed approach in this article is conceptually different from these methods in that the problem of joint feature selection and classification is defined and solved *individually* with respect to *each data instance*. In addition, the optimum classification strategy and the optimum number of features to be used for classification for each data instance individually are mathematically derived in the generic case of correlated features. Last but not least, key properties of the optimum solution are uncovered, thus enabling the proposed method to scale to large feature spaces.

III. Proposed Framework

In this section, the task of instanc-ewise supervised multiclass classification in correlated feature spaces is posed as a sequential decision process for which the optimum solution is derived. Specifically, the goal is to learn to sequentially choose a subset of features, relative to each test instance, using which each particular instance is to be assigned to one of L classes. Table I summarizes the notation used hereafter.

[1] https://github.com/IDIASLab/IFC2F

TABLE I
EXPLANATION OF MAIN SYMBOLS USED IN THIS ARTICLE

Symbol	Description
S	collection of data instances
s	data instance $s \in S$
F	Feature set
K	# of features
F_k	kth feature $F_k \in F$ $(1 \leqslant k \leqslant K)$
f_k	value of kth feature F_k $(1 \leqslant k \leqslant K)$
L	# of classes
$\mathcal{C}$	class variable
c_i	class assignment to class $\mathcal{C}$
p_i	prior probability of class i
e_k	Cost coefficient of kth feature F_k
Q_{ij}	Missclassification cost for classes c_j and c_j

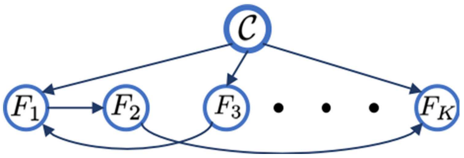


Fig. 1. Sample Bayesian network of features in F and the class variable in $\mathcal{C}$.

Dependencies between features and the class variable (see Fig. 1) are modeled using a Bayesian network $\mathcal{B} \triangleq \{G, \Theta\}$ [20], which can be learned using methods such as in [21] and [22]. Specifically, a directed acyclic graph $G = (V, E)$, where $V = F \cup \{\mathcal{C}\}$ is the set of features in F augmented with variable $\mathcal{C}$ and E is the set of edges denoting correlations[2] among the nodes in V, is given, and the set Θ of conditional probability distributions (CPDs) for G is learned during training.

The goal is to leverage feature dependencies to train an instance-wise classifier that allows the number of features used for classification to vary relative to each instance, so as to optimize the tradeoff between accuracy and sparsity at the individual instance level. Specifically, in order to select one out of L possible classes for each instance s, the proposed approach evaluates features sequentially by choosing the features that are 1) highly correlated with the class variable and 2) conditionally independent with the already evaluated feature set. At each step, the proposed approach considers the cost of examining the remaining features to decide between continuing the process or if enough information is available for a classification decision to be reached. Herein, two random variables R and D_R are defined.

Definition 1: Random variable $R \in \{0, \dots, K\}$ denotes the number of feature evaluations before the framework terminates. The event $\{R = k\}$ represents that the framework stops after evaluating k features.

Definition 2: Random variable $D_R \in \{1, \dots, L\}$ denotes the assigned value for class variable $\mathcal{C}$ based on the information accumulated up to feature F_R. The event $\{D_{\{R=k\}} = i\}$ represents assignment of class c_i using features $\{f_1, f_2, \dots, f_k\}$.

The optimum value R^* and the best class assignment $D^*_{R^*}$ for each data instance $s \in \mathcal{S}$ are obtained by minimizing

$$J(R, D_R) = \mathbb{E}\left\{\sum_{k=1}^{R} e_k\right\} + \sum_{j=1}^{L}\sum_{i=1}^{L} Q_{ij} P(D_R = j, \mathcal{C} = c_i) \tag{1}$$

where $e_k > 0$ is the feature evaluation cost representing the time and effort required to evaluate feature F_k, $Q_{ij} \geqslant 0$ is the missclassification cost of assigning class c_j when class c_i is true, and $P(D_R = j, \mathcal{C} = c_i)$ is the joint probability of assigning class c_j when class c_i is true. Specifically, $\mathbb{E}\{\sum_{k=1}^{R} e_k\}$ is the expected cost accrued due to feature evaluations, and the double summation corresponds to expected cost associated with the classification rule D_R. Thus, the optimization problem is equivalent to finding $R^*, D^*_{R^*}$, such that

$$\underset{R, D_R}{\text{minimize}}\ J(R, D_R). \tag{2}$$

To solve (2), first, a subset of highly correlated features with the class variable $\mathcal{C}$ is identified. These features are sufficient for accurately inferring $\mathcal{C}$'s value. This is achieved by finding the Markov blanket [20] $M_{\mathcal{C}}$ of $\mathcal{C}$ in G. $\mathcal{B}_{\mathcal{C}}$ denotes the induced subgraph of G with nodes $V_{\mathcal{B}_{\mathcal{C}}} = M_{\mathcal{C}} \cup \{\mathcal{C}\}$. Features in $M_{\mathcal{C}}$ are then sequentially evaluated so that at each step, the feature in the subset of currently unselected features, that provides the maximum additional information about $\mathcal{C}$ with respect to the already evaluated feature set, is selected. This is achieved using

$$\pi_k = \frac{\operatorname{diag}\left(\Delta_k(F_k|F_1, \dots, F_{k-1}, \mathcal{C})\right)\pi_{k-1}}{\Delta_k^T(F_k|F_1, \dots, F_{k-1}, \mathcal{C})\pi_{k-1}} \tag{3}$$

where $\Delta_k(F_k|F_1, \dots, F_{k-1}, \mathcal{C}) = [P(F_k|F_1, \dots, F_{k-1}, c_1), \dots, P(F_k|F_1, \dots, F_{k-1}, c_L)]^T$ can be computed using exact inference algorithms (see, e.g., [22]), $\operatorname{diag}(A)$ denotes a diagonal matrix with diagonal elements being the elements in vector A, $\pi_0 = [p_1, p_2, \dots, p_L]^T$, and

$$\pi_k \triangleq [\pi_k^1, \pi_k^2, \dots, \pi_k^L]^T \tag{4}$$

is the *a posteriori probability* vector with $\pi_k^i = P(c_i|F_1, \dots, F_k)$. In lieu of $P(\mathcal{C} = c_i|F_1 = f_1, \dots, F_k = f_k)$, $P(c_i|F_1, \dots, F_k)$ is used hereafter to improve readability.

For illustration purposes, suppose that F_1 is the feature most correlated with $\mathcal{C}$. Let $\widetilde{B}_1 \subseteq \{M_{\mathcal{C}} \cup \mathcal{C}\}$ denote the subset of F_1's Markov blanket in $\mathcal{B}_{\mathcal{C}}$. F_1 is conditionally independent to all $F_k \in \{M_{\mathcal{C}} - \widetilde{B}_1\}$ given $\widetilde{B}_1$. Therefore, the next feature to be evaluated should come from $\{M_{\mathcal{C}} - \widetilde{B}_1\}$. It follows that the kth feature to be evaluated must belong to $\{M_{\mathcal{C}} - \bigcup_{i=1}^{k-1} \widetilde{B}_i\}$.

Next, $P(D_R = j, \mathcal{C} = c_i)$ can be simplified as $P(D_R = j, c_i) = \mathbb{E}\{\pi_R^i \mathbb{1}_{\{D_R=j\}}\}$ by exploiting the fact that $x_R = \sum_{k=0}^{K} x_k \mathbb{1}_{\{R=k\}}$ for any sequence of random variables $\{x_k\}$, where $\mathbb{1}_A = 1$ when A occurs, and $\mathbb{1}_A = 0$ otherwise. Thus

$$J(R, D_R) = \mathbb{E}\left\{\sum_{k=1}^{R} e_k + \sum_{j=1}^{L} Q_j^T \pi_R \mathbb{1}_{\{D_R=j\}}\right\} \tag{5}$$

where $Q_j \triangleq [Q_{1,j}, Q_{2,j}, \dots, Q_{L,j}]^T$.

At this point, the optimum classification strategy D_R^* can be obtained for any given R by noting that $\sum_{j=1}^{L} Q_j^T \pi_R \mathbb{1}_{\{D_R=j\}} \geqslant g(\pi_R)$, where $g(\pi_R) \triangleq \min_{1 \leqslant j \leqslant L}[Q_j^T \pi_R]$. Therefore, the optimum classification strategy D_R^* for any given R is

$$D_R^* = \arg\min_{1 \leqslant j \leqslant L} \left[Q_j^T \pi_R\right] \tag{6}$$

which assigns the given data instance to the class yielding the minimum misclassification cost. This suggests that $J(R, D_R) \geqslant \min_{D_R} J(R, D_R)$, which, in turn, implies that the cost function

[2] Correlation is measured using mutual information, which quantifies the "distance" from independence between a pair of random variables [23].

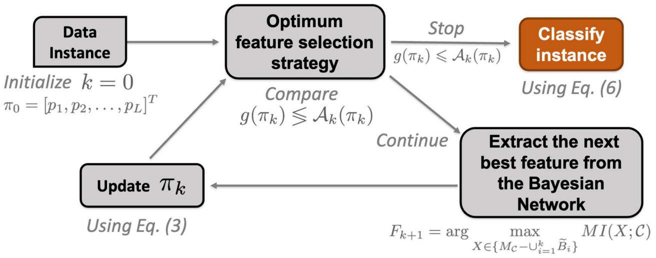

Data Instance
Initialize $k = 0$
$\pi_0 = [p_1, p_2, \ldots, p_L]^T$
Optimum feature selection strategy
Compare
$g(\pi_k) \lesseqgtr \mathcal{A}_k(\pi_k)$
Stop
$g(\pi_k) \leqslant \mathcal{A}_k(\pi_k)$
Classify instance
Using Eq. (6)
Continue
Extract the next best feature from the Bayesian Network
$F_{k+1} = \arg \max_{X \in \{M_{\mathcal{C}} - \cup_{i=1}^{k} \widetilde{B}_i\}} MI(X;\mathcal{C})$
Update π_k
Using Eq. (3)

Fig. 2. Graphical illustration of the proposed framework. $MI(X;\mathcal{C})$ denotes mutual information between variables X and $\mathcal{C}$.

in (5) can be reduced to

$$\widetilde{J}(R) = \mathbb{E}\left\{\sum_{k=1}^{R} e_k + g(\pi_R)\right\}. \tag{7}$$

Note that (7) can be minimized with respect to $R \in \{0, \ldots, K\}$ in at most $K+1$ stages using *dynamic programming* [24], as shown in Theorem 1.

Theorem 1: For $k = K-1, \ldots, 0$, the function $\bar{J}_k(\pi_k)$ is related to $\bar{J}_{k+1}(\pi_{k+1})$ as follows:

$$\bar{J}_k(\pi_k) = \min\left[g(\pi_k), \mathcal{A}_k(\pi_k)\right] \tag{8}$$

where $\mathcal{A}_k(\pi_k) \triangleq e_{k+1} + \sum_{F_{k+1}} \Delta_{k+1}^T(F_{k+1}|F_1, \ldots, F_k, \mathcal{C})\pi_k$ $\bar{J}_{k+1}\left(\frac{\operatorname{diag}(\Delta_{k+1}(F_{k+1}|F_1,\ldots,F_k,\mathcal{C}))\pi_k}{\Delta_{k+1}^T(F_{k+1}|F_1,\ldots,F_k,\mathcal{C})\pi_k}\right)$, and $\bar{J}_K(\pi_K) = g(\pi_K)$. The optimum feature selection strategy is $\{F_1, F_2, \ldots, F_{R^*}\}$, where R^* is equal to the first $k < K$ for which $g(\pi_k) \leqslant \mathcal{A}_k(\pi_k)$, or $R^* = K$ if there are no more features to be evaluated.

Fig. 2 summarizes the process of classifying a data instance during testing. Initially, $k = 0$ and $\pi_0 = [p_1, \ldots, p_L]$. At each stage k, the proposed framework compares the cost $g(\pi_k)$ of stopping to the expected cost $\mathcal{A}_k(\pi_k)$ of continuing. If $g(\pi_k) \leqslant \mathcal{A}_k(\pi_k)$, the framework stops evaluating features and classifies the instance using (6). Otherwise, it evaluates the next best feature $F_{k+1} = \arg\max_{X \in \{M_{\mathcal{C}} - \cup_{i=1}^{k} \widetilde{B}_i\}} MI(X;\mathcal{C})$ from the Bayesian network and updates π_k using (3). Details about the computation of mutual information, $MI(X;\mathcal{C})$, are provided in Section V-A. These steps are repeated until the data instance is classified using a subset of or all K features.

IV. Theoretical Results

In this section, important properties of the optimum classification strategy D_R^* in (6) and the optimum feature selection strategy in (8) are analytically derived.

Consider a general form of the function $g(\pi_R)$ used to derive the optimum classification strategy in (6) given by $g(\varpi) \triangleq \min_{1 \leqslant j \leqslant L}[Q_j^T \varpi]$, $\varpi \in [0,1]^L$, where $\varpi = [\omega_1, \ldots, \omega_L]^T$, such that $\omega_i \geqslant 0, \sum_{i=1}^{L} \omega_i = 1$. Here, the domain of $g(\varpi)$ is the probability space of ϖ, which is an $(L-1)$-dimensional unit simplex. Function $g(\varpi)$ has some interesting properties, as described in Lemma 1.

Lemma 1: The function $g(\varpi)$ is concave, continuous, and piecewise linear. In particular, $g(\varpi)$ consists of at most L hyperplanes, represented by the set $\{Q_j^T\}_{j=1}^{L}$ of L vectors.

Next, consider the general form of the function $\mathcal{A}_k(\pi_K)$ in (8) given by $\mathcal{A}_k(\varpi) = e_{k+1} + \sum_{F_{k+1}} \Delta^T$ $(F_{k+1}|F_1, \ldots, F_K, \mathcal{C})\varpi \bar{J}_{k+1}\left(\frac{\operatorname{diag}(\Delta(F_{k+1}|F_1,\ldots,F_K,\mathcal{C}))\varpi}{\Delta^T(F_{k+1}|F_1,\ldots,F_K,\mathcal{C})\varpi}\right)\Bigg]$. Lemma 2 summarizes the key properties of this function.

Lemma 2: The functions $\mathcal{A}_k(\varpi), k = 0, \ldots, K-1$, are concave, continuous, and piecewise linear.

The properties of functions $g(\varpi)$ and $\mathcal{A}_k(\varpi)$ stated in Lemmas 1 and 2, respectively, allow for a parsimonious representation of the function related to the optimum feature selection strategy in (8), as stated in Theorem 2.

Theorem 2: At every stage $k \in \{0, \ldots, K\}$, there exists a set $\{\alpha_k^i\}, \alpha_k^i \in \mathbb{R}^{1 \times L}$, of vectors such that $\bar{J}_k(\varpi) = \min_i[\alpha_k^i \varpi]$ with $\{\alpha_K^i\} \triangleq \{Q_j^T\}_{j=0}^{L}$.

The above properties can be used to derive an efficient algorithmic implementation of the optimum solution.

V. Proposed Algorithm

Theorem 2, Lemma 1 and the fact that $\bar{J}_k(\varpi) = \min[g(\varpi), \mathcal{A}_k(\varpi)], k \in \{0, \ldots, K-1\}$, allow for an efficient implementation of the optimum feature selection strategy in Theorem 1. Specifically, the decision to stop or continue the feature evaluation process depends only on the vector $\alpha_k^{\mathcal{I}} = \arg\min_{\alpha_k^i}[\alpha_k^i \varpi]$, such that if $\alpha_k^{\mathcal{I}} \in \{Q_j^T\}_{j=0}^{L}$, the feature evaluation process stops; otherwise, the next feature is to be evaluated. This is based on the fact that if $\arg\min_{\alpha_k^i}[\alpha_k^i \varpi] \in \{Q_j^T\}_{j=0}^{L}$, it implies that $g(\varpi) \leqslant \mathcal{A}_k(\varpi)$ due to the following two reasons: 1) the set $\{Q_j^T\}_{j=0}^{L}$ of L vectors represents the L hyperplanes of $g(\varpi)$ (see Lemma 1); and 2) $\bar{J}_k(\varpi) = \min_i[\alpha_k^i \varpi] = \min[g(\varpi), \mathcal{A}_k(\varpi)]$. Based on this fact, a dynamic Instance-wise joint Feature selection and Classification algorithm for Correlated Features (IFC^{2}F) is presented. Initially, ϖ is set to π_0, and $\alpha_0^{\mathcal{I}} = \arg\min_{\alpha_0^i}[\alpha_0^i \varpi]$ is computed. If $\alpha_0^{\mathcal{I}} \in \{Q_j^T\}_{j=0}^{L}$, IFC^{2}F classifies the instance under examination to the appropriate class, based on the optimum classification strategy in (6). Otherwise, the first feature is evaluated. IFC^{2}F repeats these steps until either it decides to classify the instance using $< K$ features or using all K features. Algorithm 1 describes these steps in detail. The input vector sets $\{\alpha_k^i\}$ can be computed using a standard point-based value iteration algorithm [25] during training. For simplicity, the Perseus algorithm [26] is used, among the several point-based value iteration algorithms in the literature [27]. Specifically, a fixed number β (e.g., ∼100 [26]) of reachable ϖ vectors from each stage, marginal probability tables $\Delta(F_k|F_1, \ldots, F_{k-1}, \mathcal{C})$, misclassification costs Q_{ij} and feature evaluation costs e_k were provided to the Perseus algorithm to obtain $\{\alpha_k^i\}, k \in \{0, \ldots, K-1\}$.

A. Practical Considerations

The adjusted mutual information (AMI)[3] between each feature and the class variable is computed, and $M_{\mathcal{C}}$ is acquired by removing low correlation features based on a threshold η on AMI. Specifically, η is initialized to 1 and is iteratively halved until the number of filtered features is greater than zero. For

[3] The maximum of this value represents perfect correlation between the variables, while a value around zero represents independence [28].

Algorithm 1: IFC^{2}F.

```
Input: Vector sets $\{\alpha_0^i\}, \ldots, \{\alpha_{K-1}^i\}$, and
  misclassification costs $Q_{ij}, i, j \in \{1, \ldots, L\}$
Output: Classification decision $D$ of the instance under
  examination, number $R$ of features used
Initialize $\varpi = \pi_0$
for $k = 0$ to $K$ do
  if $k = K$ or $\arg\min_{\alpha_k^i} [\alpha_k^i \varpi] \in \{Q_j^T\}_{j=0}^L$ then
    Break
  else
    Obtain next feature value $f_{k+1}$
    Update $\varpi$ using (3)
  end if
end for
Return: $D = \arg\min_{1 \leqslant j \leqslant L} [Q_j^T \varpi]$, $R = k$
```

simplicity, the Bayesian network is assumed to exhibit a tree structure rooted at the class variable (experiments are performed on three alternative network structures in Section VI-C). Such networks can be efficiently constructed (e.g., by building the maximum spanning tree [29]) using pairwise conditional mutual information. Conditional probability tables (i.e., $P(F_k|\Pi_{F_k})$, where Π_{F_k} denotes the set of parents of F_k), are estimated using a smoothed maximum likelihood estimator. Specifically, $\hat{P}(F_a = f_a | F_b = f_b, \mathcal{C} = c_i) = \frac{N_{a,b,i}+1}{N_{b,i}+V}$, where $N_{a,b,i}$ denotes the number of samples that satisfy $F_a = f_a$ and $F_b = f_b$, and belong to class c_i, $N_{b,i}$ denotes the number of samples that satisfy $F_b = f_b$, and belong to class c_i, and V is the number of quantization levels considered. The *a priori* probabilities are estimated as $P(c_i) = \frac{N_i}{\sum_{i=1}^L N_i}, i = 1, \ldots, L$, where N_i is the number of instances that belong to class c_i. To reduce both memory requirements and the number of computations when storing and computing marginal probability tables $\Delta(F_k|F_1, \ldots, F_{k-1}, \mathcal{C})$, the dependence of each variable is limited to be second order, such that the only dependence for F_k other than $\mathcal{C}$ is its first ancestor in the set $\{F_1, \ldots, F_{k-1}\}$, except for F_1, which has $\mathcal{C}$ as its only dependent.

B. Complexity Analysis

1) Preprocessing Stage: This stage consists of three steps. First, extracting the highly correlated feature set based on a threshold on mutual information is $\mathcal{O}(K \log K)$. Second, learning a tree-based Bayesian network with corresponding CPDs is $\mathcal{O}(K^2 + KL)$ [29]. Third, computing marginal probability tables from CPDs is $\mathcal{O}(KL)$. Thus, the complexity of the preprocessing stage is $\mathcal{O}(K^2 + KL)$.

2) Training Stage: In this stage, the optimum $\{\alpha_k^i\}$ vectors are determined. Computing the optimum α_k^i vectors for all K stages by considering a fixed set of belief points from each stage using Perseus algorithm is $\mathcal{O}(KLV)$ [26]. Thus, model training is $\mathcal{O}(KLV)$.

3) Testing Stage: The computational complexity of computing the minimum $\alpha_k^i \varpi$ among the set $\{\alpha_k^i\}$ is $\mathcal{O}(L)$, since: 1) the set $\{\alpha_k^i\}$ computed using the Perseus algorithm contains at most a constant number of vectors [26] and 2) the dot product between a pair of $[0, 1]^L$ vectors requires $2L - 1$ computations. The complexity of obtaining a new feature is $\mathcal{O}(1)$, while updating ϖ using (3) is $\mathcal{O}(L)$, since a dot product between a pair of L-dimensional vectors must be computed. Hence, IFC^{2}F can classify an instance in $\mathcal{O}(KL)$.

TABLE II
DATASETS USED IN THE EXPERIMENTS

Dataset	# Instances	# Features	# Classes
Madelon	$2,000$	500	2
Lung Cancer	181	$12,533$	2
MLL	72	$5,848$	3
Dexter	300	$20,000$	2
Car	174	$9,182$	11
Lung2	203	$3,312$	5
Leukemia	72	$7,129$	2
Prostate	102	$6,033$	2
Spambase	4601	57	2
Dorothea	800	$100,000$	2
News20	$19,996$	$1,355,191$	2

VI. Experimental Evaluation

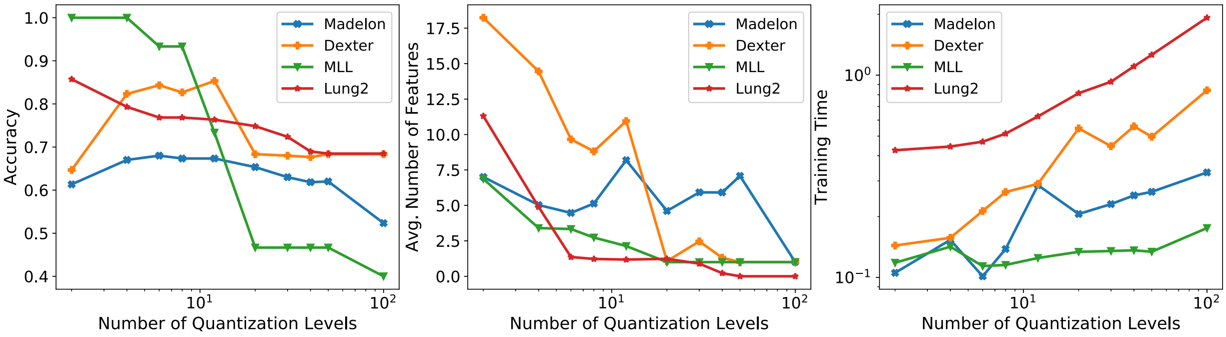
In this section, an extensive set of experiments is conducted to evaluate the performance of IFC^{2}F using 11 benchmark datasets: six DNA microarray datasets (Lung Cancer, Lung2, MLL, Car, Leukemia, and Prostate) [30], four NIPS feature selection challenge datasets (Dexter, Madelon, Dorothea, and Spambase) [31], and one high-dimensional dataset (News20) [32]. Table II summarizes these datasets. For Madelon, MLL, Dexter, and Dorothea, the originally provided training and validation sets are used, while for the remaining datasets, fivefold cross-validated results are reported. All experiments are conducted on an iMac with Quad-Core Intel Core i7 @3.30-GHz CPU, 16-GB memory, and macOS Catalina.

A. Effect of Feature Space Quantization

In Section V-A, the feature space was quantized to estimate the conditional probability tables. Herein, the effect of the number V of quantization levels on IFC^{2}F is analyzed using four datasets (Lung2, Dexter, Madelon, and MLL) (see Fig. 3). It is observed that increasing the number of bins results in a significant drop in accuracy for the MLL dataset, while for the Dexter dataset, increasing the number of bins from 2 to 20 results in an improvement in accuracy. All the datasets show a reduction in the number of features used as V is increased. These observations suggest that increasing the resolution of the feature space to a very high value can cause overfitting. However, at the same time, it can help to accommodate data sparsity in sparse datasets (e.g., Dexter). On the other hand, the linear relationship between training time and the number of bins V in Fig. 3(c) validates the $\mathcal{O}(KLV)$ complexity of IFC^{2}F's training stage (see Section V-B). In the rest of the experiments, to eliminate overfitting, V is set to a moderate value (i.e., 4), except for sparse datasets (i.e., Dexter, Dorothea, Spambase, and News20), where V is set to a slightly higher value (i.e., 10).

B. Effect of Feature Evaluation Cost

To study the behavior of IFC^{2}F for varying values of feature evaluation cost $e = \{0.1, 0.01, 0.001, 0.0001\}$, when all features incur the same cost (i.e., $e_k = e, \forall k$), the accuracy



(a) (b) (c)

Fig. 3. Variation of (a) accuracy, (b) average number of features, and (c) training time (in seconds) as a function of the number $V \in \{2, 4, 6, 8, 12, 20, 30, 40, 50, 100\}$ of bins using Lung2, Dexter, Madelon, and MLL datasets.

TABLE III

Accuracy and Average Number of Features Used by IFC^{2}F for Different Feature Evaluation Cost Values Using Lung2, Dexter, Madelon, and MLL Datasets

Dataset	e = 0.1		e = 0.01		e = 0.001		e = 0.0001	
	Accuracy	Avg. # Features	Accuracy	Avg. # Features	Accuracy	Avg. # Features	Accuracy	Avg. # Features
Madelon	0.6217	1.00	0.6700	5.03	0.6867	21.54	0.6767	42.34
Dexter	0.6467	1.00	0.8533	10.95	0.8233	25.52	0.8133	31.02
MLL	0.9333	2.20	1.00	3.40	1.00	3.73	1.00	3.73
Lung2	0.6846	0.00	0.8573	11.30	0.8918	18.01	0.8818	22.30

TABLE IV

Accuracy and Average Number of Features Used by IFC^{2}F for Different Dependence Structures Using Lung2, Dexter, Madelon, and MLL Datasets

Dataset	Tree		Line		Random		Clique	
	Accuracy	Avg. # Features	Accuracy	Avg. # Features	Accuracy	Avg. # Features	Accuracy	Avg. # Features
Madelon	0.6700	5.03	0.6150	4.73	0.6200	1.54	0.6217	1.00
Dexter	0.8533	10.95	0.7833	6.20	0.7833	12.02	0.6467	1.00
MLL	1.00	3.40	1.00	4.80	0.9333	2.73	0.4667	1.00
Lung2	0.8573	11.30	0.8672	12.57	0.9116	15.47	0.7733	1.00

and the average number of features used for classification for constant misclassification costs (i.e., $Q_{ij} = 1 \forall i \neq j, Q_{ii} = 0, i, j \in \{1, \ldots, L\}$) are measured (see Table III). Different e values result in different number of features used and levels of accuracy. Intuitively, using a small portion of the total feature set leads to low accuracy, whereas when the average number of features used increases, the performance improves dramatically. From here onwards, unless specified, results are reported for $e = 0.01$, since according to Table III, IFC^{2}F achieves the best tradeoff between accuracy and the average number of features used for this value.

C. Effect of Bayesian Network Structure

In this subsection, the behavior of IFC^{2}F is analyzed for different dependence structures by considering three alternatives in addition to the tree-based structure ("Tree") introduced in Section V-A. Initially, features are reverse sorted (highest value first) with respect to mutual information with the class variable, and the following dependence structures are considered: 1) "Line": a directed line graph having edges pointing outward starting from the first feature in the ordering; 2) "Random": a random directed acyclic graph; and 3) "Clique": a complete directed graph. Note that for all of these dependence structures, the class variable is considered as a common parent connected to all feature nodes. Table IV provides a comparison among these four dependence structures. The "Tree" structure achieves the best tradeoff between accuracy and the number of features used and additionally seems to better approximate the true structure in three out of four datasets. However, it is possible that alternative and/or simpler structures may be more appropriate in certain scenarios, as indicated by the case of Lung2, where the random structure achieves higher overall accuracy (at the expense of average number of features used). Since the true dependence structure between features is unknown, a definitive conclusion about the optimality of the "Tree" structure cannot be reached. This forms an interesting direction to be explored in future work. Henceforth, the "Tree" structure is considered in the rest of the experimental analysis.

D. Comparison With Baselines

In this subsection, IFC^{2}F is compared with 1) two dynamic feature selection methods: ETANA [5], F-ETANA [5] and 2) six streaming feature selection methods: OFS-Density [12], OFS-A3M [13], SAOLA [11], OSFS [10], Fast-OSFS [10], and Alpha-Investing [9]. In streaming feature selection methods [9]–[13], a feature is selected if it satisfies an appropriately defined criterion (e.g., belongs in the approximated Markov blanket of the class variable [10], [11]; p-statistic is greater than

TABLE V
COMPUTATIONAL COMPLEXITY OF BASELINES

Method	Computational Complexity
OFS–Density [13]	$\mathcal{O}(K^2N^2\log N)$
OFS–A3M [14]	$\mathcal{O}(K^2N^2\log N)$
SAOLA [12]	$\mathcal{O}(K^2)$
OSFS [11]	$\mathcal{O}(K^2KK^K)$
Fast–OSFS [11]	$\mathcal{O}(KK^K)$
Alpha–Investing [10]	$\mathcal{O}(K^3)$

Parameters K and N denote number of features and instances, respectively.

a dynamically varying threshold [9]) or such that the boundary region of the decision is maintained as little as possible [12], [13]. These methods are designed to handle sequentially arriving features during model training and select a *global common* subset of features that is used to classify all instances during testing. Table V summarizes the computational complexity of the baselines, as reported by their authors. The complexity of the proposed approach is discussed in Section V-B. The main reason for comparing with such methods is twofold. First, both these methods and the proposed algorithm are sequential (i.e., examine one feature at a time). Second, these baselines have been shown to outperform standard feature selection algorithms and scale well in high-dimensional settings. Similar to IFC^{2}F, ETANA and F-ETANA assume that all features are available during training, while during testing, features arrive sequentially one at a time for each data instance. However, ETANA and F-ETANA assume that features are conditionally independent given the class variable.

For a fair comparison, all streaming feature selection methods use a k-nearest-neighbor classifier with three neighbors to evaluate a selected feature subset, since it has been shown to outperform support vector machine, classification and regression tree, and J48 classifiers on the datasets used in [11] and [12]. At the same time, parameter α used by SAOLA, OSFS, and Fast-OSFS is set to 0.01, which has been shown to produce the best performance [10], [11]. The code for all baselines is either publicly available or has been provided by their authors. The same training and testing datasets are used by all methods. Finally, the same metrics (i.e., accuracy, number of features used, and time) used by the baselines are adopted. Observations from Tables VI–VIII are summarized next.

Madelon: IFC^{2}F achieves the highest accuracy. In fact, this corresponds to an improvement of 7.8% in accuracy with being 27.8% faster in joint feature selection and classification compared to ETANA, which has the second highest accuracy. ETANA, however, requires 18.7% less features on average compared to IFC^{2}F.

Lung Cancer: ETANA, F-ETANA, SAOLA, and Fast-OSFS achieve the highest accuracy, but require 50.4% to 3.75×10^3% more features on average and are 200% to 1.82×10^6% slower in joint feature selection and classification for a difference of 1.7% in accuracy compared to IFC^{2}F.

MLL: Both IFC^{2}F and ETANA achieve 100% accuracy. However, IFC^{2}F requires 32.9% less features on average and is 200% faster in joint feature selection and classification compared to ETANA.

Dexter: IFC^{2}F achieves the highest accuracy and is the fastest in joint feature selection and classification.

Car: F-ETANA achieves the highest accuracy (10.7% improvement), but requires 1.31×10^3% more features on average and is 1.26×10^3% slower in joint feature selection and classification compared to IFC^{2}F.

Lung2: OFS-Density achieves the highest accuracy, but requires 43.4% more features on average and is 5.55×10^4% slower in joint feature selection and classification compared to IFC^{2}F for a difference of 5.4% in accuracy.

Leukemia: IFC^{2}F, ETANA, and F-ETANA achieve the highest accuracy. However, IFC^{2}F requires 8.7% and 80.1% less features on average compared to ETANA and F-ETANA, respectively.

Prostate: ETANA achieves the highest accuracy, but is 50% slower in joint feature selection and classification compared to IFC^{2}F.

Spambase: OFS-A3M achieves the highest accuracy, but requires approximately nine times more features for a difference of 4.9% in accuracy and is much slower compared to IFC^{2}F.

Dorothea: IFC^{2}F and Fast-OSFS achieve the highest accuracy. However, Fast-OSFS requires approximately ten times more features on average and is much slower compared to IFC^{2}F.

Several observations can be drawn from the above results. For the majority of datasets, ETANA achieves the highest accuracy, while IFC^{2}F is competitive achieving higher or closely second performance compared to ETANA. This result demonstrates the fact that for some datasets, the assumption of features being conditionally independent given the class variable (used in ETANA) is more appropriate than assuming that features are dependent (used in IFC^{2}F). On the other hand, IFC^{2}F requires less number of features on average than ETANA, while OSFS consistently selects the least number of features among all baselines. This observation suggests that considering feature dependencies helps to get rid of redundant features. Furthermore, IFC^{2}F is the fastest algorithm to perform joint feature selection and classification in all the datasets compared to the baselines. This is due to the fact that it uses less number of features on average per data instance. Specifically, easy to classify data instances require few features as opposed to more challenging data instances that require more features to be accurately classified by IFC^{2}F.

E. Performance Assessment on a High-Dimensional Dataset

In this subsection, the performance of IFC^{2}F and the baselines is discussed within the context of the News20 dataset. Except for IFC^{2}F, ETANA, F-ETANA, and SAOLA, the rest of the methods were unable to generate results within a cutoff time of 12 days. Although SAOLA achieves the highest accuracy, it requires approximately six times more features and is ~160 times slower in joint feature selection and classification for a mere improvement of 4.6% in accuracy compared to IFC^{2}F (second last row in Tables VI–VIII). This experiment demonstrates the ability of IFC^{2}F to scale for more than 1.3 million features and provides further supporting evidence for the advantage of dynamically selecting features and classifying each data instance individually.

F. Statistical Significance

To validate the statistical significance of the results presented in Sections VI-D and VI-E, a Friedman test, which constitutes a well-known method to compare the performance of several

TABLE VI
COMPARISON OF ACCURACY

Dataset	IFC^{2}F	ETANA	F–ETANA	OFS–Density	OFS–A3M	SAOLA	Fast–OSFS	OSFS	Alpha–Investing
Madelon	**0.6700**	0.6217	0.5180	0.5117	0.5117	0.5817	0.5417	0.5817	0.6050
Lung Cancer	0.9724	**0.9890**	**0.9890**	0.9835	0.9779	**0.9890**	**0.9890**	0.9724	0.9613
MLL	**1.00**	**1.00**	0.9467	0.9600	0.9067	0.8667	0.8000	0.8000	0.9333
Dexter	**0.8533**	0.8133	0.7967	0.8527	0.7375	0.7800	0.7800	0.7967	0.5000
Car	0.7471	0.8097	**0.8274**	0.5973	0.7929	0.7982	0.6082	0.5575	0.6429
Lung2	0.8573	0.8820	0.8918	**0.9117**	0.8717	0.8817	0.8420	0.8471	0.8820
Leukemia	**0.9571**	**0.9571**	**0.9571**	0.9438	0.7914	0.9295	0.9295	0.8867	0.8324
Prostate	0.9005	**0.9310**	0.9010	0.9210	0.8148	0.8910	0.8633	0.8833	0.9014
Spambase	0.8104	0.8467	0.5109	0.7870	**0.8598**	0.8241	0.8011	0.8011	0.8074
Dorothea	**0.9429**	0.9400	0.7714	0.9314	0.9314	0.9114	**0.9429**	0.9000	0.6457
News20	0.7503	0.7352	0.6346	−−	−−	**0.7846**	−−	−−	−−
Avg. rank	3.41	2.32	4.36	4.82	6.27	4.55	6.18	6.95	6.14

The highest accuracy, and the second highest accuracy are bolded and gray-shaded, and gray-shaded, respectively. Cells are marked with "−−" if the corresponding method was unable to generate results within a cutoff time of 12 days.

TABLE VII
COMPARISON OF AVERAGE NUMBER OF FEATURES USED

Dataset	IFC^{2}F	ETANA	F–ETANA	OFS–Density	OFS–A3M	SAOLA	Fast–OSFS	OSFS	Alpha–Investing
Madelon	5.03	4.09	55.48	**2.00**	**2.00**	3.00	3.00	3.00	4.00
Lung Cancer	**1.35**	2.03	6.56	37.20	8.40	52.00	6.80	4.00	4.60
MLL	3.40	5.07	14.69	11.00	12.00	28.00	5.00	**3.00**	7.00
Dexter	10.95	12.80	243.4	10.00	104.0	21.00	9.00	6.00	**1.00**
Car	24.20	12.90	340.20	6.80	36.00	41.40	8.40	**5.20**	24.40
Lung2	11.30	15.59	27.91	16.20	18.00	28.20	9.40	**5.80**	34.40
Leukemia	**1.90**	2.08	9.53	4.40	13.40	21.60	4.60	2.20	3.20
Prostate	4.08	3.34	10.39	5.80	40.20	14.00	3.80	**1.60**	7.00
Spambase	**4.72**	7.47	56.00	7.60	42.20	24.60	33.80	33.80	42.60
Dorothea	**2.29**	2.89	8.10	17.40	34.00	32.00	24.00	3.00	113.0
News20	**43.68**	81.70	4000.6	−−	−−	241.8	−−	−−	−−
Avg. rank	2.91	3.36	7.09	4.68	6.86	6.91	4.41	2.86	5.91

The minimum and the second minimum average number of features used are bolded and gray-shaded, and gray-shaded, respectively. Cells are marked with "−−" if the corresponding method was unable to generate results within a cutoff time of 12 days.

algorithms across multiple datasets [33], is conducted. The average ranking (avg. rank) of each method is given in the last row in Tables VI–VIII. The p-values of the Friedman test on classification accuracy, the average number of features used and time required for joint feature selection and classification are 1.69×10^{-4}, 8.07×10^{-7} and 4.55×10^{-24}, respectively. Thus, there is a significant difference [33] in the performance of IFC^{2}F and the baselines.

G. Demonstration of Instance-Wise Feature Selection

Table IX demonstrates the instance-wise nature of IFC^{2}F using four illustrative examples from the IMDB movie reviews dataset (50 000 instances, 89 523 features, and two classes) [34]. The IMDB dataset is selected because the raw text of reviews is available and can be directly used to interpret the classification rationale. The training and validation sets with bag-of-words features are used as provided. The Markov blanket based feature ordering is { "bad," "great," "no," "best," "even," "plot," "nothing," "love," "don't," "waste," . . . }. Fig. 4 illustrates the evolution of the posterior probability distribution π_k as more and more features are evaluated, until the stopping condition $g(\pi_k) \leqslant \mathcal{A}_k(\pi_k)$ is satisfied. At that time, the instance is assigned to the class with the maximum posterior probability; this is a direct result of using constant missclassification costs, i.e., $Q_{01} = Q_{10} = 1, Q_{00} = Q_{11} = 0$ [see (6)]. Observe that the proposed framework evaluates more features to predict challenging reviews such as (c) and (d) compared to easy and straightforward reviews such as (a) and (b). In summary, IFC^{2}F selects *different features* for *different data instances* in a dynamic setting and assigns the class label based on the observed features.

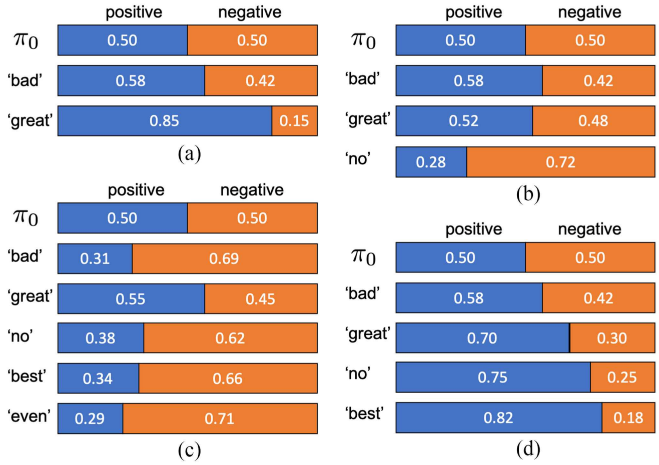


Fig. 4. (a)–(d) Evolution of posterior probability distribution, i.e., $P(\text{positive}|F_1, \ldots, F_k)$ (in blue) and $P(\text{negative}|F_1, \ldots, F_k)$ (in orange) for four IMDB reviews in Table IX.

H. Robustness to Missing Features

In this subsection, the ability of IFC^{2}F to handle missing features is evaluated. Specifically, $x\%$ of features are randomly removed from each test instance and the posterior probability π_k is kept unchanged if a feature is missing. $x\%$ is increased

TABLE VIII
COMPARISON OF TIME (IN SECONDS) REQUIRED FOR FEATURE SELECTION (F), CLASSIFICATION (C), JOINT FEATURE SELECTION AND CLASSIFICATION (F+C), MODEL TRAINING (T), AND PREPROCESSING (P)

Dataset	Time	IFC^{2}F	ETANA	F–ETANA	Time	OFS–Density	OFS–A3M	SAOLA	Fast–OSFS	OSFS	Alpha–Investing
Madelon	F+C	0.070	0.097	0.428	F	129.8	151.1	0.048	0.076	0.080	**0.033**
					C	0.011	0.010	0.010	0.011	0.011	**0.010**
	T	0.155	0.247	0.748	T	0.074	0.075	0.075	0.073	0.073	0.076
	P	35.58	0.142	0.131							
Lung Cancer	F+C	**0.001**	0.003	0.003	F	12.80	40.43	2.465	1.279	18.23	0.459
					C	0.011	0.012	0.013	0.010	0.010	0.010
	T	0.213	11.79	1.340	T	0.071	0.089	0.068	0.068	0.070	0.072
	P	146.0	1.806	1.796							
MLL	F+C	**0.001**	0.003	0.003	F	1.468	5.543	1.513	0.564	4.679	0.154
					C	0.004	0.011	0.013	0.010	0.010	0.010
	T	0.448	24.63	3.193	T	0.008	0.071	0.069	0.071	0.071	0.073
	P	291.2	0.133	0.133							
Dexter	F+C	**0.054**	0.125	0.681	F	77.88	48453	0.747	1.087	2.509	12.98
					C	0.048	0.180	0.060	0.038	0.033	0.024
	T	0.290	22.18	2.106	T	0.090	0.073	0.069	0.067	0.063	0.089
	P	35.91	0.133	0.135							
Car	F+C	**0.029**	0.044	0.395	F	8.236	40.55	1.155	0.999	13.40	0.710
					C	0.011	0.013	0.013	0.010	0.010	0.013
	T	4.950	3059.5	37.21	T	0.070	0.066	0.067	0.070	0.070	0.073
	P	462.5	361.6	4.157							
Lung2	F+C	**0.008**	0.052	0.021	F	4.437	16.52	0.702	0.840	14.54	0.366
					C	0.012	0.013	0.013	0.012	0.010	0.013
	T	0.120	782.4	6.901	T	0.070	0.067	0.066	0.072	0.071	0.070
	P	28.80	0.905	0.798							
Leukemia	F+C	**0.001**	**0.001**	0.002	F	1.793	5.553	0.649	0.433	1.079	0.182
					C	0.337	0.011	0.013	0.010	0.010	0.011
	T	0.103	8.043	2.162	T	0.270	0.069	0.068	0.068	0.070	0.072
	P	56.92	1.000	1.002							
Prostate	F+C	**0.001**	0.002	0.002	F	2.116	6.010	0.487	0.362	0.748	0.167
					C	0.011	0.012	0.012	0.010	0.010	0.010
	T	0.183	7.088	1.620	T	0.070	0.067	0.067	0.069	0.071	0.072
	P	84.57	0.822	0.813							
Spambase	F+C	**0.066**	0.237	0.482	F	53.58	65.03	0.083	38.79	387.6	0.049
					C	0.016	0.027	0.031	0.033	0.033	0.038
	T	0.030	0.285	0.012	T	0.086	0.006	0.087	0.067	0.065	0.069
	P	2.321	0.0283	0.0296							
Dorothea	F+C	**0.013**	0.035	0.033	F	2152.7	174610	16.99	53.52	518.5	217.1
					C	0.051	0.113	0.102	0.074	0.027	0.316
	T	0.213	204.6	7.208	T	0.005	0.178	0.069	0.067	0.070	0.067
	P	744.3	15.06	15.08							
News20	F+C	**9.190**	117.61	346.47	F	−−	−−	1444.8	−−	−−	−−
					C	−−	−−	36.26	−−	−−	−−
	T	0.5564	3076.1	429.7	T	−−	−−	0.106	−−	−−	−−
	P	2064.4	1335.2	1331.0							
Avg. rank		1.23	2.82	3.23		7.36	8.73	4.82	5.27	7.27	4.27

The minimum and the second minimum F+C times are bolded and gray-shaded, and gray-shaded, respectively. Cells are marked with "−−" if the corresponding method was unable to generate results within a cutoff time of 12 days.

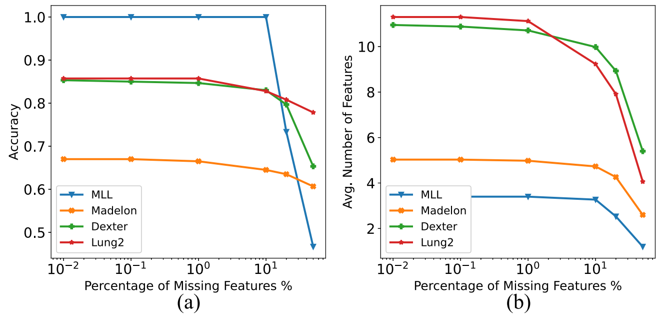


Fig. 5. Variation of (a) accuracy and (b) average number of features as the percentage of missing features increases from 0.01% to 50% across datasets.

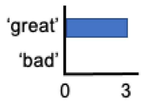
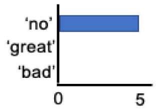
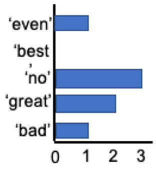
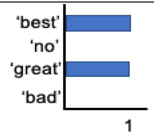
from 0.01% to 50%, and the effect on the accuracy and the average number of features used for classification is noted (see Fig. 5). Evidently, IFC^{2}F is robust for up to 10% missing features, beyond which value, the posterior probability π_k may no longer represent the underlying true class value. Thus, IFC^{2}F can identify informative features to make accurate predictions when a small subset of features is missing.

VII. CONCLUSION AND FUTURE DIRECTIONS

In this article, a framework to perform dynamic instance-wise joint feature selection and classification with correlated features is proposed. Specifically, feature dependencies are modeled using a Bayesian network. Based on the learned dependence network, a method is proposed to sequentially select the most informative features and reach a classification decision for each instance individually. The effectiveness and scalability of the proposed method is illustrated on various real-world datasets. The proposed method robustly performs well on all of them, with comparable and often superior performance compared to prior art.

The proposed method selects the most informative features from the dependence graph utilizing the proposed Markov-blanket-based feature ordering. This dependence graph, however, is learned offline during training; hence, the ordering in

TABLE IX
WORDS (FEATURES) PICKED BY IFC^{2}F ARE HIGHLIGHTED IN YELLOW

IMDB Review Text **(True Label, Predicted Label)**	
I work at a movie theater and every Thursday night we have an employee screening of one movie that comes out the next day. Today it was The Guardian. I saw the trailers and the ads and never expected much from it, and in no way really did i anticipate seeing this movie. Well turns out this movie was a lot more than I would have thought. It was a great story first of all. Ashton Kutcher and Kevin Costner did amazing acting work in this film. Being a big fan of That 70's Show I always found it hard thinking of Kutcher as anyone but Kelso despite the great acting he did in The Butterfly Effect, but after seeing this movie I think I might be able to finally look at him as a serious actor. It was also a great tribute to the unsung heroes of the U.S. Coast Guard. **(positive, positive)**	
I saw this only because my 10-yr-old was bored. He and his friend hated it but of course liked being at the movies. This is the first time I've strongly disagreed with Ebert in many years. There is not a single thing to recommend this film. Willis is good, as always. But the story stinks, is unbelievable, there is no real story, no action, no interesting cinematic sequences, no surprises, and worst of all, the child star is A thoroughly repulsive slug guaranteed to turn off any parent who does not have a dweeby fat slob for a kid. By all means stay away and spare your child - unless you want to punish him or her. There is no excuse for such lousy directing or writing and one hopes these filmmakers will suffer accordingly. **(negative, negative)**	
I felt compelled to comment on this film because it's listed as the fourth lowest-rated sci-film of all time on the IMDb. WHAT!?!? Sure, this movie is crappy, but it's HILARIOUS! It's not awful on an Ed Wood level, it's more surreal and uneven. There are some classic moments in the film. The brain surgery is gross and great and even nuttier when you consider that the film was rated PG! Gor chasing after his dolly before getting battery acid dumped on his face- "Mine! Gimmee!" Zandor Vorkoff's speeches at the beginning of the film- "Before Amir, Kali was but another weak nation struggling to break free from centuries of stagnant feudalism!" Angelo Rossito also has some great lines- "No, Gor! No!" "You want these keys, don't you, my pretties?" It is absolutely wrong that this is the 4th lowest-rated sci-film on the IMDb because it is ENTERTAINING. No matter how bad a film is, if it still manages to be weird, quirky, unsettling, or entertaining, it has merit and doesn't deserve to be dumped on and dismissed. I won't defend most of Al Adamson's films, but this one, along with Dracula VS. FRANKENSTEIN and BLOOD OF GHASTLY HORROR, are entertaining enough to make up for their awfulness. **(positive, negative)**	
Jack Lemmon was one of our great actors. His performances in Days Of Wine And Roses, The Apartment, Some Like It Hot, Missing (to name the first ones that come to mind) were all worthy of Best Actor nomination. His only win was for Save The Tiger, and that's a shame. He gets melancholy down to a science, but never brings it into balance with the driver in his character. He actually did a similar character much better toward the end of his career in the one-note Glengarry Glen Ross. As for the movie, wonderful supporting work by Jack Gilford as Lemmon's partner and Thayer David as an arsonist, go for naught because the rest of the script is a muddled jumble of cliched vignettes, angst, neurotic nostalgia, and pointless moralizing. Worth seeing once as a time capsule into 1970's style experimental direction by Avildsen. **(negative, positive)**	

The true/predicted label is given at the end of each review. The second column reports features selected for each review in ascending order (Y-axis) versus feature value (X-axis).

which features are selected is common for all test instances. In future work, the goal is to address this limitation by dynamically learning the network structure, since the number of selected features and the interpretability of the classification depend on the graph structure at hand. At the same time, to keep the preprocessing time small, the proposed method opts for filtering out features based on the mutual information between each feature and the class label. In future work, approaches such as multivariate mutual information can be explored to better capture feature dependencies. Finally, the proposed method assumes that all data instances are available at once during training, which may not hold in cases where data instances are provided sequentially. In the future, the applicability of online learning methods [35] in the proposed setting can be considered.

APPENDIX A
PROOF OF THEOREM 1

At the end of the Kth stage, assuming that all the features have been examined, the only remaining expected cost is the optimum misclassification cost of selecting among L classes, which is $\bar{J}_K(\pi_k) = g(\pi_k)$.

Then, consider any intermediate stage $k = 0, 1, \ldots, K-1$. Being at stage k, with available information π_k, the optimum strategy has to choose between, either to terminate and incur cost $g(\pi_k)$, which is the optimum misclassification cost of selecting among L classes, or continue with the next feature F_{k+1}, and incur cost e_{k+1} and an additional cost $\bar{J}_{k+1}(\pi_{k+1})$ to continue optimally at stage $k+1$. Thus, the total cost of continuing optimally is $e_{k+1} + \bar{J}_{k+1}(\pi_{k+1})$. However, at stage k, the assignment f_{k+1} of the next feature F_{k+1} is not known. Thus, the expected *optimum cost-to-go*, which is equal to $e_{k+1} + \mathbb{E}\{\bar{J}_{k+1}(\pi_{k+1})|\pi_k\}$, needs to be considered. Using Bayes' rule to express π_{k+1} in terms of π_k, and by the definition of the expectation operator, the *optimum cost-to-go* $\mathcal{A}_k(\pi_k)$ takes the following form:

$$\begin{aligned}\mathcal{A}_k(\pi_k) &\triangleq e_{k+1} + \mathbb{E}\left\{\bar{J}_{k+1}(\pi_{k+1})|\pi_k\right\}\\ &= e_{k+1} + \sum_{F_{k+1}} P(F_{k+1}|F_1, \ldots, F_k)\\ &\quad\times \bar{J}_{k+1}\left(\frac{\operatorname{diag}\left(\Delta(F_{k+1}|F_1, \ldots, F_k, \mathcal{C})\right)\pi_k}{\Delta^T(F_{k+1}|F_1, \ldots, F_k, \mathcal{C})\pi_k}\right). \end{aligned} \tag{9}$$

Next, the term $P(F_{k+1}|F_1, F_2, \ldots, F_k)$ must be simplified. Specifically, using the Bayes' rule and the law of total probability, it can be shown that

$$\begin{aligned} P&\left(F_{k+1}|F_1, F_2, \ldots, F_k\right)\\ &= \frac{P(F_1, F_2, \ldots, F_{k+1})}{P(F_1, F_2, \ldots, F_k)}\\ &= \frac{\sum_{j=1}^{L} P(F_1, \ldots, F_{k+1}\mathcal{C}_j)}{\sum_{j=1}^{L} P(F_1, \ldots, F_k, \mathcal{C}_j)}\\ &= \frac{\sum_{j=1}^{L} P(F_1, \ldots, F_{k+1}|\mathcal{C}_j)P(\mathcal{C}_j)}{\sum_{j=1}^{L} P(F_1, \ldots, F_k|\mathcal{C}_j)P(\mathcal{C}_j)}. \end{aligned} \tag{10}$$

Using the chain rule, (10) can be simplified as follows:

$$\begin{aligned} &P\left(F_{k+1}|F_1, F_2, \ldots, F_k\right)\\ &= \frac{\sum_{j=1}^{L} P(F_1, \ldots, F_k|\mathcal{C}_j)P(F_{k+1}|F_1, \ldots, F_k, \mathcal{C}_j)P(\mathcal{C}_j)}{\sum_{j=1}^{L} P(F_1, \ldots, F_k|\mathcal{C}_j)P(\mathcal{C}_j)} \end{aligned}$$

$$=\sum_{j=1}^{L}\frac{P(F_1,\ldots,F_k|\mathcal{C}_j)P(\mathcal{C}_j)}{\sum_{j=1}^{L}P(F_1,\ldots,F_k|\mathcal{C}_j)P(\mathcal{C}_j)}P(F_{k+1}|F_1,\ldots,F_k,\mathcal{C}_j)$$

$$=\sum_{j=1}^{L}P(\mathcal{C}_j|F_1,\ldots,F_k)P(F_{k+1}|F_1,\ldots,F_k,\mathcal{C}_j)$$

$$=\sum_{j=1}^{L}\pi_k^jP(F_{k+1}|F_1,\ldots,F_k,\mathcal{C}_j)$$

$$=\Delta^T(F_{k+1}|F_1,\ldots,F_k,\mathcal{C})\pi_k. \tag{11}$$

Finally, substituting (11) into (9), the desired result can be acquired

$$\mathcal{A}_k(\pi_{k+1})=e_{k+1}+\sum_{F_{k+1}}\Delta^T(F_{k+1}|F_1,\ldots,F_k,\mathcal{C})\pi_k \times\bar{J}_{k+1}\left(\frac{\text{diag}(\Delta(F_{k+1}|F_1,\ldots,F_k,\mathcal{C}))\pi_k}{\Delta^T(F_{k+1}|F_1,\ldots,F_k,\mathcal{C})\pi_k}\right) \tag{12}$$

which completes the proof.

Appendix B
Proof of Lemma 1

Consider the definition of $g(\varpi)$:

$$g(\varpi)\triangleq\min_{1\leqslant j\leqslant L}\left[Q_j^T\varpi\right],\quad \varpi\in[0,1]^L.$$

The term $Q_j^T\varpi$ is linear with respect to ϖ, and since the minimum of linear functions is a concave piecewise linear function, $g(\varpi)$ is a concave piecewise linear function as well. Concavity also ensures the continuity of this function. Minimization over finite L hyperplanes guarantees that the function $g(\varpi)$ is made up of at most L hyperplanes. Hence, the set $\{Q_j^T\}_{j=1}^L$ of L vectors represents those L hyperplanes.

Appendix C
Proof of Lemma 2

First, consider the function $\mathcal{A}_{K-1}(\varpi)$ given by

$$\mathcal{A}_{K-1}(\varpi)=e_K+\sum_{F_K}\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi \times\bar{J}_K\left(\frac{\text{diag}\left(\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\right)\varpi}{\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi}\right). \tag{13}$$

Using the fact that $\bar{J}_K(\pi_K)=g(\pi_K)$, (13) can be rewritten as follows:

$$\mathcal{A}_{K-1}(\varpi)=e_K+\sum_{F_K}\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi \times g\left(\frac{\text{diag}\left(\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\right)\varpi}{\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi}\right). \tag{14}$$

Using the definition of $g(\varpi)$, (14) can be rewritten as follows:

$$\mathcal{A}_{K-1}(\varpi)=e_K+\sum_{F_K}\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi \times\min_{1\leqslant j\leqslant L}\left[\frac{Q_j^T\,\text{diag}\left(\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\right)\varpi}{\Delta^T(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\varpi}\right]. \tag{15}$$

Using the facts that Q_j and $\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C})$ are nonnegative vectors, (15) can be simplified as follows:

$$\mathcal{A}_{K-1}(\varpi)=e_K +\sum_{F_K}\min_{1\leqslant j\leqslant L}\left[Q_j^T\,\text{diag}\left(\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C})\right)\varpi\right]. \tag{16}$$

Note that the term $Q_j^T\,\text{diag}(\Delta(F_K|F_1,\ldots,F_{K-1},\mathcal{C}))\varpi$ is linear with respect to ϖ. Using the facts that 1) $e_K>0$, 2) the minimum of linear functions is a concave piecewise linear function, and 3) the nonnegative sum of concave/piecewise linear functions is also a concave/piecewise linear function implies that $\mathcal{A}_{K-1}(\varpi)$ is a concave piecewise linear function. Concavity also ensures the continuity of this function.

Then, consider the function $\mathcal{A}_{K-2}(\varpi)$ given by

$$\mathcal{A}_{K-2}(\varpi)=e_{K-1}+\sum_{F_{K-1}}\Delta^T(F_{K-1}|F_1,\ldots,F_{K-2},\mathcal{C})\varpi \times\bar{J}_{K-1}\left(\frac{\text{diag}\left(\Delta(F_{K-1}|F_1,\ldots,F_{K-2},\mathcal{C})\right)\varpi}{\Delta^T(F_{K-1}|F_1,\ldots,F_{K-2},\mathcal{C})\varpi}\right). \tag{17}$$

Note that $\bar{J}_{K-1}(\varpi)=\min[g(\varpi),\mathcal{A}_{K-1}(\varpi)]$ (see Theorem 1). Using the facts that 1) $g(\varpi)$ is a concave, piecewise linear function, 2) $\mathcal{A}_{K-1}(\varpi)$ is a concave, piecewise linear function, and 3) the minimum of two concave/piecewise linear functions is also a concave/piecewise linear function implies that $\bar{J}_{K-1}(\varpi)$ is also concave and piecewise linear. Furthermore, the nonnegative sum of concave/piecewise linear functions is also a concave/piecewise linear function. Based on this fact and the facts that $e_{K-1}>0$ and $\Delta(F_{K-1}|F_1,\ldots,F_{K-2},\mathcal{C})$ is a nonnegative vector, the function $\mathcal{A}_{K-2}(\varpi)$ is concave and piecewise linear. Concavity also ensures the continuity of this function. Using similar arguments, the concavity, the continuity, and the piecewise linearity of functions $\mathcal{A}_k(\varpi), k=0,\ldots,K-3$, can also be guaranteed.

Appendix D
Proof of Theorem 2

At the final stage, i.e., $k=K$, $\bar{J}_K(\varpi)=g(\varpi)=\min_{1\leqslant j\leqslant L}[Q_j^T\varpi]$. Hence, $\{\alpha_K^i\}\triangleq\{Q_j^T\}_{j=0}^L$. The rest of the proof is very intuitive. Using the facts that 1) $g(\varpi)$ and $\mathcal{A}_k(\varpi)$ are concave and piecewise linear with respect to ϖ, 2) $\bar{J}_k(\varpi)=\min[g(\varpi),\mathcal{A}_k(\varpi)], k\in\{0,\ldots,K-1\}$ (see Theorem 1), and 3) the minimum of two concave/piecewise linear functions is also a concave/piecewise linear function implies that the function $\bar{J}_k(\varpi)$ is also concave and piecewise linear. Finally, since $\bar{J}_k(\varpi)$ is a concave and piecewise linear function defined on a probability space, it is noted that $\bar{J}_k(\varpi)=\min_i[\alpha_k^i\varpi]$, where the set $\{\alpha_k^i\}_i$ of vectors represents its linear pieces.

References

[1] D. Belk, "Diagnostic tests—True cost of heathcare," 2020. [Online]. Available: https://truecostofhealthcare.org/diagnostic-tests/

[2] D. P. Kao *et al.*, "Characterization of subgroups of heart failure patients with preserved ejection fraction with possible implications for prognosis and treatment response," *Eur. J. Heart Failure*, vol. 17, no. 9, pp. 925–935, 2015.

[3] Z. Zhao and H. Liu, "Searching for interacting features in subset selection," *Intell. Data Anal.*, vol. 13, no. 2, pp. 207–228, 2009.

[4] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active classification: Theory and application to underwater inspection," in *Robot. Res.*. Berlin, Germany: Springer, 2017, pp. 95–110.

[5] Y. W. Liyanage, D.-S. Zois, and C. Chelmis, "Dynamic instance-wise joint feature selection and classification," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 169–184, Apr. 2021.

[6] C. Molnar, "Interpretable machine learning," 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[7] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, "A survey on online feature selection with streaming features," *Front. Comput. Sci.*, vol. 12, no. 3, pp. 479–493, 2018.

[8] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 592–599.

[9] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2005, pp. 384–393.

[10] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, May 2013.

[11] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 660–669.

[12] P. Zhou, X. Hu, P. Li, and X. Wu, "OFS-density: A novel online streaming feature selection method," *Pattern Recognit.*, vol. 86, pp. 48–61, 2019.

[13] P. Zhou, X. Hu, P. Li, and X. Wu, "Online streaming feature selection using adapted neighborhood rough set," *Inf. Sci.*, vol. 481, pp. 258–279, 2019.

[14] Y. Wu, S. C. Hoi, T. Mei, and N. Yu, "Large-scale online feature selection for ultra-high dimensional sparse data," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 4, 2017, Art. no. 48.

[15] J. Chen, L. Song, M. Wainwright, and M. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 883–892.

[16] J. Yoon, J. Jordon, and M. van der Schaar, "INVASE: Instance-wise variable selection using neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[17] Q. Xiao and Z. Wang, "Mixture of deep neural networks for instance-wise feature selection," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput.*, 2019, pp. 917–921.

[18] G. Dulac-Arnold, L. Denoyer, P. Preux, and P. Gallinari, "Datum-wise classification: A sequential approach to sparsity," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2011, pp. 375–390.

[19] J. Janisch, T. Pevný, and V. Lisỳ, "Classification with costly features as a sequential decision-making problem," *Mach. Learn.*, vol. 109, pp. 1587–1615, 2020.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[21] R. E. Neapolitan, *Learning Bayesian Networks*, vol. 38, Upper Saddle River, NJ, USA: Pearson/Prentice-Hall, 2004.

[22] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[24] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 2005.

[25] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1/2, pp. 99–134, 1998.

[26] M. T. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *J. Artif. Intell. Res.*, vol. 24, pp. 195–220, 2005.

[27] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Auton. Agents Multi-Agent Syst.*, vol. 27, no. 1, pp. 1–51, 2013.

[28] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[29] N. Friedman and M. Goldszmidt, "Building classifiers using Bayesian networks," in *Proc. Nat. Conf. Artif. Intell.*, 1996, pp. 1277–1284.

[30] K. Yang, Z. Cai, J. Li, and G. Lin, "A stable gene selection in microarray data analysis," *BMC Bioinform.*, vol. 7, no. 1, 2006, Art. no. 228.

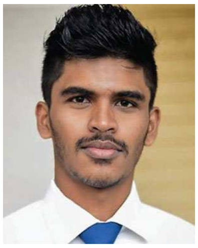
[31] *Clopinet: Feature Selection Challenge*, Accessed: Apr. 2021. [Online]. Available: http://clopinet.com/isabelle/Projects/NIPS2003/

[32] *LIBSVM data sets*, Accessed: Apr. 2021. [Online]. Available: https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

[33] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[34] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2011, pp. 142–150.

[35] S. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021.



Yasitha Warahena Liyanage (Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2016. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University at Albany, Albany, NY, USA.

His research interests include quickest change detection, optimal stopping theory, and machine learning.

Daphney-Stavroula Zois (Member, IEEE) received the B.S. degree in computer engineering and informatics from the University of Patras, Patras, Greece in 2007, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, CA, USA 2010 and 2014.

She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University at Albany, Albany, NY, USA. Her previous appointments include the University of Illinois, Urbana–Champaign, Urbana, IL, USA. Her research interests include decision making under uncertainty, machine learning, detection and estimation theory, intelligent systems design, and signal processing.

Dr. Zois received the Viterbi Dean's and Myronis Graduate Fellowships and the National Science Foundation CAREER Award. She has served and is serving as co-Chair, technical program committee member, or reviewer in international conferences and journals, such as AAAI Conference on Artificial Intelligence, IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE Global Conference on Signal and Information Processing, International Conference on Learning Representations, and IEEE Transactions on Signal Processing.

Charalampos Chelmis (Member, IEEE) received the B.S. degree in computer engineering and informatics from the University of Patras, Patras, Greece, in 2007, and the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA, USA, in 2010 and 2013, respectively.

He is currently an Assistant Professor of Computer Science with the University at Albany, Albany, NY, USA, and the Director of the Intelligent Big Data Analytics, Applications, and Systems Lab, focusing on problems involving big, often networked, data.

Dr. Chelmis has served and is serving as Co-Chair, Technical Program Committee Member, or Reviewer in numerous international conferences and journals such as the International World Wide Web Conference, International Conference on Advances in Social Network Analysis and Mining, and International Conference on Weblogs and Social Media. He is currently an Associate Editor for Social Network Analysis and Mining Journal and the Journal of Parallel and Distributed Systems. He was a Guest Editor for the Encyclopedia of Social Network Analysis and Mining.