Two Methods for Constructing a Gene Ontology-based Feature Network for a Bayesian Network Classifier and Applications to Datasets of Aging-related Genes

Cen Wan
School of Computing
University of Kent
Canterbury, United Kingdom
cw439@kent.ac.uk

Alex A. Freitas
School of Computing
University of Kent
Canterbury, United Kingdom
A.A.Freitas@kent.ac.uk

ABSTRACT

In the context of the classification task of data mining or machine learning, hierarchical feature selection methods exploit hierarchical relationships among features in order to select a subset of features without hierarchical redundancy. Hierarchical feature selection is a new research area in classification research, since nearly all feature selection methods ignore hierarchical relationships among features. This paper proposes two methods for constructing a network of features to be used by a Bayesian Network Augmented Naïve Bayes (BAN) classifier, in datasets of aging-related genes where Gene Ontology (GO) terms are used as hierarchically related predictive features. One of the BAN network construction method relies on a hierarchical feature selection method to detect and remove hierarchical redundancies among features (GO terms); whilst the other BAN network construction method simply uses a conventional, flat feature selection method to select features, without removing the hierarchical redundancies associated with the GO. Both BAN network construction methods may create new edges among nodes (features) in the BAN network that did not exist in the original GO DAG (Directed Acyclic Graph), in order to preserve the generalization-specialization (ancestor-descendant) relationship among selected features. Experiments comparing these two BAN network construction methods, when using two different hierarchical feature selection methods and one flat feature selection method, have shown that the best results are obtained by the BAN network construction method using one type of hierarchical feature selection method, i.e., select Hierarchical Information-Preserving features (HIP).

Categories and Subject Descriptors

[**Computing methodologies**]: Machine learning—*Machine learning algorithms, Machine learning approaches, Learning in probabilistic graphical models, Bayesian network models*

BCB '15, September 09-12, 2015, Atlanta, GA, USA
© 2015 ACM. ISBN 978-1-4503-3853-0/15/09...$15.00
DOI: http://dx.doi.org/10.1145/2808719.2808722

General Terms

Algorithms, Experimentation, Performance

Keywords

Hierarchical Feature Selection, Bayesian Network Augmented Naïve Bayes, Gene Ontology, Aging

1. INTRODUCTION

This work addresses the classification task of data mining or machine learning, where the set of instances to be classified represents aging-related genes, the predictive features describing those genes are Gene Ontology terms (describing functions or properties of genes) [15] and the binary class variable (whose value is to be predicted) indicates if an instance is a "pro-longevity" or "anti-longevity" gene. Pro-longevity genes are defined as those genes whose decreased expression (due to knock-out, mutations or RNA interference) reduces lifespan, and/or those whose over-expression extends lifespan; *vice versa*, anti-longevity genes are defined as the genes whose decreased expression extends lifespan, and/or those whose over-expression decreases it [17].

Feature selection methods aim at improving the predictive accuracy of a classification algorithm by removing redundant or irrelevant features (attributes) [12]. We focus on feature selection methods executed in a data pre-processing phase, before building the classification model that will be used to classify new instances (in the actual classification phase). Pre-processing feature selection methods can be categorized into filter and wrapper methods. Filter methods evaluate the quality of feature subsets by using an evaluation function that is independent from the classification algorithm to be used in the classification phase. By contrast, wrapper methods evaluate the quality of feature subsets by using an evaluation function which is specifically based on the predictive performance of the classification algorithm to be used in the classification phase. In this work we focus on the filter approach, since it is much more efficient (faster) than the wrapper approach – because in the wrapper approach the classification algorithm has to be run a large number of times, once for each candidate feature subset being evaluated, which is very time-consuming.

There are a large number of pre-processing feature selection methods proposed in the literature, but nearly all of them are flat methods in the sense that they implicitly assume that there is no hierarchical relationship among the set of features. By contrast, in this work we focus on a new type of feature selection methods, here called hierarchical feature selection methods, which take into account hierarchical generalization-specialization relationships among features in order to perform a more effective feature selection process by detecting and removing hierarchical redundancies (a term to be precisely defined later) among features.

To the best of our knowledge, our previous works [16, 17] seem to be the first two papers proposing hierarchical feature selection methods for the classification task. Most of those hierarchical feature selection methods work with the well-known and simple Naïve Bayes classification algorithm, which assumes that features are independent from each other given the class value. However, one of the methods proposed in [16] was a more sophisticated Bayesian network augmented Naïve Bayes (BAN) algorithm, which allows each feature (represented as a node in the network) to depend on one or more features (represented as parent nodes in the network). In that algorithm, the network topology was directly defined by the hierarchical relationships among the Gene Ontology (GO) terms used as predictive features, i.e., there was an edge in the BAN network for each edge in the GO graph, as will be explained in more detail later. We refer to this type of algorithm as "GO-hierarchy-aware BAN" (GO–BAN).

The contribution of this current work is to propose a new approach for constructing the feature (GO term) network used by the GO–BAN algorithm. The basic idea of this approach consists of two steps. First, we use a feature selection algorithm to select features. Second, we construct the GO–BAN network by using new artificially created edges that directly connect features (nodes) which are hierarchically related but are distant (separated by more than one edges) in the GO graph. That is, we create a new edge that directly connects each selected feature (node) to its nearest selected ancestor, even though that ancestor might be more than one edges away in the original graph. We report the results of experiments evaluating this new approach for constructing the GO–BAN network when using the two most successful hierarchical feature selection methods proposed in [17]; as well as when using a well-known flat feature selection method, Correlation-based Feature Selection (CFS) [9], used here as a baseline method.

It is worth mentioning that the Gene Ontology is a popular resource for protein or gene function annotation, using a unified and standardized vocabulary to describe the functions of genes [15]. In terms of using GO terms as features on aging-related research, Freitas et al. [7] used GO terms and protein-protein interaction information as features to classify DNA repair genes into aging-related or non-aging related categories; whilst Fang et al. [6] used GO terms and protein-protein interaction information to classify aging-related genes into DNA repair or non-DNA repair genes. Note, however, that these works treated GO terms as flat features, rather than directly exploiting the GO terms' hierarchical generalization-specialization relationships to perform more effective feature selection, as is the case in this current work.

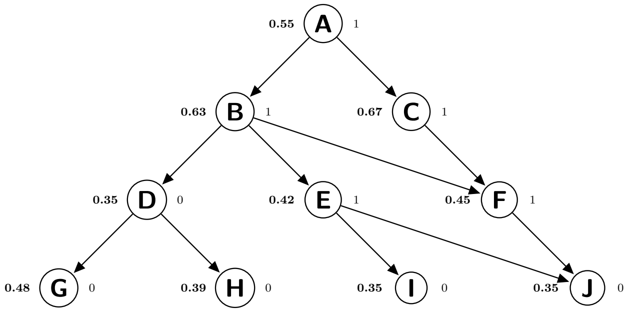

Figure 1: Example of A Small DAG of Features

This paper is organized as follows. Section 2 briefly reviews the Gene Ontology and its hierarchical structure, Gene Ontology-based Bayesian Network Augmented Naïve Bayes (GO–BAN), eager learning and lazy learning, flat and hierarchical feature selection methods. Section 3 describes the newly proposed methods to construct the feature network used by the GO–BAN classifier. Section 4 presents the experimental methods and computational results, followed by discussion in Section 5. Finally, a conclusion and future research directions are presented in Section 6.

2. BACKGROUND

2.1 The Hierarchical Structure of the Gene Ontology

The Gene Ontology (GO) is a major and very popular bioinformatics resource that uses structured and unified vocabularies to describe gene functions [15]. The GO consists of three types of GO terms: biological process, molecular function and cellular component terms. Majority of GO terms are structured by an "is-a" relationship, represented by a Directed Acyclic Graph (DAG). Each GO term is represented by a node in a DAG, and an edge from node A to node B indicates that A is a parent of (more generic than) B, or conversely, B is a specific case of the more generic A. For instance, in the DAG of biological process GO terms, term GO:0008150 (biological process) is the root term and parent of term GO:0008152 (metabolic process), which is in turn the parent of term GO:0006807 (nitrogen compound metabolic process).

In this work, we use GO terms as binary features which take values "1" or "0", indicating whether or not (respectively) each GO term is annotated for an individual aging-related gene. The type of "is-a" hierarchical relationship among the GO terms leads to some redundancies in hierarchically related feature values. More precisely, for each instance, if the value of a given feature is "1" in that instance, this implies that the values of all its ancestor features (in the GO DAG) are also "1" in that instance. Conversely, if the value of a given feature is "0" in an instance, this implies that the values of all its descendent features are also "0" in that instance. For example, consider Figure (1), an example DAG of hierarchically organized features where the figures on the left side of nodes denote the relevance values (discussed later) and the figures on the right side of nodes denote the current instance's values. Since node E has value "1", both its ancestor nodes B and A also have value "1"; and since node D has value “0”, both its descendant nodes G and H also have value “0”.

2.2 Gene Ontology–based Bayesian Network Augmented Naïve Bayes (GO–BAN)

Bayesian Network Augmented Naïve Bayes (BAN) is a type of semi-Naïve Bayesian classification algorithm that allows features to be dependent on other features. In a BAN for the classification task of data mining, each node in the network represents a feature (a GO term in our case) and each edge pointing from a node A to another node B represents that feature B depends on its parent feature A [8, 16]. In addition, the class variable is a parent of all the other nodes (all features) in the network. In order to classify a new instance, BAN uses Equation (1) to compute the probability of each class value y given the values of the features in the instance, and assign to the instance the class value with the highest probability.

$$P(y|x_1, x_2, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|Par(x_i), y) \qquad (1)$$

In Equation (1), the probability of a class value y given all feature values $x_1, ..., x_n$ of an instance is estimated by calculating the product of the prior probability of class value y times the probability of each feature value x_i given its parent feature(s) *Par(*x_i*)* and y.

As proposed in [16], in the original Gene Ontology-based Bayesian Network Augmented Naïve Bayes (GO–BAN), the edges connecting features in the BAN network are exactly the edges pre-defined in the GO DAG. That is, for each node (feature) x_i in the BAN network, the parents of that node in that network are the GO terms which are parents of the GO term corresponding to x_i in the GO DAG, plus the class variable (which is a parent of all nodes in a BAN).

2.3 Eager Learning and Lazy Learning

Eager learning methods build a classifier in the training phase, before observing any testing instance to be classified. Then, the built classifier is used for classifying all testing instances. In contrast, lazy learning methods build a classifier only in the testing phase, when a new testing instance to be classified is observed [1,14]. Hence, an individual classifier is built for each testing instance. Lazy learning methods are in general slower than eager learning methods, but lazy methods have the advantage of building a classifier specifically adapted to the feature values in each testing instance.

2.4 Flat and Hierarchical Feature Selection Methods

Conventional feature selection methods are “flat” in the sense that they ignore the hierarchical dependencies among the features. In this work, for the purpose of comparison with hierarchical feature selection methods, we use one well-known flat feature selection named Correlation-based Feature Selection (CFS), which is based on the eager learning approach and on the principle of selecting the feature subset whose features have strong correlations with the class variable but weak correlations among each other [9]. Hence, CFS tries to minimize feature redundancy by favouring the selection of a feature subset with low correlation among the selected features, but that correlation is measured across the entire training set – using a conventional eager learning approach. This kind of feature redundancy is different from the kind of hierarchical feature redundancy considered by hierarchical feature selection methods (see below), which refers to feature values in a specific testing instance being classified.

Unlike flat feature selection methods like CFS, hierarchical feature selection methods exploit hierarchical dependencies among feature values in each testing instance in order to remove hierarchical redundancy. In this work, hierarchical feature redundancy is defined as the case where two features are located in the same path from a root to a leaf node in the DAG and have the same value in the current testing instance being classified. For example, in Figure (1), feature B is redundant with respect to feature A, because feature A is the parent of B, and both of them have the value “1”.

There are few papers in the area of hierarchical feature selection, mainly focusing on tasks other than the classification task addressed in this work. For example, Alexa et al. [2] exploits the generalization-specialization relationship among GO terms to select “enriched” GO terms, i.e., terms that occur significantly more often than expected. In addition, other works have proposed hierarchical feature selection methods for the task of linear regression [11, 13, 18, 19], where the variable to be predicted is continuous. In the context of the classification task, our previous papers [16, 17] proposed three hierarchical feature selection methods that were used to select features for the Naïve Bayes classifier, namely: Selecting Hierarchical Information Preserved Features (HIP), Selecting Most Relevant Features (MR) and the Hybrid of HIP and MR (HIP–MR).

All these three methods are based on the lazy learning approach, and the first two methods have shown the best predictive performance. Hence, here we briefly discuss only HIP and MR. The HIP method selects only the *core* features whose values in the current instance logically imply the values of all other features in that instance, due to the “is-a” relationship explained earlier. For example, the *core* features for the example DAG shown in Figure (1) are D, E, F, I and J. In terms of feature D, its value “0” in the current instance logically implies that its descendant features G and H also have value “0” in that instance. Analogously, feature E’s value “1” implies that its ancestor features B and A also have value “1”.

Unlike HIP, MR considers not only the “is-a” relationship among features, but also a measure of the relevance (predictive power) of features. For each path from a root node to a leaf node in the feature DAG, MR divides the features in that path into two sets, the set of features with value “1” and the set of features with value “0”; and then it selects only the maximum relevance feature in each set. For the example DAG shown in Figure (1), MR selects features B, C, G, H, I and J. Feature B has the maximum relevance value among two sets of features with value “1”: (A, B, E) and (A, B, F). Analogously, features G and H have the maximum relevance value among the sets of features with value “0” – (D, G) and (D, H), respectively.

3. PROPOSED METHODS

In this paper, we evaluate the predictive performance of four different versions of the Gene Ontology-based Bayesian Network Augmented Naïve Bayes (GO–BAN) method, namely: (a) the original GO–BAN method proposed in [16], where the BAN network directly uses the GO DAG induced by the input features and no feature selection method is used; (b) GO–BAN where the BAN network is constructed based on the result of the hierarchical feature selection method HIP; (c) GO–BAN where the BAN network is constructed based on the result of the hierarchical feature selection method MR; (d) GO–BAN where the BAN network is constructed based on the result of the flat feature selection method CFS.

Note that in the approaches (b), (c), (d), the construction of the BAN network is not trivial, because the feature selection methods can select features that are hierarchically related (one is the ancestor or descendant of the other) by are not directly connected by an edge in the GO DAG. For instance, in Figure (1), a method could select features A and D. In such cases, if the BAN network contained only edges occurring in the GO DAG, there would be no edge connecting A and D in the BAN, suggesting these features are independent, which would be misleading, given their hierarchical dependency. Therefore, it is necessary to create artificial edges, not present in the GO DAG, which are nonetheless based on hierarchical dependencies represented in the GO DAG, so that these artificial edges can be used in the BAN network. Hence, we propose two methods for constructing the BAN network based on the features selected in a pre-processing phase and on the structure of the GO DAG. The first BAN network construction method was designed for the case where features have been selected by an eager feature selection method (CFS in this work, but other methods could be used). The second BAN network construction method was designed for the case where features have been selected by a lazy feature selection method (HIP and MR in this work).

3.1 Correlation-based Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (CFS+GO–BAN)

Correlation-based Feature Selection (CFS), as an eager method, selects a single subset of features for classifying all testing instances. To construct the feature network of the GO–BAN algorithm from the set of features selected by CFS in a pre-processing phase, we propose the method described in Algorithm (1).

In the first phase of Algorithm (1), in lines (1)–(3), the feature DAG, training dataset and testing dataset will be initialized. The initial feature DAG simply contains one node for each GO term (feature) in the dataset and all the edges in the GO DAG where both GO terms connected by the edge are used as features in the dataset. Next, in line (4), CFS conducts flat feature selection; then the set of selected features $\mathbb{X}_{CFS}$ will be used to re-create the training and testing datasets, in lines (5)–(6).

The second phase (lines (7)–(11)) of CFS+GO–BAN (Algorithm (1)) re-constructs the edges between selected features according to the pre-defined hierarchical relationships in the DAG created in line (1). In details, for each feature x_s selected by CFS, the algorithm considers all paths leading from a root node of the DAG to x_s. As shown in lines (8)–(10), for each of those paths, the algorithm finds the closest ancestor of x_s in that path that was also selected by CFS, denoted (Closest Selected Ancestor) CloSelAnc(x_s), and adds CloSelAnc(x_s) to the set of parents of x_s in the GO–BAN network; i.e., it adds an edge pointing from CloSelAnc(x_s) to x_s on the GO–BAN network. In the third and last phase of Algorithm (1), lines (12)–(14), each testing instance is classified using the previously constructed GO–BAN network.

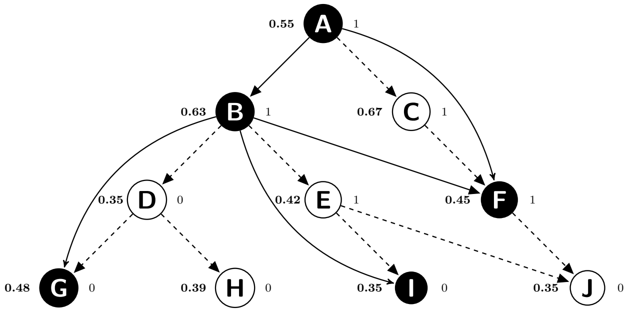

Figure 2: Nodes Selected by CFS and Corresponding Reconstructed Edges According to Gene Ontology Hierarchy (CFS+GO–BAN)

Algorithm 1 Correlation-based Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (CFS+GO–BAN)

```
 1: Initialize DAG with all features in Dataset;
 2: Initialize TrainSet;
 3: Initialize TestSet;
 4: $\mathbb{X}_{CFS}$ = CFS(TrainSet);
 5: Create TrainSet_CFS with features $\mathbb{X}_{CFS}$;
 6: Create TestSet_CFS with features $\mathbb{X}_{CFS}$;
 7: for each $x_s \in \mathbb{X}_{CFS}$ do
 8:     for each path $k$ in DAG from root to $x_s$ do
 9:         Par($x_s$) ← Par($x_s$) ∪ CloSelAnc($x_s$);
10:     end for
11: end for
12: for each Inst_CFS$_{<w>}$ ∈ TestSet_CFS do
13:     Classify(Par($\mathbb{X}_{CFS}$), TrainSet_CFS,
            Inst_CFS$_{<w>}$);
14: end for
```

To further explain how Algorithm (1) works, Figure (2) shows an example DAG where the selected nodes (features) are shown in black and the edges represent generalization-specialization relationships among GO terms (features) in the GO DAG. The dashed edges are the edges that are included in the GO DAG but are not included in the constructed GO–BAN network. The solid edges are the edges included in the constructed GO–BAN network; some of these solid edges represent parent-child relationships between selected features in the GO DAG, whilst other solid edges represent new edges which were artificially created to represent a directed connection between two selected features, some of them are separated by two or more edges in a given path of the GO DAG. Note that a selected node can have more than one selected ancestor nodes in an individual path, e.g., node G has two selected ancestors, B and A. In this case only its closest selected ancestor node (B) – in the path A-B-D-G – will be assigned to the set of parent nodes of G in lines (8)–(10) of Algorithm (1). Analogously, only the closest selected ancestor of node I in the path A-B-E-I, namely node B, will be added to the set of parents of node I. Furthermore, node F is assigned two parent nodes, namely B, which is F's closest selected ancestor in path A-B-F, and A, which is F's only selected ancestor in path A-C-F.

Note that, CFS being an eager and flat feature selection method, cannot guarantee the elimination of hierarchical redundancies (a concept discussed in Section 2.4) between features. Therefore, CFS can select features that have the same value (either "1" or "0") in an instance and are located in the same path in the GO DAG. In the example DAG in Figure (2), CFS has selected features A and B, which is a case of hierarchical redundancy (the value "1" of B in an instance implies the value "1" of A in that instance). Such hierarchical redundancies in the GO–BAN network are avoided by using hierarchical feature selection algorithms, as discussed in the next Section.

3.2 Hierarchical Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (HFS+GO–BAN)

Recall that the Hierarchical Feature Selection (HFS) methods used in this work perform lazy learning, i.e., they select a set of features specific for each testing instance. We evaluate the predictive performance of GO–BAN when using two lazy HFS methods in a pre-processing phase, i.e., HIP and MR (as reviewed on Section 2.4). Hence, in this Section we propose another method to construct the GO–BAN network from the set of features selected by HIP or MR. Note that the proposed method is generic enough to be used with any other lazy HFS method.

Algorithm (2) works in a way analogous to Algorithm (1). The core part of both algorithms consists of finding the closest selected ancestor of each selected feature x_s in each path of the GO DAG and adding that ancestor to the set of parents of feature x_s. The main difference between these two algorithms is as follows. Since Algorithm (1) uses an eager feature selection algorithm, its core part (the loop in lines (7)–(11)) is performed before processing the testing instances in lines (12)–(14). By contrast, since Algorithm (2) uses a lazy feature selection method, both the use of a HFS method in line (5) and the algorithm's core part (the loop in lines (8)–(12)) are performed within a loop over all testing instances. Another difference is that line (9) of Algorithm (1) involves finding the closest selected ancestor of selected feature x_s in path k; whilst the corresponding line (10) of Algorithm (2) is somewhat simpler; it is not necessary to select the closest ancestor of x_s among several ancestors, simply because x_s will have at most one selected ancestor feature. This is due to the fact that the HFS method executed in line (5) (i.e., HIP or MR) eliminates hierarchical redundancies among features.

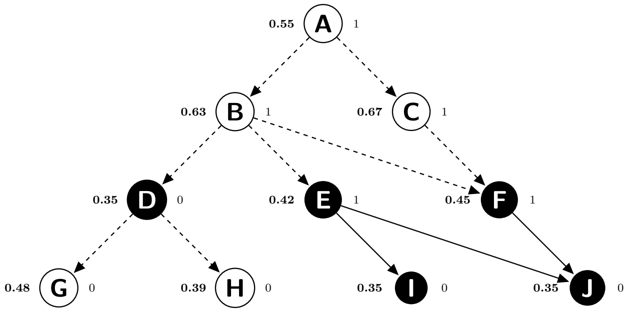

Figure 3: Nodes Selected by HIP and Corresponding Reconstructed Network According to Gene Ontology Hierarchy (HIP+GO–BAN)

The initialization phase of HFS+GO–BAN (lines (1)–(3) in Algorithm (2)) is the same as the initialization phase of Algorithm (1). Then, for each testing instance ($\mathbf{Inst}_{<w>}$), a lazy learning HFS method (either HIP or MR) will be run (line (5) in Algorithm (2)). Next, the set of hierarchically selected features $\mathbb{X}_{HFS}$ is used to re-create the new training dataset **TrainSet_HSF** and the current testing instance $\mathbf{Inst\_HSF}_{<w>}$. In lines (8)–(12), the GO–BAN network is constructed. For each selected feature x_s in $\mathbb{X}_{HSF}$, for each path in the DAG from a root node to x_s, the only selected ancestor of x_s (if such ancestor exists) is added to the set of parents of x_s in the GO–BAN network in line (10).

Algorithm 2 Hierarchical Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (HFS+GO–BAN)

```
1: Initialize DAG with all features in Dataset;
2: Initialize TrainSet;
3: Initialize TestSet;
4: for each $\mathbf{Inst}_{<w>} \in$ TestSet do
5:     $\mathbb{X}_{HFS} = \mathbf{HFS}(\mathbf{DAG}, \mathbf{TrainSet}, \mathbf{Inst}_{<w>})$;
6:     Create TrainSet_HFS with features $\mathbb{X}_{HFS}$;
7:     Create $\mathbf{Inst\_HFS}_{<w>}$ with features $\mathbb{X}_{HFS}$;
8:     for each $x_s \in \mathbb{X}_{HFS}$ do
9:         for each path $k$ in DAG from root to $x_s$ do
10:            $\mathbf{Par}(x_s) \leftarrow \mathbf{Par}(x_s) \cup \mathbf{SelAnc}(x_s)$;
11:        end for
12:    end for
13:    Classify($\mathbf{Par}(\mathbb{X}_{HFS})$, TrainSet_HFS,
         $\mathbf{Inst\_HFS}_{<w>}$);
14: end for
```

To further explain how Algorithm (2) works when HIP is used, consider the example DAG in Figure (3), where the nodes selected by HIP are marked in black (nodes D, E, I, F and J). Each of these nodes has at most one selected ancestor node in each path from the root to that node. Hence, Algorithm (2) assigns node E as the parent of node I in path A-B-E-I; node E as the parent of node J in path A-B-E-J; node F as the parent of node J in paths A-B-F-J and A-C-F-J. Node D is not assigned any parent, since none of its ancestor nodes in the DAG were selected by HIP.

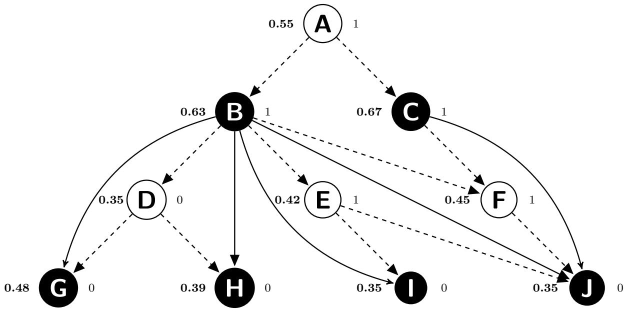

Figure 4: Nodes Selected by MR and Corresponding Reconstructed Network According to Gene Ontology Hierarchy (MR+GO–BAN)

To further explain how Algorithm (2) works when MR is used, consider the DAG in Figure (4), where again the selected nodes are marked in black (nodes B, G, H, C, I and J). Again, each selected node has at most one selected ancestor node in each path from the root to that node. Hence, Algorithm (2) assigns node B as the parent of node G in path A-B-D-G; node B as parent of node H in paths A-B-D-H and A-B-H; node B as parent of node I in path A-B-E-I; node B as parent of node J in paths A-B-E-J and A-B-F-J; node C as parent of node J in path A-C-F-J.

4. COMPUTATIONAL EXPERIMENTS

4.1 Aging-related Datasets

We create the datasets by integrating aging-related genes information about four model organisms, i.e., *Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus* and *Saccharomyces cerevisiae*, from the Human Ageing Genomic Resources (HAGR) GenAge database (Build 16) [3] and the Gene Ontology (GO) database (version: 2013-08-07) [15]. The detailed methods for creating the datasets are described in [16,17], where only biological process GO terms were used as predictive features. In this paper, the datasets have been extended, and consist of all three types of GO terms, i.e., biological process (BP), molecular function (MF), cellular component (CC), as well as different combinations of them, i.e., BP and MF terms, BP and CC terms, MF and CC terms, and finally using all BP, MF and CC terms as predictive features. More detailed characteristics about all the created datasets is shown in Table (1), where # **F**, # **E**, and # **I** denote the number of input features, edges in the GO DAG, and instances, respectively. Note that the root terms for the DAGs of BP (GO:0008150), MF (GO:0003674), and CC (GO:0005575) GO terms have been removed from the corresponding datasets, since they have no predictive power.

4.2 Experimental Methodology

There are 4 methods being compared, namely: GO–BAN without feature selection (as a baseline method), GO–BAN based on features selected by the HIP method, GO–BAN based on features selected by the MR method, and GO–BAN based on features selected by the CFS method. We used a well-known 10-fold cross validation procedure to evaluate the performance of classifiers as measured by their GMean value, which is calculated by the square root of the product of Sen. and Spe., i.e., $GMean = \sqrt{Sen. \times Spe.}$. Sen. denotes proportion of positive (pro-longevity) instances that are correctly classified as positive; while Spe. denotes the proportion of negative (anti-longevity) instances that are correctly classified as negative. The GMean measure is suitable to evaluate classifiers applied to datasets where there is a significantly imbalanced distribution of classes, which is the case in some of our datasets. In such imbalanced class datasets, maximizing GMean is challenging because there is a trade-off between Sen. and Spe.

Table 1: Characteristics about the Aging-related Datasets

***Caenorhabditis elegans* (Worm)**							
	BP	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
# **F**	830	218	143	1048	973	361	1191
# **E**	1437	259	217	1696	1654	476	1913
# **I**	528	279	254	553	557	432	572
***Drosophila melanogaster* (Fly)**							
	BP	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
# **F**	698	130	75	828	773	205	903
# **E**	1190	151	101	1341	1291	252	1442
# **I**	127	102	90	130	128	123	130
***Mus musculus* (Mouse)**							
	BP	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
# **F**	1039	182	117	1221	1156	299	1338
# **E**	1836	205	160	2041	1996	365	2201
# **I**	102	98	100	102	102	102	102
***Saccharomyces cerevisiae* (Yeast)**							
	BP	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
# **F**	679	175	107	854	786	282	961
# **E**	1223	209	168	1432	1391	377	1600
# **I**	215	157	147	222	234	226	238

4.3 Experimental Results

Table (2) compares the predictive performance of three feature selection methods working with GO–BAN and GO–BAN without feature selection. In general, HIP+GO–BAN shows the best performance among all 4 methods, being ranked as the best method 23 (out of 28) times (GMean values in boldface). In terms of the average ranks for those methods, HIP obtains the best rank of 1.2 on average over the 28 datasets, which is better than the average rank of MR(2.2), CFS(2.8) and no feature selection (3.8).

We performed a significance test on the predictive accuracies of different feature selection methods by adopting the Friedman test and Holm *post-hoc* method. The Friedman test is a non-parametric statistical test based on the ranks of each classifier's predictive performance on each dataset [5, 10], and the Holm *post-hoc* method is used for coping with the multiple-comparison problem that arises when applying significance tests to multiple pairwise method comparisons [4]. We used HIP+GO–BAN as the control (best) feature selection method to be compared with the other methods. The detailed results are shown in Table (3), where column 2 shows the average rank of each method (recall that the lower the rank, the better the predictive performance); column 3 shows the calculated p-value; column 4 shows the adjusted significance level (α). In column 3, a boldfaced value indicates that the p-value is lower than the corresponding adjusted significance level, which means the difference of GMean values between HIP+GO–BAN and the corresponding method is statistically significant. The outcomes of the significant tests show that HIP+GO–BAN significantly outperforms MR+GO–BAN, CFS+GO–BAN and GO–BAN without feature selection.

Table 2: Predictive Accuracy for GO–BAN with Hierarchical HIP, MR, and Flat CFS Method

Feature Types	**GO–BAN without Feature Selection**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
Caenorhabditis elegans (Worm) Datasets												
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	28.7 ± 2.2	86.5 ± 1.8	49.8	54.5 ± 3.2	73.4 ± 2.7	**63.2**	52.2 ± 3.1	74.0 ± 2.2	62.2	45.0 ± 2.6	80.9 ± 2.5	60.3
MF	34.7 ± 4.5	66.5 ± 4.5	**48.0**	43.8 ± 4.5	52.5 ± 5.2	**48.0**	35.5 ± 3.0	63.3 ± 3.4	47.4	31.4 ± 6.6	70.9 ± 6.0	47.2
CC	33.7 ± 4.5	81.4 ± 2.2	52.4	55.1 ± 5.0	63.5 ± 4.0	**59.2**	40.8 ± 4.3	73.1 ± 2.6	54.6	35.7 ± 4.3	74.4 ± 3.9	51.5
BP+MF	30.0 ± 2.7	84.7 ± 1.7	50.4	55.9 ± 3.2	74.1 ± 2.5	64.4	63.8 ± 2.2	73.2 ± 2.1	**68.3**	52.1 ± 3.7	77.6 ± 2.2	63.6
BP+CC	29.1 ± 2.1	86.6 ± 1.7	50.2	58.7 ± 3.6	72.7 ± 2.5	**65.3**	54.0 ± 2.8	74.7 ± 2.3	63.5	47.4 ± 2.7	79.1 ± 1.5	61.2
MF+CC	35.3 ± 2.9	80.2 ± 3.2	53.2	55.9 ± 3.1	64.5 ± 3.6	**60.0**	47.1 ± 3.4	70.2 ± 3.9	57.5	46.5 ± 4.1	72.1 ± 4.0	57.9
BP+MF+CC	31.2 ± 2.9	85.2 ± 1.5	51.6	58.1 ± 3.8	73.4 ± 2.6	**65.3**	55.3 ± 4.0	72.0 ± 2.6	63.1	50.7 ± 4.1	75.4 ± 2.1	61.8
Drosophila melanogaster (Fly) Datasets												
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	100.0 ± 0.0	0.0 ± 0.0	0.0	75.8 ± 4.4	52.8 ± 8.6	**63.3**	80.2 ± 3.5	44.4 ± 10.2	59.7	78.0 ± 4.1	25.0 ± 7.8	44.2
MF	91.2 ± 3.3	26.5 ± 3.4	49.2	64.7 ± 7.2	50.0 ± 10.0	56.9	80.9 ± 5.2	47.1 ± 9.1	**61.7**	85.3 ± 4.3	32.4 ± 7.1	52.6
CC	93.5 ± 2.6	28.6 ± 11.1	51.7	79.0 ± 6.6	46.4 ± 11.4	60.5	85.5 ± 4.6	42.9 ± 10.2	60.6	88.7 ± 3.5	46.4 ± 11.4	**64.2**
BP+MF	97.8 ± 1.5	0.0 ± 0.0	0.0	72.8 ± 3.9	63.2 ± 9.3	**67.8**	80.4 ± 3.7	44.7 ± 8.2	59.9	83.7 ± 3.5	28.9 ± 6.2	49.2
BP+CC	98.9 ± 1.1	0.0 ± 0.0	0.0	73.6 ± 4.7	62.2 ± 8.4	**67.7**	80.2 ± 4.1	51.4 ± 10.9	64.2	82.4 ± 4.4	40.5 ± 10.2	57.8
MF+CC	95.3 ± 1.9	31.6 ± 5.3	54.9	80.0 ± 6.2	60.5 ± 7.6	**69.6**	83.5 ± 4.9	55.3 ± 8.2	68.0	90.6 ± 3.0	52.6 ± 4.5	69.0
BP+MF+CC	98.9 ± 1.1	2.6 ± 2.5	16.0	73.9 ± 4.7	68.4 ± 5.3	71.1	81.5 ± 3.7	63.2 ± 7.7	**71.8**	88.0 ± 2.6	44.7 ± 8.2	62.7
Mus musculus (Mouse) Datasets												
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	98.5 ± 1.4	26.5 ± 5.0	51.1	75.0 ± 5.1	70.6 ± 5.1	**72.8**	88.2 ± 4.7	44.1 ± 7.7	62.4	85.3 ± 4.3	44.1 ± 5.9	61.3
MF	90.8 ± 3.3	27.3 ± 10.0	49.8	84.6 ± 3.0	45.5 ± 12.2	**62.0**	87.7 ± 3.0	39.4 ± 10.6	58.8	87.7 ± 2.9	30.3 ± 9.6	51.5
CC	86.4 ± 3.3	35.3 ± 11.2	55.2	80.3 ± 3.0	50.0 ± 10.1	**63.4**	78.8 ± 3.8	44.1 ± 11.1	58.9	78.8 ± 3.3	38.2 ± 12.6	54.9
BP+MF	98.5 ± 1.4	29.4 ± 6.4	53.8	69.1 ± 5.8	70.6 ± 8.1	**69.8**	86.8 ± 4.0	41.2 ± 9.6	59.8	89.7 ± 2.2	41.2 ± 8.0	60.8
BP+CC	98.5 ± 1.4	29.4 ± 6.4	53.8	66.2 ± 6.0	76.5 ± 8.0	**71.2**	77.9 ± 5.3	52.9 ± 9.6	64.2	82.4 ± 5.6	47.1 ± 11.7	62.3
MF+CC	91.2 ± 3.2	26.5 ± 8.8	49.2	79.4 ± 4.2	61.8 ± 12.5	70.0	83.8 ± 5.0	58.8 ± 13.1	**70.2**	79.4 ± 4.8	44.1 ± 9.6	59.2
BP+MF+CC	98.5 ± 1.4	26.5 ± 10.5	51.1	70.6 ± 6.0	76.5 ± 8.8	**73.5**	86.8 ± 4.0	50.0 ± 6.9	65.9	83.8 ± 3.3	52.9 ± 8.4	66.6
Saccharomyces cerevisiae (Yeast) Datasets												
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	0.0 ± 0.0	100.0 ± 0.0	0.0	63.3 ± 6.0	76.8 ± 3.1	**69.7**	33.3 ± 8.6	89.7 ± 2.5	54.7	20.0 ± 5.4	94.6 ± 1.9	43.5
MF	0.0 ± 0.0	99.2 ± 0.8	0.0	23.1 ± 6.7	80.2 ± 3.9	**43.0**	0.0 ± 0.0	90.8 ± 3.0	0.0	0.0 ± 0.0	94.7 ± 1.6	0.0
CC	12.5 ± 6.1	99.2 ± 0.8	35.2	29.2 ± 10.2	83.7 ± 4.1	**49.4**	20.8 ± 6.9	93.5 ± 2.7	44.1	20.8 ± 7.5	93.5 ± 1.6	44.1
BP+MF	0.0 ± 0.0	100.0 ± 0.0	0.0	73.3 ± 6.7	71.9 ± 3.0	**72.6**	23.3 ± 7.1	89.6 ± 2.6	45.7	26.7 ± 8.3	96.4 ± 1.1	50.7
BP+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	63.3 ± 10.5	78.4 ± 2.9	**70.4**	40.0 ± 8.3	87.3 ± 2.5	59.1	26.7 ± 6.7	96.6 ± 1.1	50.8
MF+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	41.4 ± 8.3	80.7 ± 3.0	**57.8**	13.8 ± 6.3	88.8 ± 2.3	35.0	13.8 ± 6.3	93.4 ± 1.5	35.9
BP+MF+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	76.7 ± 7.1	73.6 ± 2.8	**75.1**	33.3 ± 5.0	87.0 ± 2.5	53.8	23.3 ± 8.7	94.2 ± 1.6	46.8

Table 3: Statistical Test Results of the Algorithms' GMean Values According to the Non-parametric Friedman Test with the Holm *Post-hoc* Test at the $\alpha = 0.05$ Significance Level

Algorithms	*Ave. Rank*	*P-value*	*Adjusted* α
HIP+GO–BAN (ctrl)	1.2	–	–
MR+GO–BAN	2.2	**3.74 E-03**	0.050
CFS+GO–BAN	2.8	**3.52 E-06**	0.025
No FS+GO–BAN	3.8	**4.85 E-14**	0.017

5. DISCUSSION

Table (4) reports a number of statistics about the size of the constructed GO–BAN's DAGs, when using different feature selection methods. More precisely, the columns referring to GO–BAN without feature selection report the original number of features (**F**) and edges (**E**) in the feature DAG for each dataset, and the average dimensionality of a conditional probability table (CPT) in that DAG, denoted $\mathbf{D}(CPT)$. To calculate $\mathbf{D}(CPT)$, note that each node is associated with a number of variables given by its number of parent feature nodes plus two – accounting for one class variable (which is a parent of all feature nodes) and the feature represented by the node itself. Since all (feature and class) variables can take two values, the dimensionality of each CPT is given by Equation (2), where $\#Par$ is the number of parent features. The table columns referring to GO–BAN using HIP and MR as feature selection methods report the average number of selected features (**AvF**), the average number of edges in the constructed DAG (**AvE**), and the average CPT dimensionality in the DAG for the corresponding feature selection method, where each average is computed over the DAGs constructed for all testing instances (since HIP and MR select a specific feature set for each testing instance) across all 10 cross-validation interactions. Finally, in the table columns referring to GO–BAN using the feature selection method CFS, the average is computed over the 10 cross-validation iterations only, since in each iteration CFS selects the same set of features to classify all available testing instances.

$$\mathbf{D}(CPT) = 2^{(\#Par+2)} \tag{2}$$

In general, the three feature selection methods selected substantially fewer features and so constructed GO–BAN DAGs with substantially fewer edges, compared with the original DAGs (without performing feature selection). More precisely, among the three feature selection methods, CFS selected the smallest number of features in 27 out of the 28 datasets (the only exception is the dataset for *S. cerevisiae* with MF features). MR selected the largest number of features in all 28 datasets; and the number of features selected by HIP is in general an intermediate value between the numbers selected by the other two methods. However, HIP+GO–BAN constructed DAGs having in general fewer edges than the DAGs constructed by MR+GO–BAN and CFS+GO–BAN. Figure (5) shows the average CPT dimensionality ($\mathbf{D}(CPT)$) in the DAGs constructed by each method, where the average was computed over all the 28 datasets. As shown in this figure, despite CFS selecting a smaller feature set than HIP and MR, the CFS+GO–BAN method constructs DAGs with the largest average CPT dimensionality ($\mathbf{D}(CPT)$) value of 5.65, among the three feature selection methods – although this value is still much smaller than the value for GO–BAN without feature selection (14.6). This $\mathbf{D}(CPT)$ value of 5.65 for CFS+GO–BAN is substantially higher than the $\mathbf{D}(CPT)$ values obtained by MR+GO–BAN (4.78) and by HIP+GO–BAN (4.26). This indicates that, although CFS selected the smallest number of features, on average the features selected by CFS have a higher number of parent nodes in the constructed DAGs, leading to the highest $\mathbf{D}(CPT)$ values for CFS among feature selection methods.

These results are consistent with the discussion in Section 3.1, i.e., the features selected by CFS can have more than one ancestor features that have the same values and are also located in the same path in the DAG, constituting a case of hierarchical redundancy (defined in Section 2.4), a type of redundancy that is not eliminated by CFS; and this leads to a higher number of parents per node and so a substantially higher $\mathbf{D}(CPT)$ value for CFS.

Unlike CFS, both HIP and MR remove the hierarchical redundancy between features, which means there will exist at most two nodes being selected and at most one dependency being constructed for individual path; and this leads to substantially lower $\mathbf{D}(CPT)$ values for HIP+GO–BAN and MR+GO–BAN, by comparison with CFS+GO–BAN. The reason for HIP+GO–BAN having a smaller $\mathbf{D}(CPT)$ value than MR+GO–BAN is that HIP selected in general substantially fewer features than MR (as shown in Table 4), which led to substantially smaller numbers of edges and parent features per node. In particular, the lowest $\mathbf{D}(CPT)$ value of 4.26 obtained by HIP+GO–BAN suggests that most nodes in the constructed DAG have no parent feature (like the case of the Naïve Bayes classifier, where each feature is independent to all other non-class features), since a $\mathbf{D}(CPT)$ value of 4 means a CPT has only four probability values, arising from the four combinations of two values of the current feature and two values of the class variable. The small size of the CPTs constructed by HIP+GO–BAN suggests that this method is the one that most mitigates the problem of over-fitting associated with large CPTs; because the larger the average dimensionality of CPTs in a constructed DAG, the larger the number of "parameters" (probability values) to be estimated from the training data, and the larger the risk of over-fitting.

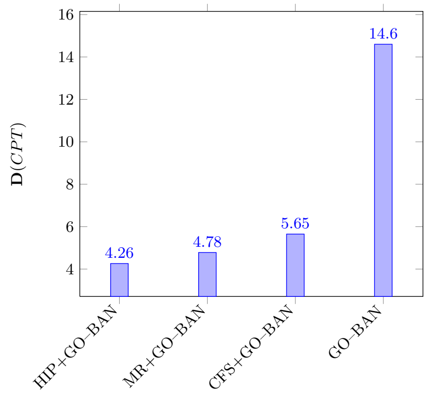

Figure 5: Average $\mathbf{D}(CPT)$ Values for Different Feature Selection Methods Working with GO–BAN over 28 Datasets

Table 4: Information about Number of Features, Edges and Dimensionalities of CPT Tables for the Constructed GO–BAN Classifier

Feature Types	**GO–BAN without FS**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
Caenorhabditis elegans (Worm) Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	830	1437	17.66	69.27	2.19	4.13	145.67	32.21	4.95	42.1	8.5	4.83
MF	218	259	10.32	29.81	2.91	4.40	50.52	9.02	4.73	27.8	6.3	4.91
CC	143	217	14.03	29.73	2.09	4.31	54.98	7.84	4.61	23.3	2.9	4.50
BP+MF	1049	1696	16.13	91.88	4.43	4.20	195.41	31.89	4.69	54.4	10.0	4.74
BP+CC	974	1654	17.12	90.01	3.11	4.15	189.84	32.43	4.73	53.9	11.4	4.88
MF+CC	362	476	11.79	51.85	3.89	4.31	102.00	14.57	4.60	40.0	7.5	4.75
BP+MF+CC	1193	1913	15.88	112.96	5.33	4.19	244.66	38.32	4.66	60.9	10.8	4.72
Drosophila melanogaster (Fly) Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	698	1190	17.28	82.53	3.94	4.21	141.74	19.83	4.66	31.2	5.4	4.77
MF	130	151	10.29	22.87	2.65	4.49	31.76	5.99	4.80	13.3	2.7	4.81
CC	75	101	12.05	20.73	1.58	4.31	27.60	8.39	5.33	14.6	4.6	5.37
BP+MF	829	1341	16.17	120.99	6.39	4.26	172.68	27.38	4.73	31.8	6.4	4.93
BP+CC	774	1291	16.76	100.38	5.02	4.21	167.14	29.84	4.83	33.5	6.6	4.84
MF+CC	206	252	10.94	40.65	3.77	4.38	58.59	10.07	4.73	21.3	5.5	5.07
BP+MF+CC	905	1442	15.83	121.34	7.48	4.22	201.47	31.71	4.97	33.6	7.9	5.08
Mus musculus (Mouse) Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	1039	1836	17.18	128.60	7.48	4.25	197.48	28.37	4.64	36.6	6.5	4.79
MF	182	205	9.68	44.06	4.39	4.41	50.37	10.95	4.92	25.3	8.5	5.47
CC	117	160	12.37	36.68	2.87	4.33	38.75	11.85	5.50	15.7	2.4	4.64
BP+MF	1222	2041	16.06	171.32	11.70	4.29	245.42	38.58	4.69	43.7	10.2	5.04
BP+CC	1157	1996	16.69	164.83	10.29	4.27	234.87	40.58	4.77	40.2	8.4	4.94
MF+CC	300	365	10.74	78.96	7.03	4.37	90.04	19.76	4.99	27.5	7.8	5.24
BP+MF+CC	1340	2201	15.73	207.56	14.51	4.29	286.44	49.50	4.77	46.3	8.9	4.84
Saccharomyces cerevisiae (Yeast) Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	679	1223	18.85	54.58	1.97	4.15	107.24	13.51	4.55	31.4	19.0	7.68
MF	175	209	10.43	24.59	1.78	4.30	40.98	5.90	4.58	35.6	8.4	4.96
CC	107	168	14.56	28.56	1.14	4.16	35.34	9.51	5.15	20.7	18.0	7.98
BP+MF	855	1432	17.12	76.54	3.41	4.19	150.73	17.36	4.51	31.1	18.6	7.69
BP+CC	787	1391	18.26	77.91	2.63	4.14	144.09	21.46	4.65	34.5	33.3	10.57
MF+CC	283	377	12.00	48.11	2.28	4.19	84.59	11.81	4.58	29.8	18.3	7.07
BP+MF+CC	963	1600	16.83	99.96	4.03	4.17	191.24	25.35	4.57	34.9	28.4	9.21

6. CONCLUSION

We proposed two methods for constructing a feature DAG (network) to be used by a Bayesian Network Augmented Naïve Bayes (GO–BAN) classifier, in datasets of aging-related genes where Gene Ontology (GO) terms are used as hierarchically related predictive features. One BAN network construction method relies on a hierarchical feature selection method to detect and remove hierarchical redundancies among features (GO terms); whilst the other BAN network construction method simply uses a conventional, "flat" feature selection method to select features, without removing the hierarchical redundancies associated with the GO. Both BAN network construction methods may create new edges among nodes (features) in the BAN network that did not exist in the original GO DAG, in order to preserve the generalization-specialization (ancestor-descendant) relationship among selected features. Our experimental results showed that the first BAN network construction method, when using either HIP or MR as a hierarchical feature selection method, obtained in general higher predictive accuracies across the 28 aging-related datasets than the second BAN network construction method, when the latter used CFS as a flat feature selection method. The experiments also indicated that the BAN network construction method using HIP obtained a statistically significantly better predictive accuracy than the accuracy obtained by the BAN network construction method using MR, the BAN network construction method using CFS, and the baseline approach of using a BAN network directly given by the GO DAG without performing feature selection. An advantage of HIP, in this context, is that it selected substantially fewer features than MR or CFS, which led to substantially smaller conditional probability tables in the BAN network. As a result, the BAN networks constructed based on the features selected by HIP were less prone to over-fitting, since they had fewer parameters (probability values) to be estimated from the training dataset than the BAN networks constructed based on the features selected by MR or CFS. Future research could involve more experiments with the proposed BAN network construction methods, using other types of hierarchical and flat feature selection methods and using other types of classification datasets with hierarchical features.

7. ACKNOWLEDGEMENTS

We thank Dr. João Pedro de Magalhães for his valuable general advice for this project.

8. REFERENCES

[1] D. W. Aha. *Lazy Learning.* Kluwer Academic Publishers, Norwell, MA, 1997.

[2] A. Alexa, J. Rahnenführer, and T. Lengauer. Improved scoring of functional groups from gene expression data by decorrelating go graph structure. *Bioinformatics*, 22(13):1600–1607, Apr. 2006.

[3] J. P. de Magalhães, A. Budovsky, G. Lehmann, J. Costa, Y. Li, V. Fraifeld, and G. M. Church. The human ageing genomic resources: online databases and tools for biogerontologists. *Aging Cell*, 8(1):65–72, Feb. 2009.

[4] J. Demsăr. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, Jan. 2006.

[5] J. Derrac, S. Garcia, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, Mar. 2011.

[6] Y. Fang, X. Wang, E. K. Michaelis, and J. Fang. Classifying aging genes into DNA repair or non-DNA repair-related categories. In D. S. Huang, K. H. Jo, Y. Q. Zhou, and K. Han, editors, *Lecture Notes in Intelligent Computing Theories and Technology*, pages 20–29. Springer, Berlin Heidelberg, 2013.

[7] A. A. Freitas, O. Vasieva, and J. P. de Magalhães. A data mining approach for classifying DNA repair genes into ageing-related or non-ageing-related. *BMC Genomics*, 12(27):1–11, Jan. 2011.

[8] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, Nov. 1997.

[9] M. A. Hall. *Correlation-based Feature Subset Selection for Machine Learning.* PhD thesis, University of Waikato, Hamilton, New Zealand, 1998.

[10] N. Japkowicz and M. Shah. *Evaluating learning algorithms: a classification perspective.* Cambridge University Press, New York, USA, 2011.

[11] R. Jenatton, J. Y. Audibert, and F. Bach. Structured variable selection with sparity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

[12] H. Liu and H. Motoda. *Feature selection for knowledge discovery and data mining.* Kluwer Academic Publishers, Norwell, MA, 1998.

[13] A. F. T. Martins, N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo. Structured sparsity in structured prediction. In *Proc. the 2011 conference on empirical methods in natural language processing (EMNLP 2011)*, pages 1500–1511, Edinburgh, UK, July 2011.

[14] R. B. Pereira, A. Plastino, B. Zadrozny, L. H. de C. Merschmann, and A. A. Freitas. Lazy attribute selection: Choosing attributes at classification time. *Intelligent Data Analysis*, 15(5):715–732, Aug. 2011.

[15] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.

[16] C. Wan and A. A. Freitas. Prediction of the pro-longevity or anti-longevity effect of *Caenorhabditis Elegans* genes based on Bayesian classification methods. In *Proc. IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)*, pages 373–380, Shanghai, China, Dec. 2013.

[17] C. Wan, A. A. Freitas, and J. P. de Magalhães. Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(2):262–275, Mar. 2015.

[18] J. Ye and J. Liu. Sparse methods for biomedical data. *ACM SIGKDD Explorations Newsletter*, 14(1):4–15, June 2012.

[19] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annual of Statistics*, 37(6):3468–3497, 2009.