Hierarchical Dependancy Constrained Averaged One-Dependence Estimators Classifiers for Hierarchical Feature Spaces

Cen Wan C.WAN@DCS.BBK.AC.UK
Department of Computer Science and Information Systems, Birkbeck, University of London

Alex A. Freitas A.A.FREITAS@KENT.AC.UK
School of Computing, University of Kent

Abstract

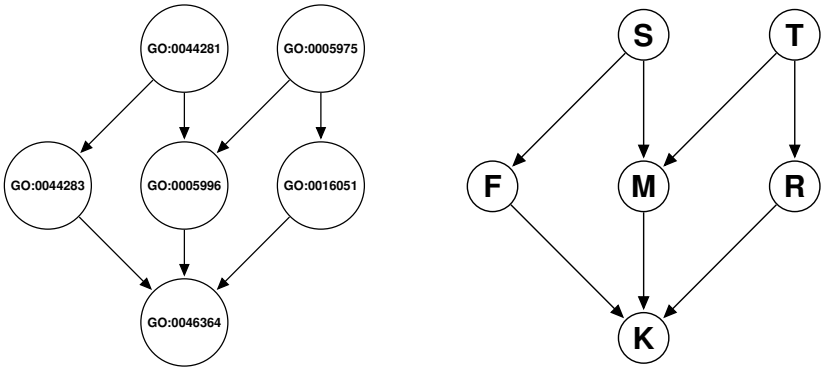
The Averaged One-Dependence Estimators classifier is a type of probabilistic graphical model that constructs an ensemble of one-dependency networks, using each feature in turn as a parent node for all other features, in order to estimate the distribution of the data. In this work, we propose two new types of Hierarchical dependency constrained Averaged One-Dependence Estimators (Hie-AODE) algorithms, which consider the pre-defined parent-child relationship between features during the construction of individual one-dependence estimators, when coping with hierarchically structured features. Experiments with 28 real-world bioinformatics datasets showed that the proposed Hie-AODE methods obtained better predictive performance than the conventional AODE classifier, and enhanced the robustness against imbalanced class distributions.

Keywords: Hierarchical feature spaces, Averaged One-Dependence Estimators, Gene Ontology.

1. Introduction

This work addresses the classification task of machine learning. We propose two new types of Averaged One-Dependence Estimators (AODE) classifier, namely Hie-AODE and Hie-AODE-Lite, which consider the pre-defined hierarchical dependency within a tree or a directed acyclic graph (DAG) of features as a type of hierarchical constraint during the classifier's training stage. In this work the features are Gene Ontology (GO) terms, which are structured as a DAG by using a type of "is-a" relationship to represent gene properties at different levels of description – from very generic properties (closer to a root of the DAG) to very specific properties (closer to a leaf node in the DAG). Those genes are classified as pro-longevity or anti-longevity. However, the proposed algorithms can also be applied to any types of hierarchically structured features. For instance, in text mining or document classification problems, the features represent the presence or absence of words in a document, since words are naturally organised into generalisation-specialisation hierarchies that are readily available for machine learning (e.g. the WordNet (Miller, 1995) hierarchy).

The pre-defined hierarchical dependency information (i.e. the ancestor-descendant relationships) is informative and can be exploited for different machine learning tasks. Several authors proposed a series of feature selection methods (Jenatton et al., 2011a; Mairal and Yu, 2013; Ristoski and Paulheim, 2014; Wan et al., 2015; Wan and Freitas, 2016, 2017; Zhao et al., 2016; da Silva et al., 2018) that exploit the hierarchical dependency information to reduce the dataset's dimensionality and obtained in general better predictive performance than other feature selection methods that don't exploit the hierarchical dependency information. In addition, the pre-defined hierarchical dependency information was also exploited as a type of constraint for training a regression model (Mairal et al., 2010; Jenatton et al., 2011b), and for constructing a Bayesian graphical model (Wan



(a) (b)

Figure 1: Examplar Gene Ontology and a symbolic representation.

and Freitas, 2013; Wan and Freitas, 2015). However, none of the above studies has exploited the hierarchical dependency information to propose new AODE methods. This is an interesting direction because AODE methods have the power of being an ensemble of classifiers, whilst being also computationally efficient (Webb et al., 2005).

This paper is organised as follows. Section 2 briefly reviews the background about Gene Ontology and the conventional AODE classifier. Section 3 proposes two new hierarchical dependency constrained AODE methods, viz. Hie-AODE and Hie-AODE-Lite. Section 4 presents the experimental methodology and results, and Section 5 presents conclusions and future research directions.

2. Background

2.1 The Hierarchical Structure of the Gene Ontology

The Gene Ontology (GO) (The Gene Ontology Consortium, 2000) is a popular bioinformatics database that provides unified vocabularies (i.e. Gene Ontology terms) to describe the functions of gene and gene products. Gene Ontology terms are categorised into three domains: Biological Process (BP), Molecular Function (MF) and Cellular Component (CC). Within each domain, GO terms are structured as a hierarchy by using an "is_a" relationship (a type of generalisation-specialisation relationship), which defines the parent-child and ancestor-descendant relationships between GO terms. As shown in Figure 1, those 6 example GO terms are hierarchically structured, e.g. GO:0044281 is the parent of GO:0044283 and GO:0005996, which are also the parent of GO:0046364. This type of hierarchy therefore is a Directed Acyclic Graph (DAG), as shown in Figure 2, where to simplify the notation, those 6 GO terms are represented by 6 arbitrary letters: S, T, F, M, R and K. In this work, we use GO terms as predictive features to describe genes, leading to a sparse matrix including binary values of features. For each single gene, the GO term's value *1* denotes that GO term is used to annotate that gene; whilst the value *0* denotes that gene is not annotated with that GO term.

2.2 Conventional Averaged One-Dependence Estimators (AODE) Algorithm

Averaged One-Dependence Estimators (AODE) (Webb et al., 2005) is a type of semi-naïve Bayes classification algorithm. Unlike the well-known *0-dependence* naïve Bayes, AODE is considered as

an *1-dependence* classifier. It builds an one-dependence estimator for each feature in turn by using that feature as a parent of all other features in the graph representing dependencies between features. Moreover, the class attribute is a parent of all features, in all one-dependency estimators. Then, it calculates the normalised posterior probability over the summation of the product of all features' one-dependence estimators to make the final classification. More precisely, AODE predicts for each new instance the most likely class as computed by Equation (1).

$$\underset{y}{\operatorname{argmax}} \left(\sum_{i=1}^{n} P(y, x_i) \prod_{j=1}^{n} P(x_j | y, x_i) \right) \tag{1}$$

$where\ x_i \in \mathbb{X}\ (full\ feature\ set),\ x_j \in \mathbb{X} - x_i\ and\ y\ is\ a\ class\ label;$

The conventional strategy of building one-dependence estimators for individual features and aggregating their predictions improves the generalisation capacity of the trained classifier, which often shows better predictive performance than other types of semi-naïve Bayes classification algorithms. However, the learning principle of AODE also has the drawback that some parent-child dependencies of the conventional one-dependence estimator violate the ancestor-descendant relationships that are pre-defined by the given feature hierarchy (in this work, the GO hierarchy). Figures 3.a, 3.d, 3.g, 3.j, 3.m, and 3.p show the topology of each one-dependence estimator built for each feature. When building the one-dependence estimator for feature M, for example, all other features are considered dependent on M and the class attribute. Analogously, when using each of the other features (R, F, K, S, or T) as a parent to build the one-dependence estimators, all other features are considered as the children of the corresponding parent feature and of the class attribute. This learning principle leads to the issue of ignoring the pre-defined parent-child dependencies between features. For example, when building the one-dependence estimator for feature M, all other features are considered as M's children, but features S and T are actually parents of M in the hierarchy.

3. Proposed Methods

In this work, we propose two novel Hierarchical dependency constrained Averaged One-Dependence Estimators algorithms, namely Hie-AODE and Hie-AODE-Lite, which exploit the pre-defined hierarchical dependency information between features to constrain the parent-child relationships during the one-dependency estimator learning stage.

In general, Hie-AODE constructs the one-dependence estimator for each parent feature x by using all other features in the training set except those which are ancestors of x in the feature hierarchy. This principle guarantees to avoid the violation of parent-child relationships defined by the existing feature hierarchy. Figures 3.b, 3.e, 3.h, 3.k, 3.n, and 3.q show an example topology of one-dependence estimators built by Hie-AODE. When using e.g. the feature M as the parent feature in an one-dependence estimator, all other features except S and T are assigned as the child features of M, due to the fact that S and T are parents of M in the given feature hierarchy, as shown in Figure 3.b. In addition, note that, when building an estimator for a feature which is a leaf of the whole feature hierarchy, the one-dependency estimator built by Hie-AODE is equivalent to the full independence estimator (a.k.a. Naïve Bayes). As shown by Figure 3.k, all other features are independent to K, since they are all parent or ancestor features of K. Conversely, when building an estimator for a feature which is a root of the whole hierarchy, the one-dependency estimator built by Hie-AODE is equivalent to the one built by the conventional AODE. As shown in Figure 3.n, when using e.g. feature S as a parent to build the one-dependence estimator, all other features are dependent on S, because feature S is not the child or descendant of any feature.

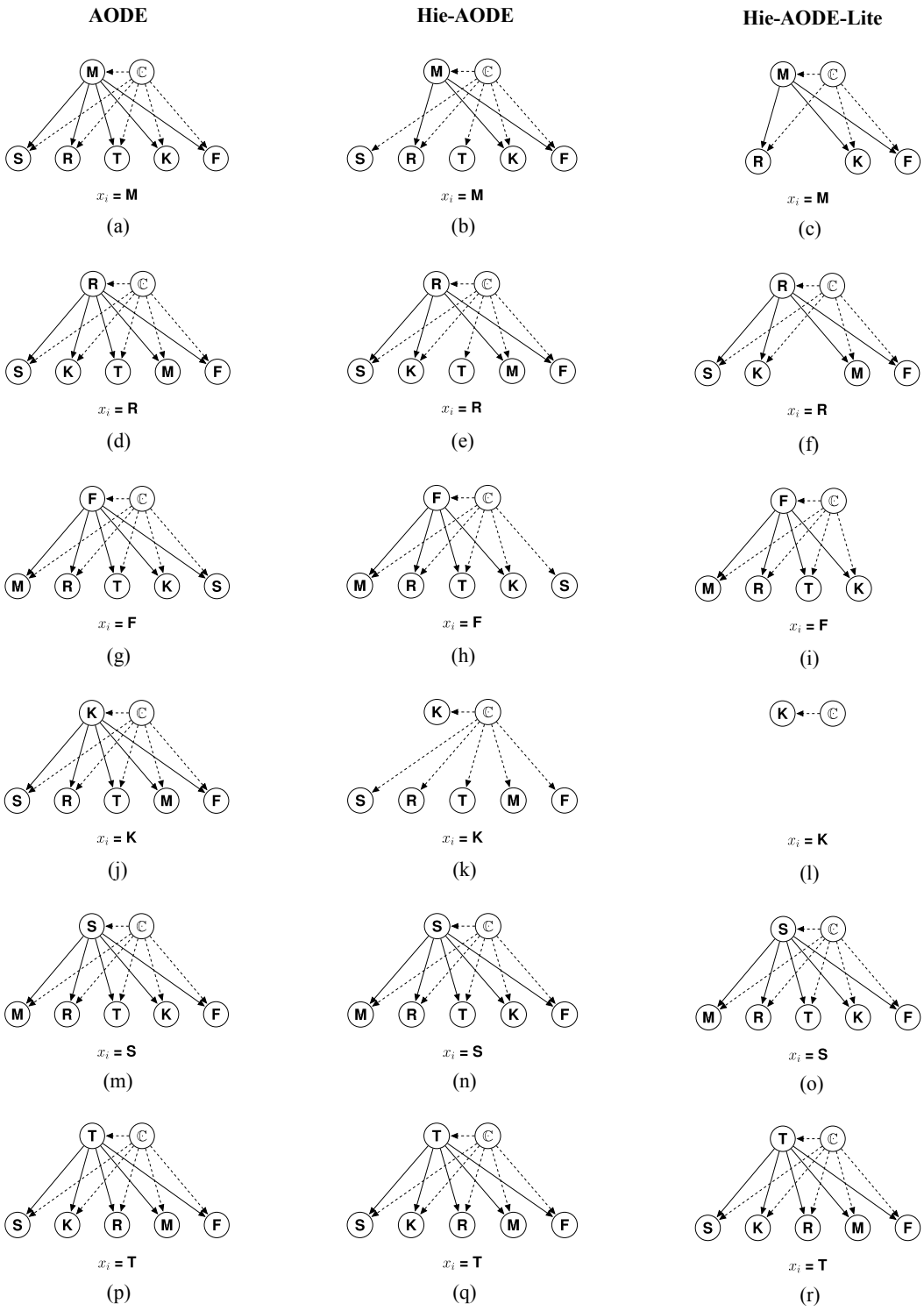


Figure 2: Example of feature dependencies represented by AODE, Hie-AODE and Hie-AODE-Lite for the feature DAG in Figure 2. The dashed edges show that all features are dependent on the class attribute. The solid edges represent dependencies between features.

Algorithm 1 Hie-AODE and Hie-AODE-Lite.

1: Initialise $\mathbf{DAG}$ with all features in Dataset;
2: Initialise $\mathbf{TrainSet}$;
3: Initialise $\mathbf{TestSet}$;
4: **for** each $x_i \in \mathbb{X}$ **do**
5: | Initialise $\mathbb{A}(x_i)$ given the $\mathbf{DAG}$;
6: | **for** each $x_j \in \mathbb{X}$ & $x_j \neq x_i$ & $x_j \notin \mathbb{A}(x_i)$ **do**
7: | | Create $\mathbb{CPT}(x_j, x_i, \mathbf{TrainSet})$
8: | **end for**
9: **end for**
10: **for** each $\mathbf{Inst} \in \mathbf{TestSet}$ **do**
11: | Classify($\mathbf{Inst}$, $\mathbb{X}$, $\mathbb{CPT}$);
12: **end for**

$$\underset{y}{\operatorname{argmax}} \left(\sum_{i=1}^{n} P(y, x_i) \prod_{j'=1}^{n'} P(x_{j'}|y, x_i) \prod_{k=1}^{a} P(x_k|y) \right) \tag{2}$$

$$\underset{y}{\operatorname{argmax}} \left(\sum_{i=1}^{n} P(y, x_i) \prod_{j'=1}^{n'} P(x_{j'}|y, x_i) \right) \tag{3}$$

where $x_{j'} \in \mathbb{X} - \mathbb{A}(x_i) - x_i$, *and* $x_k \in \mathbb{A}(x_i)$;

Hie-AODE-Lite constructs the one-dependence estimator for each feature in a similar way to Hie-AODE. However, the former directly removes the independent features which only depend on the class attribute from the AODE graph structure. This strategy reduces the number of parameters that are estimated by some individual one-dependence estimators during the training stage of the conventional AODE. Figures 3.c, 3.f, 3.i, 3.l, 3.o and 3.r show the topology of feature dependencies considered by Hie-AODE-Lite. For example, when building the one-dependence estimator for feature M (Figure 3.c), features S and T are removed from the feature set, due to the fact that those two features are parents of feature M in the feature hierarchy (Figure 2). Analogously, when building the one-dependence estimator for feature R, feature T is removed from the feature set.

The high-level pseudocode of Hie-AODE and Hie-AODE-Lite is shown in Algorithm 1, where lines 1 – 3 initialise the feature hierarchy (i.e. the $\mathbf{DAG}$), the training and testing datasets. In lines 4 – 9, during the training stage of Hie-AODE and Hie-AODE-Lite, each feature x_i is used as a parent to build an one-dependence estimator. In order to consider the pre-defined hierarchical dependency constraint (i.e. ancestor-descendant relationships between features), line 5 initialises the set of ancestors for each parent feature x_i (denoted by $\mathbb{A}(x_i)$) according to the feature hierarchy $\mathbf{DAG}$. Then lines 6 – 8 create the conditional probability tables (CPT) for all other non-ancestor features in the feature set $\mathbb{X}$, given their dependency on x_i and the class attribute. Note that Hie-AODE also creates the CPTs for feature x_i's ancestor features that depend only on the class attribute, unlike Hie-AODE-Lite. Hence, although Hie-AODE and Hie-AODE-Lite share the same pseudocode of Algorithm 1 at a high level of abstraction, they compute different CPTs at line 7. As a result of their difference, during the prediction stage, in lines 10 – 12, Hie-AODE and Hie-AODE-Lite adopt

Equations 2 and 3 respectively to calculate the posterior probability given the testing instance's values of feature set $\mathbb{X}$ and the calculated CPTs. In Equation 2, for each Hie-AODE one-dependence estimator, the product of the conditional probabilities of all non-ancestor features that are dependent on the parent feature x_i and the class attribute y (i.e. $\prod_{j'=1}^{n'} P(x_{j'}|y, x_i)$, where n' denotes the number of non-ancestor features of x_i) is multiplied by the product of the probabilities of all ancestor features that are only dependent on the class attribute (i.e. $\prod_{k=1}^{a} P(x_k|y)$, where a denotes the number of ancestor features of x_i). However, in Equation 3, the posterior probability of each Hie-AODE-Lite one-dependence estimator only includes the conditional probability of all non-ancestor features that are dependent on the parent feature and the class attribute.

4. Computational Experiments

4.1 Experimental Methodology

We evaluate the predictive performance of the proposed hierarchical dependency constrained AODE algorithms on 28 ageing-related gene datasets (Wan and Freitas, 2017). Those datasets comprise from 102 to 572 genes from 4 different model organisms, where each organism is associated with 7 different datasets having different sets of features, as shown in Table 1. In this table, #F, #E and #I denote, respectively, the number of features (GO terms), the number of edges in the feature hierarchy, and the number of instances (genes) in each dataset. Each gene is assigned a class label of pro-longevity or anti-longevity, and is described by different Gene Ontology terms (predictive features) from different domains, i.e. Biological Process (BP), Molecular Function (MF) and Cellular Component (CC), and their different types of combinations, i.e. BP+MF, BP+CC, MF+CC, and BP+MF+CC.

We compare the proposed Hie-AODE and Hie-AODE-Lite with the conventional AODE algorithm that does not consider the pre-defined hierarchical dependency constraints. In addition, another type of hierarchical dependency constrained regression method – proximalGraph (Mairal et al., 2010), and the Gene Ontology hierarchy-based Bayesian network augmented naïve Bayes (GO-BAN) method (Wan and Freitas, 2015) are also compared. The sensitivity and specificity values are calculated to evaluate the performance on predicting the positively and negatively labeled instances respectively.

The sensitivity is the proportion of positive (pro-longevity) instances correctly predicted as positive, while the specificity is the proportion of negative (anti-longevity) instances correctly predicted as negative. The Geometric Mean (GMean) of sensitivity and specificity (the square root of the product of sensitivity and specificity values) is also adopted to evaluate the performance on predicting the positively and negatively labeled instances simultaneously. We use these measures because they were also used in previous work using these datasets (Wan and Freitas, 2015, 2017; da Silva et al., 2018). The values of those performance metrics are obtained by conducting a well-known stratified 10-fold cross validation.

Table 1: Characteristics of the ageing-related datasets.

	Worm Datasets			Fruit-fly Datasets			Mouse Datasets			Yeast Datasets		
Feature Types	**#F**	**#E**	**#I**	**#F**	**#E**	**#I**	**#F**	**#E**	**#I**	**#F**	**#E**	**#I**
BP	830	1,437	528	698	1,190	127	1,039	1,836	102	679	1,223	215
MF	218	259	279	130	151	102	182	205	98	175	209	157
CC	143	217	254	75	101	90	117	160	100	107	168	147
BP+MF	1,048	1,696	553	828	1,341	130	1,221	2,041	102	854	1,432	222
BP+CC	973	1,654	557	773	1,291	128	1,156	1,996	102	786	1,391	234
MF+CC	361	476	432	205	252	123	299	365	102	282	377	226
BP+MF+CC	1,191	1913	572	903	1,442	130	1,338	2,201	102	961	1,600	238

4.2 Results on Predictive Accuracy

Table 2 shows the GMean values obtained by five different classification methods over 28 datasets. Overall, Hie-AODE-Lite is the best-performing method, due to its highest GMean values (figures in bold) obtained on 13 out of 28 datasets and the best average rank of 2.04. The second best-performing method is Hie-AODE, which obtained the highest GMean values on 7 out of 28 datasets and the second best average rank of 2.45. The conventional AODE method obtained the highest GMean values on 5 out of 28 datasets with an average rank of 2.55, which is better than proximalGraph's and GO-BAN's average ranks, i.e. 3.43 and 4.53.

We used the pairwise two-tailed Wilcoxon signed-rank test at the 0.05 significance level to compare the average ranks based on the GMean values obtained by Hie-AODE-Lite and each of the other four methods. The results confirm that Hie-AODE-Lite significantly outperforms the conventional AODE, proximalGraph and GO-BAN, with p-values of 3.8e-02, 7.20e-03 and 4.19e-05, respectively. In addition, there is no significant difference between the predictive performance of Hie-AODE-Lite and Hie-AODE, due to a p-value of 5.8e-02.

Note that GO-BAN performed much worse than Hie-AODE and Hie-AODE-Lite, even though GO-BAN also uses the pre-defined feature hierarchy to constrain its feature dependencies network, based on the same principle used by Hie-AODE and Hie-AODE-Lite to constrain their one-dependence estimators. An explanation for the much worse performance of GO-BAN is that its dependence network doest not represent dependencies between non-hierarchically related features, where a feature is neither an ancestor nor a descendant of the other feature. An example is the pair of features F and M in Figure 2. By contrast, Hie-AODE and Hie-AODE-Lite are more flexiable, they represent such dependencies.

Table 2: Predictive accuracy for conventional AODE, Hie-AODE, Hie-AODE-Lite, proximalGraph and GO-BAN methods.

Feature Types	AODE			Hie-AODE			Hie-AODE-Lite			proximalGraph			GO-BAN		
Worm (*Caenorhabditis elegans*) Datasets															
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	40.2±3.5	79.6±2.3	**56.6**	40.6±3.5	77.1±2.5	55.9	40.1±3.1	79.6±3.1	56.5	23.5±1.8	76.8±1.9	42.4	28.7±2.2	86.5±1.8	49.8
MF	49.6±2.9	48.8±5.2	49.2	49.6±3.4	48.2±4.9	48.9	55.3±3.2	48.8±4.6	51.9	63.5±4.8	44.4±4.6	**53.1**	34.7±4.5	66.5±4.5	48.0
CC	43.2±5.1	74.4±2.5	56.7	42.3±4.5	72.4±3.0	55.3	48.3±6.0	69.9±3.7	**58.1**	63.0±5.7	38.0±4.4	49.0	33.7±4.5	81.4±2.2	52.4
BP + MF	40.9±2.8	78.8±1.9	56.8	46.0±2.7	77.4±2.2	**59.7**	39.8±3.0	77.4±2.0	55.5	20.2±2.6	76.8±1.3	39.4	30.0±2.7	84.7±1.7	50.4
BP + CC	40.9±3.0	80.5±2.2	**57.4**	41.8±2.6	77.9±2.6	57.1	40.8±1.9	79.1±2.2	56.8	24.0±2.0	75.3±1.2	42.5	29.1±2.1	86.6±1.7	50.2
MF + CC	46.5±2.8	71.3±3.8	57.6	50.0±2.2	67.5±3.9	**58.1**	45.3±3.5	68.3±3.6	55.6	57.1±5.6	41.6±3.5	48.7	35.3±2.9	80.2±3.2	53.2
BP + MF + CC	39.0±2.8	78.5±2.1	55.3	39.9±3.1	77.1±1.8	55.5	39.8±4.0	78.8±2.4	**56.0**	23.3±2.4	72.2±3.5	41.0	31.2±2.9	85.2±1.5	51.6
Fly (*Drosophila melanogaster*) Datasets															
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	85.7±2.9	23.3±9.5	44.7	86.8±2.8	29.2±10.0	**50.3**	87.8±3.5	26.7±10.4	48.4	60.2±5.3	34.2±8.8	45.4	100.0±0.0	0.0±0.0	0.0
MF	82.6±6.2	36.5±6.1	54.9	83.1±4.5	36.5±6.1	**55.1**	83.1±4.5	27.8±3.4	48.1	54.9±5.6	54.0±8.7	54.4	91.2±3.3	26.5±3.4	49.2
CC	85.4±5.8	40.0±9.7	58.4	82.1±6.8	43.3±10.0	59.6	85.4±5.2	46.7±11.3	**63.2**	41.3±6.6	70.0±8.8	53.7	93.5±2.6	28.6±11.1	51.7
BP + MF	88.9±4.4	37.5±5.6	57.7	87.8±4.2	37.5±5.6	57.4	91.1±4.0	37.5±5.6	**58.4**	72.4±3.3	27.5±7.5	44.6	97.8±1.5	0.0±0.0	0.0
BP + CC	86.7±4.0	42.5±8.7	60.7	84.6±4.1	42.5±8.7	60.0	85.7±3.3	45.0±8.5	**62.1**	71.2±5.7	30.0±8.3	46.2	98.9±1.1	0.0±0.0	0.0
MF + CC	94.2±3.2	45.0±3.3	65.1	94.2±3.2	47.5±4.5	**66.9**	94.2±3.2	42.5±3.8	63.3	61.4±3.9	60.0±6.3	60.7	95.3±1.9	31.6±5.3	54.9
BP + MF + CC	87.8±2.6	37.5±8.5	57.4	87.8±2.6	37.5±8.5	57.4	88.9±2.3	40.0±9.3	**59.6**	78.0±4.8	30.0±6.9	48.4	98.9±1.1	2.6±2.5	16.0
Mouse (*Mus musculus*) Datasets															
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	92.9±3.2	44.5±5.9	**64.3**	92.9±3.2	44.5±5.9	**64.3**	91.4±3.8	44.5±5.9	63.8	28.6±5.3	68.8±8.1	44.3	98.5±1.4	26.5±5.0	51.1
MF	77.1±5.2	40.0±11.2	55.5	77.1±5.2	40.0±11.2	55.5	79.8±4.4	43.3±10.4	58.8	69.0±6.7	57.5±4.5	**63.0**	90.8±3.3	27.3±10.0	49.8
CC	83.8±3.6	31.0±7.9	51.0	81.0±3.3	31.0±7.9	50.1	82.4±3.1	39.0±11.8	**56.7**	81.0±5.5	35.7±8.5	53.8	86.4±3.3	35.3±11.2	55.2
BP + MF	91.4±3.2	37.7±6.7	58.7	91.4±3.2	37.7±6.7	58.7	88.6±2.9	37.7±6.7	57.8	66.0±4.1	61.0±7.5	**63.5**	98.5±1.4	29.4±6.4	53.8
BP + CC	88.6±4.7	36.2±6.5	56.6	90.0±4.8	39.5±8.7	59.6	85.7±4.8	42.8±7.6	**60.6**	80.0±5.0	37.8±9.5	55.0	98.5±1.4	29.4±6.4	53.8
MF + CC	88.6±4.2	51.8±12.4	**67.7**	90.0±3.7	48.5±11.3	66.1	88.6±3.6	48.5±11.3	65.6	77.4±4.9	37.8±9.3	54.1	91.2±3.2	26.5±8.8	49.2
BP + MF + CC	90.0±4.8	38.7±8.2	59.0	90.0±4.8	38.7±8.9	59.0	90.0±4.3	42.0±8.6	**61.5**	79.4±5.4	40.3±9.9	56.6	98.5±1.4	26.5±10.5	51.1
Yeast (*Saccharomyces cerevisiae*) Datasets															
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	6.7±4.4	97.8±1.2	25.6	6.7±4.4	97.8±1.2	25.6	13.3±5.4	93.0±2.3	**35.2**	96.7±3.2	10.8±2.7	32.3	0.0±0.0	100.0±0.0	0.0
MF	3.3±3.3	94.7±1.9	17.7	0.0±0.0	92.5±2.5	0.0	0.0±0.0	91.0±2.8	0.0	45.0±10.3	61.1±3.8	**52.4**	0.0±0.0	99.2±0.8	0.0
CC	15.0±7.6	97.6±1.2	38.3	18.3±7.6	96.8±1.3	42.1	21.7±7.5	96.0±1.3	**45.6**	10.0±6.3	86.2±5.3	29.4	12.5±6.1	99.2±0.8	35.2
BP + MF	3.3±3.3	98.9±0.7	18.1	10.0±5.1	98.9±0.7	31.4	13.3±5.4	94.8±1.1	35.5	53.3±9.7	59.3±3.4	**56.2**	0.0±0.0	100.0±0.0	0.0
BP + CC	10.0±5.1	99.5±0.5	31.5	13.3±5.4	99.5±0.5	36.4	23.3±7.1	95.0±1.7	**47.0**	23.3±6.7	82.9±3.5	44.0	0.0±0.0	100.0±0.0	0.0
MF + CC	26.7±9.7	97.4±1.2	**51.0**	26.7±9.7	97.4±1.2	**51.0**	26.7±9.7	94.9±0.8	50.3	21.7±7.5	85.3±2.2	43.0	0.0±0.0	100.0±0.0	0.0
BP + MF + CC	3.3±3.3	99.5±0.5	18.1	6.7±4.4	99.5±0.5	25.8	23.3±8.7	94.7±1.8	**47.0**	20.0±7.0	82.6±2.6	40.7	0.0±0.0	100.0±0.0	0.0
Ave. Ranks		2.55			2.45			**2.04**			3.43			4.53	

Table 3: The formulas for calculating the number of dependency edges for different AODE algorithms.

Methods	#FF_Edges	#CF_Edges	#Edges
AODE	$\sum_{i=1}^{n}(n-1) = n^2 - n$	$\sum_{i=1}^{n} n = n^2$	$2n^2 - n$
Hie-AODE	$\sum_{i=1}^{n}(n-a_i-1) = n^2 - n - \sum_{i=1}^{n} a_i$	$\sum_{i=1}^{n} n = n^2$	$2n^2 - n - \sum_{i=1}^{n} a_i$
Hie-AODE-Lite	$\sum_{i=1}^{n}(n-a_i-1) = n^2 - n - \sum_{i=1}^{n} a_i$	$\sum_{i=1}^{n}(n-a_i) = n^2 - \sum_{i=1}^{n} a_i$	$2n^2 - n - 2\times\sum_{i=1}^{n} a_i$

4.3 The differences among the numbers of dependency edges for AODE, Hie-AODE or Hie-AODE-Lite

We further discuss the difference between AODE and our proposed methods (i.e. Hie-AODE and Hie-AODE-Lite) by comparing the total number of dependency edges per One Dependency Estimator (ODE). The total number of dependency edges #Edges for each ODE consists of two types of dependency edges, i.e. the feature–feature dependency edges, #FF_Edges; and the class–feature dependency edges, #CF_Edges. The formulas for calculating those figures for different AODE methods are summarised in Table 3, where n denotes the total number of features (which is also the number of ODEs), and a_i denotes the number of ancestors of the i_{th} feature in the pre-defined hierarchy.

Table 4 reports the total number of edges used by AODE (#Edges_AODE), and the corresponding difference to the smaller number of edges considered by Hie-AODE (#Diff_Edges_Hie-AODE) and Hie-AODE-Lite (#Diff_Edges_Hie-AODE-Lite), respectively. That is, in terms of the formulas shown in Table 3, #Diff_Edges_Hie-AODE = $\sum_{i=1}^{n} a_i$, and #Diff_Edges_Hie-AODE-Lite = $2\times\sum_{i=1}^{n} a_i$. In general, the values of #Edges_AODE range from 11,175 to 3,579,150 over all 28 different datasets. The datasets that consist only of the CC type of features lead to the smallest values of #Edges_AODE for each model organism, due to the fact that the number of CC type of features is also the smallest, compared with other types of features. Conversely, the datasets that consist of the combination of BP, MF and CC types of features lead to the largest value of #Edges_AODE for each model organism, because of their largest numbers of features. The values of #Diff_Edges_Hie-AODE range from 295 to 10,838, reflecting the smaller values of #FF_Edges for Hie-AODE, compared with AODE. Analogously, the values of #Diff_Edges_Hie-AODE-Lite range from 590 to 21,676, also reflecting the smaller values of both #FF_Edges and #CF_Edges for Hie-AODE-Lite, compared with AODE.

Table 4: The total number of edges for AODE and the difference to the number of edges for Hie-AODE and Hie-AODE-Lite.

Feature Types	Worm Datasets			Fly Datasets			Mouse Datasets			Yeast Datasets		
	#Edges_AODE	#Diff_Edges Hie-AODE	#Diff_Edges Hie-AODE-Lite	#Edges_AODE	#Diff_Edges Hie-AODE	#Diff_Edges Hie-AODE-Lite	#Edges_AODE	#Diff_Edges Hie-AODE	#Diff_Edges Hie-AODE-Lite	#Edges_AODE	#Diff_Edges Hie-AODE	#Diff_Edges Hie-AODE-Lite
BP	1,376,970	8,006	16,012	973,710	5,043	10,086	2158,003	9,743	19,486	921,403	7,681	15,362
MF	94,830	994	1,988	33,670	537	1,074	66,066	652	1,304	61,075	749	1,498
CC	40,755	648	1,296	11,175	295	590	27,261	443	886	22,791	524	1,048
BP+MF	2,195,560	9,000	18,000	1,370,340	5,580	11,160	2,980,461	10,395	20,790	1,457,778	8,430	16,860
BP+CC	1,892,485	8,654	17,308	1,194,285	5,338	10,676	2,671,516	10,186	20,372	1,234,806	8,205	16,410
MF+CC	260,281	1,642	3,284	83,845	832	1,664	178,503	1,095	2,190	158,766	1,273	2,546
BP+MF+CC	2,835,771	9,648	19,296	1,629,915	5,875	11,750	3,579,150	10,838	21,676	1,846,081	8,954	17,908

4.4 The robustness against class imbalanced distribution

We further investigate the predictive performance of the proposed Hie-AODE and Hie-AODE-Lite algorithms by evaluating their robustness against the class imbalance problem. We calculate the Pearson correlation coefficient r between the GMean values and the degree of class imbalance $\mathbf{D}$, which ranges from 0.0 to 1.0 and is calculated by Equation 4, where $\#Minor$ denotes the number of instances assigned to the minority class, and $\#Major$ denotes the number of instances assigned to the majority class.

$$\mathbf{D} = 1 - \frac{\#Minor}{\#Major} \tag{4}$$

The conventional AODE algorithm has a weak robustness against the class imbalance problem, due to the its strong negative correlation coefficient value r of −0.729, suggesting that in general higher degrees of class imbalance lead to lower GMean values. However, both Hie-AODE and Hie-AODE-Lite show somewhat stronger robustness, due to their better r values of −0.662 and −0.553, respectively. This is consistent with the fact that both algorithms obtain higher GMean values than the standard AODE algorithm. In addition, GO-BAN shows the weakest robustness (with r=−0.789) among all 5 algorithms, whereas proximalGraph shows the strongest robustness (with r=−0.283).

5. Conclusion

In this work, we propose two novel hierarchical dependency constrained averaged one-dependence estimators classification algorithms. Both proposed algorithms successfully outperformed the standard AODE algorithm (which ignores hierarchical feature dependencies), and also outperformed two other algorithms that exploit hierarchical feature dependencies during the classifier's training stage. This finding confirms the usefulness of the information about hierarchical dependencies between features for the task of classification, and encourages future research on new algorithms for exploiting this type of hierarchal dependency constraints. Future work could also involve experiments with other versions of AODE (Yang et al., 2006; Webb et al., 2012) and more general model averaging Bayesian network classifiers (Chen et al., 2016; Liao et al., 2018; Rohekar et al., 2018; Chen et al., 2018; Chen and Yuan, 2019; Talvitie et al., 2019).

References

C. Chen and C. Yuan. Learning diverse bayesian network. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, page 7793–7800, 2019.

C. Chen, C. Yuan, Z. Ye, and C. Chen. Solving m-modes in loopy graphs using tree decompositions. In *Proceedings of the 9th International Conference on Probabilistic Graphical Models*, page 145–156, 2018.

E. Y.-J. Chen, A. C. Choi, and A. Darwiche. Enumerating equivalence classes of bayesian networks using ec graphs. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, page 591–599, 2016.

P. N. da Silva, A. Plastino, and A. A. Freitas. A novel genetic algorithm for feature selection in hierarchical feature spaces. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 738–746, San Diego, USA, 2018.

R. Jenatton, J. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12, 2011a.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12, 2011b.

Z. Liao, C. Sharma, J. Cussens, and P. van Beek. Finding all bayesian network structures within a factor of optimal. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 7892–7899, 2018.

J. Mairal and B. Yu. Supervised feature selection in graphs with path coding penalties and network flows. *Journal of Machine Learning Research*, 14, 2013.

J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. In *Proceedings of the 2010 Advances Neural Information Processing Systems*, pages 1558–1566, San Diego, USA, 2010.

G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38, 1995.

P. Ristoski and H. Paulheim. Feature selection in hierarchical feature spaces. In *Proceedings of the International Conference on Discovery Science (DS 2014)*, pages 288–300, 2014.

R. Y. Rohekar, Y. Gurwicz, S. Nisimov, G. Koren, and G. Novik. Bayesian structure learning by recursive bootstrap. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 10546–10556, 2018.

T. Talvitie, A. Vuoksenmaa, and M. Koivisto. Exact sampling of directed acyclic graphs from modular distributions. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

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

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 *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)*, pages 373–380, Shanghai, China, Dec. 2013.

C. Wan and A. A. Freitas. A new hierarchical redundancy eliminated tree augmented naive bayes classifier for coping with gene ontology-based features. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016) Workshop on Computational Biology*, New York, USA, 2016.

C. Wan and A. A. Freitas. Two methods for constructing a gene ontology-based feature selection network for a Bayesian network classifier and applications to datasets of aging-related genes. In *Proceedings of the Sixth ACM Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB 2015)*, pages 27–36, Atlanta, USA, Sept. 2015.

C. Wan and A. A. Freitas. An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features. *Artificial Intelligence Review*, 2017.

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.

G. Webb, J. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1), 2005.

G. Webb, J. Boughton, F. Zheng, K. Ting, and H. Salem. Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive bayesian classification. *Machine Learning*, 86, 2012.

Y. Yang, G. Webb, J. Cerquides, K. Korb, J. Boughton, and K. M. Ting. To select or to weigh: A comparative study of model selection and model weighing for spode ensembles. In *Proceedings of the 17th European Conference on Machine Learning*, pages 533–544, Berlin, Germany, 2006.

Y. Zhao, M. Chung, B. A. Johnson, C. S. Moreno, and Q. Long. Hierarchical feature selection incorporating known and novel biological information: Identifying genomic features related to prostate cancer recurrence. *Journal of the American Statistical Association*, 111(516), 2016.