UNIVERSIDADE FEDERAL FLUMINENSE

PABLO NASCIMENTO DA SILVA

NOVEL FEATURE SELECTION METHODS FOR HIERARCHICAL AND UNCERTAIN FEATURE SPACES AND THEIR APPLICATION TO THE BIOLOGY OF AGEING

NITERÓI

2019

UNIVERSIDADE FEDERAL FLUMINENSE

PABLO NASCIMENTO DA SILVA

NOVEL FEATURE SELECTION METHODS FOR HIERARCHICAL AND UNCERTAIN FEATURE SPACES AND THEIR APPLICATION TO THE BIOLOGY OF AGEING

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: Engenharia de Sistemas e Informação.

Orientador:
ALEXANDRE PLASTINO

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

S586n Silva, Pablo Nascimento da
Novel Feature Selection Methods for Hierarchical and Uncertain Feature Spaces and Their Application to the Biology of Ageing / Pablo Nascimento da Silva ; Alexandre Plastino, orientador. Niterói, 2019.
99 f.

Tese (doutorado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: http://dx.doi.org/10.22409/PGC.2019.d.12067073729

1. Mineração de dados (Computação). 2. Aprendizado de máquina. 3. Seleção de atributos. 4. Classificação de dados . 5. Produção intelectual. I. Plastino, Alexandre, orientador. II. Universidade Federal Fluminense. Instituto de Computação. III. Título.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

PABLO NASCIMENTO DA SILVA

Novel Feature Selection Methods for Hierarchical and Uncertain Feature Spaces and Their Application to the Biology of Ageing

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: Engenharia de Sistemas e Informação.

Aprovada em 10 de Dezembro de 2019.

BANCA EXAMINADORA

Prof. Alexandre Plastino de Carvalho - Orientador, UFF

Prof. Daniel Cardoso Moraes de Oliveira, UFF

Profª. Flavia Cristina Bernardini, UFF

Prof. Eduardo Soares Ogasawara, CEFET-RJ

Fernanda Araujo Baião Amorim

Profª. Fernanda Araujo Baião Amorim, PUC-Rio

Niterói

2019

Aos meus pais Cinéa e Erivaldo.

Agradecimentos

Antes de mais nada, gostaria de agradecer aos meus pais, Erivaldo e Cinéa, por todo o carinho, suporte e incentivo que recebi durante toda a minha vida.

Ao meu orientador Professor Alexandre Plastino, por tudo que me ensinou durante essa longa jornada, pela paciência, por todo o apoio, incentivo e confiança fundamentais para o desenvolvimento desta tese.

Ao professor Alex A. Freitas pela colaboração durante o doutorado e por ter me recebido na University of Kent.

Aos professores e funcionários do IC/UFF por manterem um ambiente agradável e adequado para realização de pesquisas de alto nível.

Aos amigos que fiz na UFF e em Canterbury.

À CAPES pelo apoio financeiro.

A todos os demais que, direta ou indiretamente, contribuíram para o desenvolvimento desta tese.

Muito obrigado!!!

Abstract

Ageing is a complex process characterised by a continuous decline in the function of an organism that occurs with increasing age. The study of genes that regulates ageing could lead to the creation of new medicines and treatments to increase the lifespan of an organism. In this thesis, we focus on studying new methods to improve the performance of classifying ageing-related genes.

In this work, genes are described in terms of Gene Ontology (GO) features and Protein-Protein Interaction (PPI) features. Gene Ontology features are hierarchically organised. A hierarchical feature space is formed by binary features that are related via generalisation-specialisation relationships. Although there are many methods for the traditional feature selection problem, methods which properly consider hierarchical structures are still very underexplored. So, we propose two novel methods to cope with this problem: (i) a lazy method based on the hypothesis that positive feature values provide more meaningful and accurate information, named Select Relevant Positive Feature Values (RPV); and (ii) an evolutionary approach, named Genetic Algorithm for Hierarchical Feature Spaces (GA-HFS), applying two novel biased mutation operators tailored to deal with redundant features in hierarchical feature spaces.

The use of PPI features for classification is not straightforward since its values are uncertain, i.e., such values are numeric scores representing the likelihood of interaction between proteins. To address this problem, this work explores a new probabilistic Jaccard distance measure to handle uncertain features, which can be used within the nearest neighbour classifier. Additionally, we propose a novel Lazy Feature Selection Method for Uncertain Features (LFSUF), which deals with the uncertainty in features.

We empirically show that appropriately exploring the hierarchical and uncertain features can improve the predictive performance for the classification of ageing-related genes.

Keywords: Classification, Biology of Ageing, Feature Selection, Hierarchical Feature Spaces, Uncertain Feature Spaces

Resumo

O envelhecimento é um processo biológico complexo e pode ser caracterizado por um declínio contínuo das funções de um organismo que ocorre com o aumento da idade. O estudo de genes que regulam o envelhecimento pode levar à criação de novos medicamentos e tratamentos para aumentar a expectativa de vida de um organismo. Nesta tese, focaremos no estudo de novos métodos capazes de aumentar a capacidade preditiva de classificadores para identificar genes relacionados com o envelhecimento.

Neste trabalho, genes são descritos através de atributos da Gene Ontology (GO) e atributos que representam interações entre proteínas, chamados atributos PPI (Protein-Protein Interactions). Atributos da Gene Ontology são organizados hierarquicamente. Um espaço de atributos hierárquico é formado por atributos binários que possuem relações de generalização-especialização. Embora exista uma grande variedade de métodos para a seleção de atributos tradicional, métodos que consideram a estrutura hierárquica são pouco explorados. Portanto, são propostos dois métodos de seleção de atributos para explorar espaços de atributos hierárquicos: (i) um método lazy baseado na hipótese de que atributos positivos são mais relevantes e informativos para a classificação, chamado Select Relevant Positive Feature Values (RPV); e (ii) um método evolucionário chamado Genetic Algorithm for Hierarchical Feature Spaces (GA-HFS) que aplica dois novos operadores de mutação enviesada, especificamente desenvolvidos para tratar o problema dos atributos redundantes em espaços de atributos hierárquicos.

O uso the atributos PPI para classificação não é trivial, pois seus valores são incertos, i.e., tais valores são escores numéricos que representam a probabilidade de duas proteínas interagirem. Para solucionar este problema, este trabalho explora uma nova medida de distância chamada Probabilistic Jaccard para lidar com as características incertas dos atributos, e que pode ser utilizada com o algoritmo de classificação baseado nos vizinhos mais próximos. Adicionalmente, é proposto um novo método de seleção de atributos chamado Lazy Feature Selection Method for Uncertain Features (LFSUF), capaz de lidar com a incerteza dos atributos.

Mostramos empiricamente que explorar apropriadamente as características hierárquicas e incertas de atributos pode melhorar a capacidade preditiva de classificadores que identificam genes relacionados ao envelhecimento.

Keywords: Classificação, Biologia do Envelhecimento, Seleção de Atributos, Espaços de Atributos Hierárquicos, Espaços de Atributos Incertos.

List of Figures

List of Abbreviations

AUCPR	:	Area Under the Precision-Recall Curve;
BP	:	Biological Process;
CbHE	:	Correlation-based Hierarchical Elimination;
CC	:	Cellular Component;
CFS	:	Correlation-based Feature Selection;
DAG	:	Directed Acyclic Graph;
GA	:	Genetic Algorithm;
GA-HFS	:	Genetic Algorithm for Hierarchical Feature Selection;
GM	:	Geometric Mean;
GO	:	Gene Ontology;
GTD	:	Greedy Top-down Search Strategy;
HAGR	:	Human Ageing Genomic Resources;
HIP	:	Select Hierarchical Information-Preserving Features;
HIP-MR	:	Select Hierarchical Information-Preserving and Most Relevant Features;
KDE	:	Kernel Density Estimation;
KEGG	:	Kyoto Encyclopedia of Genes and Genomes;
KNN	:	K-Nearest Neighbours;
LFSUF	:	Lazy Feature Selection for Uncertain Features;
MF	:	Molecular Function;
MR	:	Select Most Relevant Features;
NB	:	Naïve Bayes;
PPI	:	Protein-Protein Interaction;
RPV	:	Select Relevant Positive Feature Values;
SHE	:	Simple Hierarchical Elimination;
TSEL	:	Tree-Based Feature Selection;
WEKA	:	Waikato Environment for Knowledge Analysis;

Contents

Chapter 1

Introduction

The classification task of machine learning is one of the most relevant types of supervised learning in the knowledge discovery scenario [15, 50]. A previously trained classification model automatically assigns a class label to an instance, based on the values of its features. The classification problem can be defined as follows. Let $X = \{X_1, \ldots, X_d\}$ be a set of d predictive features and $L = \{l_1, \ldots, l_q\}$ be a set of q class labels, where $q \geq 2$. Let $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be a dataset with N instances, where, for the i-th instance, x_i corresponds to a vector $(x_{i1}, x_{i2}, \ldots, x_{id})$, which stores values for the d features in X and each $y_i \in L$ corresponds to a single target class. The goal of the classification task is to learn a classifier from D that, given an unlabelled instance $t = (x, ?)$, predicts its class label y.

Ageing is a complex process characterized by a continuous decline in the function of an organism that occurs with increasing age, ultimately leading to death [20, 22, 32]. Even for related species, the speed at which such functional deterioration happens differs to some extent [20]. Although ageing research has advanced significantly in the last decades, it is still unclear which biological mechanisms contribute to the ageing process, even though genetic factors clearly make a major contribution to it [49].

Experiments in model organisms have identified several hundred genes that influence the ageing process (speeding it up or slowing it down) [37]. The discovery of such genes in model organisms may lead to the identification of homologous genes in humans, which could lead to pharmacological interventions to treat ageing. Hence, in this thesis, we are particularly interested in a problem from the biology of ageing, which is to automatically classify ageing-related genes into two different classes: Pro-longevity and Anti-longevity genes. Pro-longevity genes are those genes whose decreased expression reduces lifespan and/or those whose over-expression extends lifespan [44, 48]. Conversely, anti-longevity genes are those genes whose decreased expression extends lifespan and/or those whose over-expression decreases it.

This thesis focus on improving predictive models for ageing-related genes. So, we provide a brief overview of biological concepts to a better understanding of the biological roles of genes and proteins in an organism. Let's start with the definition of organism. An organism is any individual entity that propagates the properties of life. In other words, an organism is any kind of living individual [2]. Some examples of organisms are: viruses, bacteria and worms. Some of these organisms have been widely studied, usually due to a particular experimental advantage (e.g., easy to maintain and breed in a laboratory setting)[12]. Model organisms are non-human organisms that are used in the laboratory to help scientists understand biological processes, serving as a proxy for understanding the biology of humans [12]. In this thesis, we employ four types of model organisms: *Caenorhabditis elegans (worm)*, *Drosophila melanogaster (fly)*, *Mus musculus (mouse)* and *Saccharomyces cerevisiae (yeast)*. These are the most used model organisms since they are very well studied and have many proteins correlated with human proteins. Proteins are large, complex molecules that play many critical roles in the organism. They do most of the work in cells and are required for the structure, function, and regulation of the body's tissues and organs [2]. On the other hand, genes are molecular structures that contain information needed to make functional proteins. In other words, the genetic material of a gene can be seen as a recipe to the construction of a protein. The process of building a protein from a gene is not simple, and is tightly controlled within each cell [40]. This process is called gene expression, which is the main characteristic to define the classes used in this thesis. Note that, since one gene is closely linked with a single protein, in this thesis, the words gene and protein are interchangeable.

One particular problem of building datasets to the classification that relies on genes (and proteins) as input features is that gene's data (features) can be extracted from a large variety of sources. It is common to have more than one database collecting data for the same gene's characteristic. Among the most commonly used data are the Gene Ontology (GO) dataset [3], the Kyoto Encyclopedia of Genes and Genomes (KEGG) Pathways [21], Motif-based features [9] and Protein-Protein Interaction information [6]. Throughout this thesis, and in the literature as well, the task of predicting the class of ageing-related genes is mainly carried out using data derived from the Gene Ontology (GO) database [5, 10, 45, 46, 47, 48]. GO features are organized into an hierarchical structure (or a Hierarchical Feature Space). Each instance represents a gene, which may be associated with terms derived from an ontology of biological processes or functions.

Hence, a general feature (e.g., biological process) would be the ancestor of more specific features (e.g., reproduction, metabolic process and biological regulation).

In hierarchical feature spaces (e.g., GO features hierarchy), each instance in the dataset can be described as a binary feature vector, such that each feature takes either a positive or a negative value. Also, the binary features are linked via generalization-specialization relationships. In any given instance, a feature value is deemed positive (negative) when the property associated with the feature has been (has not been) observed for that instance. In a generalization-specialization hierarchy, also known as "IS-A" hierarchy, for any given instance t, if a feature x has positive value in t, denoted $(x = 1)$, then all ancestors of x in the feature hierarchy also have positive value in t. In contrast, if a feature x has negative value in t, denoted $(x = 0)$, then all descendants of x in the feature hierarchy also have negative value in t. Note that this type of hierarchical feature spaces are also sparse, i.e., in general, the instances contain much fewer positive than negative feature values.

There are other characteristics of genes (or proteins) that could be useful to the classification of ageing-related genes. So, in the second part of this thesis, we build predictive models using not only GO term features, but also Protein-Protein Interaction (PPI) features [6], a widely used characteristic of proteins that could potentially help understanding those proteins linked with ageing [11, 36]. In a PPI dataset, each PPI indicates whether or not a protein (instance or object to be classified) interacts with another protein. As PPI information is an important indicator of gene functions, the use of PPI features may improve the classifier's predictive accuracy. Also, as no protein works in isolation, the analysis of highly predictive PPI features could improve the interpretability of the classification model, leading to a better understanding of the ageing problem in general.

However, the use of PPI features for classification is not straight-forward. First, the values of PPI features are uncertain, i.e., such values are numeric scores representing the likelihood of interaction of two proteins (e.g., protein-A interacts with protein-B in 90% of the documented cases). Second, among the vast number of possible protein interactions, few interactions are realised, leading to a high feature sparsity and dimensionality. Note that the addition of PPI features brings a major challenge: the selection of the subset of protein interactions that are most suitable to perform an accurate prediction.

In both GO and PPI datasets, data usually have a large number of features, many of which are not important for predicting the correct class. Some features can be redundant (highly correlated with each other) or irrelevant for predicting the class variable, decreasing the classifier's predictive accuracy, making the learning process slower, and reducing the comprehensibility of the results.

Feature selection methods have been successfully employed to cope with these problems. They aim at selecting a reduced subset of features to predict the target class, yet increasing the predictive accuracy of the classifier [25, 26]. In other words, feature selection can be defined as finding a feature subset $F \subseteq X$, such that the predictive model trained on F ($h(F)$) has a higher predictive accuracy than the predictive model trained on X ($h(X)$). It is a challenging problem because the number of candidate feature subsets grows exponentially with the number of features. More precisely, the number of candidate feature subsets is $2^d - 1$, where d is the number of features.

Feature selection methods can be categorized into embedded, wrapper and filter methods [25, 26]. Embedded methods are incorporated into the classification algorithm, selecting features during the construction of a classification model. Wrapper and filter methods are instead used in a data pre-processing step. Wrapper methods measure the relevance of a feature subset by evaluating the predictive accuracy of a classifier built using that subset. Hence, they select features tailored to the target classification algorithm, but they tend to be very time-consuming. By contrast, filter methods generically evaluate the predictive power of features, by using a relevance measure that is independent of the target classification algorithm. Filter methods tend to be much faster and more scalable than wrapper methods.

Feature selection methods (as well as classification methods) can also be categorized as eager or lazy. Eager methods select a subset of features based on the training instances. Then, a model trained with the selected features is used to predict the class of any test instance. By contrast, lazy methods select a feature subset tailored for each test instance [1, 35], by observing the feature values (but not the class, of course) in that test instance. Consequently, lazy learning methods use one classification model for each testing instance, while eager methods build a single classifier for all testing instances.

Note, however, that although many methods address the feature selection problem [13, 24, 25, 26, 27, 35], only few of them explore the hierarchical or the uncertain information in order to improve their effectiveness [39, 45, 46, 48]. Existing hierarchical feature selection methods usually find a suitable subset of features by keeping those features with higher values of relevance and removing redundancy among hierarchically related features. On the other hand, to the best of our knowledge, there is no available uncertain feature selection method capable of exploring the type of uncertain features used in this thesis.

The main goal of this thesis is to present novel feature selection methods for improving the predictive performance of models for classifying ageing-related genes into two classes: Pro-/Anti- Longevity Genes.

We propose the following approaches to achieve this goal: (i) the introduction of novel feature selection methods capable of coping with the hierarchical structure present in the Gene Ontology datasets. These hierarchical feature selection methods are specially designed for being capable of exploring the information contained in the hierarchy of features. In this thesis, two novel hierarchical feature selection methods are proposed: a lazy learning method called Select Relevant Positive Feature Values (RPV) and an eager wrapper approach based on Genetic Algorithms called Genetic Algorithm for Hierarchical Feature Selection (GA-HFS). (ii) the proposal of a lazy feature selection method called Lazy Feature Selection for Uncertain Features (LFSUF) specially designed to cope with datasets with uncertain features; and (iii) a new probabilistic Jaccard distance measure to handle uncertain features, which can be used within the nearest neighbour classifier.

This thesis is composed of a collection of four research articles and this complementary text linking their content. This text provides a thorough overview of this thesis' objectives and contributions, whereas each research article provides the details about one contribution, its experimental evaluation and conclusions. Research articles are provided as Appendix of this text. The following articles are part of this thesis:

1. **da Silva, P.N.**, Plastino, A., Freitas, A.A. "Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method" - Submitted to Applied Intelligence.

2. **da Silva, P.N.**, Plastino, A., Freitas, A.A. "A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces". In Proc. of the 2018 SIAM International Conference on Data Mining (SDM) (2018), pages 738-746. SIAM.

3. Martire, I., **da Silva, P.N.**, Plastino, A., Fabris, F., Freitas, A.A. "A Novel Probabilistic Jaccard Distance Measure for Classification of Sparse and Uncertain Data". In Proc. of Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) (2017), pages 81-88.

4. **da Silva, P.N.**, Plastino, A., Fabris, F., Freitas, A.A. "A Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti-Longevity Genes" - Submitted to IEEE/ACM Transactions on Computational Biology and Bioinformatics.

The first article presents the proposal of a lazy feature selection method for hierarchical datasets that prioritizes, for each instance being classified, the selection of positive features rather than negative features. We also propose a new relevance measure to evaluate the predictive importance of a feature given its value on the instance being predicted. Currently, this paper is going through the second round of reviews in the journal Applied Intelligence.

The second article introduces a novel feature selection method based on a genetic algorithm search that follows the wrapper paradigm. It introduces two new mutation operators for genetic algorithms that are capable of handling the hierarchical structure of the GO datasets by removing hierarchical correlation among features and consequently improving the predictive performance of the classification task. This work was published in the conference SIAM SDM 2018.

The third article presents a new distance measure capable of exploring the uncertain values in the PPI features. It was designed to work with the nearest neighbour classifier. This work was done in collaboration with Igor Martire while I serving as his undergraduate dissertation's co-advisor, and it was published in the conference KDMiLe 2017.

The fourth article introduces a novel feature selection method able to cope with datasets with uncertain features, like those present in PPI features. We also demonstrate that using PPI and GO features together can increase the predictive performance of the problem of classifying genes into Pro-Longevity and Anti-Longevity. This article is currently under review in the IEEE/ACM Transactions on Computational Biology and Bioinformatics.

The remainder of this thesis is organised as follows. In Chapter 2, we present a brief overview of articles 1 and 2. So, we introduce the concept of hierarchical feature spaces, and the Gene Ontology (GO) features used in this thesis, discuss the related work and give a brief overview of this thesis' contributions in the area of feature selection for hierarchical feature spaces. In Chapter 3, we present a brief overview of articles 3 and 4. In other words, we describe the concept of uncertain feature spaces and the Protein-Protein Interaction (PPI) features used in this thesis, discuss the relevant related work and give a brief overview of contributions for uncertain feature spaces. In Chapter 4, we present results of comparisons among the proposed methods. It was done to understand which of the proposed methods has the best performance overall. The content of Chapter 4 is new and is not present in any other research article. Lastly, conclusions and future work are present in Chapter 5.

Chapter 2

Feature Selection for Hierarchical Feature Spaces

The main objective of this chapter is to introduce the feature selection methods proposed in this thesis, capable of exploring the structure present in hierarchical feature spaces. This chapter is organised as follows. First, in Section 2.1, we define the hierarchical feature space and introduce the Gene Ontology (GO) datasets. Section 2.2 presents the related work. Lastly, Section 2.3 describes the contributions of this thesis to the classification of ageing-related genes by using novel hierarchical feature selection methods.

2.1 Hierarchical Feature Spaces

When the feature set X in a dataset D is hierarchically structured, we call it a hierarchical feature space. This can be represented as a Direct Acyclic Graph (DAG). In this DAG, a vertex (node) represents a feature, and an edge represents a generalization-specialization relationship between features. In this sense, an edge ($X_a \rightarrow X_b$) indicates that X_a is a parent (immediate ancestor) of X_b and X_b is a child (immediate descendant) of X_a. More generally, a feature X_c is an ancestor of a different feature X_d if and only if there is a sequence of edges leading from X_c to X_d in the feature DAG, consequently, we say that X_d is a descendant of X_c. The root nodes are the most general features, while the leaf nodes are the most specific ones. In generalization-specialization hierarchies (also known as "IS-A' hierarchy'), for any give instance t, if a feature x has positive value in t (i.e., $x = 1$), then all ancestors of x in the hierarchy also have positive values in t. In contrast, if a feature x has negative value in t (i.e., $x = 0$), then all descendants of x in the feature hierarchy also have negative value in t. Note that this structure produces a hierarchical redundancy among features, since a specific feature value logically implies the values of all its ancestors or descendants: all ancestors of positive-valued features have positive values and all descendants of negative-valued features have negative values.

In this thesis, we describe genes (proteins) as gene ontology features extracted from the Gene Ontology (GO) dataset [3], an example of hierarchical feature space. GO annotates genes using terms from an expert-defined ontology. These annotations are from three different types: (i) Molecular Function (MF), which describes the molecular activities of individual genes; (ii) Cellular Component (CC), which contains information about where the gene products are active; and (iii) Biological Process (BP), containing the pathways and more general processes to which that gene product's activity contributes.

Figure 2.1 shows a sample of feature DAG extracted from the Gene Ontology dataset. We can use this figure to illustrate the concepts previously presented. For example, feature GO:0008150 (biological process) is the root of the DAG; GO:0019954 (asexual reproduction) is one child node of GO:0000003 (reproduction); and GO:0032501 (multicellular organism process) is the parent of both GO:0032504 (multicellular organism reproduction) and GO:0003008 (system process). Also, we can check node's ancestors in the same figure. For instance, if the feature node GO:0003008 is annotated (i.e., it has a positive value for a given instance), obligatorily nodes GO:0032501 and GO:0008150 will also be annotated with positive values. Similarly, if feature node GO:0000003 is not annotated (i.e., it has a negative value for a given instance), its decedents will also have negative values.

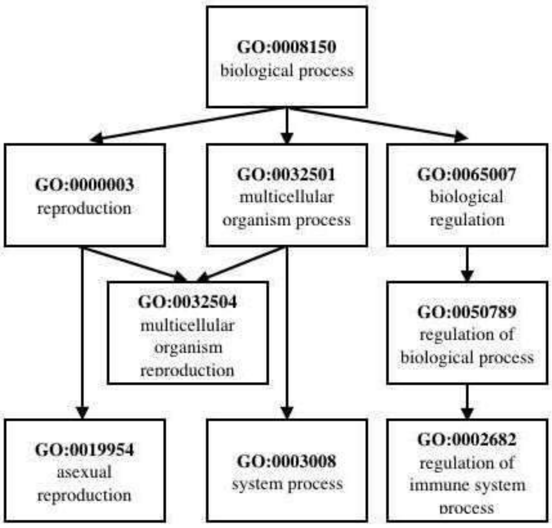


Figure 2.1: Example of hierarchical dataset from the Gene Ontology hierarchy DAG.

In the datasets explored in this work, each instance represents a gene, and each gene is associated with terms derived from the Gene Ontology. We built 28 datasets of ageing-related genes, involving the effect of genes on an organism's longevity. These datasets were built by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (version: Build 17) [30] and the Gene Ontology (GO) database (version: 2015-10-10) [3]. HAGR is a database with information about ageing-related genes in four model organisms: *C. elegans (worm)*, *D. melanogaster (fly)*, *M. musculus (mouse)* and *S. cerevisiae (yeast)*. The GO database provides three ontology types: biological process (BP), molecular function (MF) and cellular component (CC). So, for each of the 4 model organisms, we built 7 datasets, with 7 combinations of feature types (feature hierarchies), denoted: BP, CC, MF, BP.CC, BP.MF, CC.MF and BP.CC.MF. Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term in the GO hierarchy and a binary class variable indicating if the instance is either positive (pro-longevity gene) or negative (anti-longevity gene) according to the HAGR database. A full description of the datasets can be found in Section 5.1 of the article "Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method".

2.2 Related Work

Traditional (non-hierarchical) feature selection methods can be employed in hierarchical feature spaces by ignoring the hierarchical relationships among features. For instance, the non-hierarchical eager Correlation-based Feature Selection (CFS) method selects a subset of features that are weakly correlated with each other (little redundancy) and highly correlated with the class variable [13]. Another example of traditional feature selection is the eager ReliefF method which estimates the quality of attributes according to how well their values distinguish among instances that are near to each other [24]. However, using hierarchical methods on hierarchical feature spaces may avoid, more effectively, the selection of hierarchically redundant features.

Hierarchical feature selection methods are a special case of feature selection methods that exploit characteristics of the feature DAG to improve the predictive accuracy. This is typically done by removing hierarchically redundant features [39, 48].

SHSEL [39] is a hierarchical feature selection method that performs eager learning. SHSEL assumes that, if two features are directly hierarchically related (one is a parent of the other), they are usually highly correlated and tend to be similarly relevant for building the classification model. Hence, for each pair of directly hierarchically related features, SHSEL removes the most specific feature (in the hierarchy) if the correlation between them is higher than a user-specific threshold. After that, using only the remaining features, it keeps for each path in the hierarchy the features whose relevance is higher than the average relevance of features in that path. Moreover, the Greedy Top-Down search strategy (GTD) [29] is an eager method that selects the features with the highest relevance value in each path from each leaf to the root node in the hierarchy. Likewise, an eager learning hierarchical method called Tree-based feature Selection (TSEL) [19] has been used in the special case of tree-structured features.

Some hierarchical methods proposed in the literature are based on the lazy learning paradigm, such as the Select Hierarchical Information-Preserving Features (HIP) method [48], the Select Most Relevant Features (MR) method [48], and the hybrid HIP-MR method [45, 48]. Since the hybrid HIP-MR obtained worse results than its base methods HIP and MR in [45, 48], it is no longer considered. Next, we briefly describe HIP and MR.

The HIP method eliminates hierarchical redundancy by selecting only the core features in the current test instance – i.e., features whose values are non-redundant since they cannot be inferred from the values of other features. In other words, HIP selects the subset of the most specific positive-valued features (which imply their ancestors) and the most general negative-valued features (which imply their descendants). The values of the features selected by HIP for an instance imply the values of all other features for that instance, so it ensures that hierarchical redundancy is completely eliminated. However, HIP does not take into account the relevance of the selected features.

In a similar vein, the MR method not only eliminates hierarchical redundancy but also selects features with higher relevance. For each feature in the DAG, MR considers all paths between the feature and the root (for positive feature values) or between the feature and the leaves (for negative values). Then, the most relevant feature in each path is kept. However, unlike HIP, in general, MR does not select all core features, i.e., it removes some hierarchically non-redundant features.

It is worth noting that there is some work being done to cope with the problem of feature selection for hierarchical feature spaces. However, we could identify some unexplored characteristics of hierarchical features that can improve the performance of such methods. For example, none of these methods makes a distinction between positive and negative feature values, which could lead to better algorithms since positive values are much more informative and scarce. Also, there is no intelligent heuristic search being applied to the selection of hierarchical features.

2.3 Contributions

In this section, we briefly describe the contributions and outline the main results of the lazy feature selection method Select Relevant Positive Feature Values (RPV) and the Genetic Algorithm for Hierarchical Feature Selection (GA-HFS)[5] which employs biased hierarchical mutation operators. The first method is described in the article "Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method" submitted to the journal Applied Intelligence, the full text is available in Appendix A and a summary of this contribution is provided in Section 2.3.1. The second method is described in article "A novel genetic algorithm for feature selection in hierarchical feature spaces" published in the SIAM International Conference on Data Mining (SIAM SDM), available in Appendix B. A summary of the second contribution is provided in Section 2.3.2. All datasets and code used in these contributions are available at `https://github.com/pablonsilva/thesis_resources`.

2.3.1 Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method

The Select Relevant Positive Feature Values (RPV) method, the first contribution of this thesis, is a novel lazy feature selection method that explores a singular characteristic: in many hierarchical and sparse datasets, a positive feature value is almost always much more informative than a negative feature value. This characteristic can be well illustrated on ageing datasets where instances represent a Pro-/Anti-longevity class and features represent hierarchically related gene functions. In this type of bioinformatics dataset, a positive feature value is clearly informative, because the assignment of positive feature values to instances is based on the results of biological experiments confirming that certain function(s) has(ve) been observed for a given gene. However, a negative feature value is less informative and harder to interpret, because in general it just means the corresponding gene function "has not been observed yet", i.e., it is possible that no experiment has been performed yet to determine whether or not the gene has the function represented by a given feature, since biological experiments are very time consuming and costly. That is, a negative value is best interpreted as "lack of evidence of a gene function", rather than "evidence that the gene does not have that function".

Hence, discovered patterns including positive feature values (which are less common in typically sparse bioinformatics datasets) are more meaningful to biologist users and may be more accurate than patterns including mainly negative feature values. Nevertheless, none of the previously proposed feature selection methods for hierarchical and sparse spaces has yet explored this remarkable peculiarity of positive feature values. Hence, we hypothesize that prioritizing positive feature values might increase the predictive accuracy of a classifier. This contribution can be divided into two parts, described as follows: Firstly, we propose a novel lazy feature selection method for hierarchical and sparse feature spaces which relies on the higher relevance of features with positive values for the classification task. The basic idea of this method is to select, for each test instance, a subset of the most specific positive features in the hierarchy as well as its relevant ancestors. Secondly, we introduce a new lazy version of a relevance measure that evaluates the predictive relevance of a feature value for the current test instance.

Our method, named Select Relevant Positive Feature Values (RPV), has some interesting properties: (i) it selects rare but informative and relevant positive features; (ii) it selects smaller feature subsets; and (iii) it is based on a new lazy feature relevance measure (LazyR) which assesses the predictive power of a feature value specifically in the current test instance being classified.

The computational experiments involved 33 datasets (28 datasets using GO features and 5 datasets from other domains, mainly text classification). We compare our proposed method against the traditional ReliefF and CFS feature selection methods, and against HIP, MR and SHSEL. Also, we compared the proposed relevance measure (LazyR) to the widely used Information Gain [4] and R [41, 45] measures. Experiments were carried out using two different classification algorithms (Naïve Bayes and 1-NN) and two different predictive accuracy measures (AUCPR and the Geometric Mean of Sensitivity and Specificity). The results of these experiments have shown that the proposed RPV method obtained in general the best predictive accuracy across those four classification scenarios. More precisely, a statistical significance test was used to compare RPV against each of 10 other feature selection approaches, in each of the above four classification scenarios; and the results of that test have shown that RPV obtained predictive accuracy statistically significantly better than another feature selection approach in 28 out of the 40 cases. In addition, in none of those cases, RPV's predictive accuracy was significantly worse than the accuracy of any other feature selection approach. Furthermore, RPV selected, in general, the smallest subset of features, among all evaluated feature selection methods. Results of this experiments have shown that RPV achieved the better predictive performance for the 28 bioinformatics datasets, and also for the 5 datasets from the general domain, showing that RPV work well for more than one domain.

2.3.2 A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces

A Genetic Algorithm (GA) is a stochastic search method inspired by Charles Darwin's natural evolution theory [38]. A GA works with a population of individuals (candidate feature subsets) that iteratively undergo selection and modification, evolving towards a good solution for a given problem. In essence, a GA works as follows. First, an initial population of individuals is randomly created. Then, the quality of each individual is evaluated by a fitness function. At each generation (iteration), the best individuals (those with the highest fitness values) are selected more often for reproduction. The selected individuals undergo genetic operations, like crossover (which combines parts of two individuals to create a new individual) and mutation (where a small part of an individual is replaced according to a randomly generated value). The reproduction process produces offspring which will replace the parents, creating a new generation of individuals which are expected to be better than the previous generation's individuals. This process is repeated until a stopping criterion (e.g., a fixed number of generations) is satisfied.

Redundant features can potentially decrease the predictive capacity of the classifier and should be eliminated. This is a huge problem in hierarchical datasets because when two features are hierarchically related (i.e., when two features are on the same path within the hierarchy), they tend to be highly correlated (or redundant) with each other. So, the second main contribution of this thesis is the proposal of a Genetic Algorithm that employs a genetic operator (a mutation operator to be more precise) able to explore the hierarchy of features, reducing the internal redundancy present in hierarchical datasets. This method is called Genetic Algorithm for Hierarchical Feature Selection (GA-HFS). In this thesis, two distinct genetic mutation operators for genetic algorithms are proposed. These two operators are based on the principle that reducing the number of hierarchically redundant features often leads to higher predictive accuracy. Also, the proposed operators attempt to reduce the number of correlated features, by applying to each feature a different probability of mutation. This probability is defined according to the correlation among hierarchically redundant features in the candidate solution.

In total, we introduce three versions of a genetic algorithm (GA) for feature selection. The first one employs a simple mutation, considering a single value of mutation to all features without considering the relationship among the features. The other two mutation operators are tailored for feature selection in hierarchical feature spaces. The first operator, Simple Hierarchical Elimination (SHE) mutation, sets a fixed biased mutation probability to each feature with hierarchical redundancy, where the probability of removing such features is greater than the probability of changing the selection status of other features. The second mutation operator, Correlation-based Hierarchical Elimination (CbHE), sets the probability of removing a hierarchically redundant feature in a data-driven way, based on the correlation among hierarchically related features.

The experiments compared the predictive accuracy of Naïve Bayes with features selected by 8 different approaches. The methods were evaluated on the 24 datasets using Gene ontology features. Four datasets were kept out of this evaluation since they were used by the irace [28] procedure to calibrate the parameters of the GA-HFS (population size, number of generations, elitism size, tournament size, crossover probability and mutation probability). In summary, the two proposed GAs using the two novel hierarchical mutation operators achieved better predictive accuracies than traditional and state-of-the-art hierarchical feature selection methods. Actually, those two best GAs obtained significantly higher predictive accuracy than 4 or 5 other approaches, depending on the accuracy measure (GM or AUCPR). Also, those two best GAs, using new hierarchical mutation operators, selected overall substantially fewer features than the GA using a non-hierarchical mutation operator.

Chapter 3

Feature Selection for Uncertain Feature Spaces

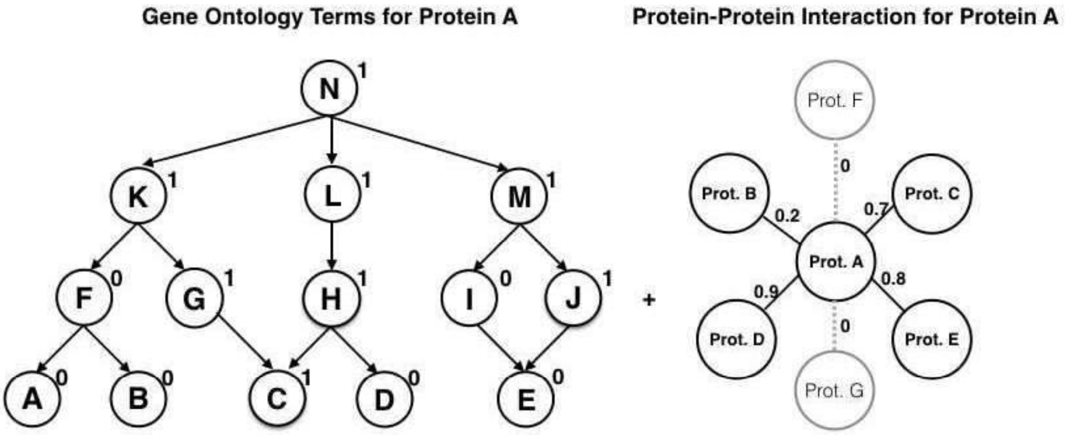
The main objective of this chapter is to introduce the methods, proposed in this thesis, capable of exploring the structure present in uncertain feature spaces. This chapter is organised as follows. First, in Section 3.1, we define the uncertain feature space. Section 3.2 presents the related work. Lastly, Section 3.3 describes the contributions of this thesis by exploring uncertain feature spaces.

3.1 Uncertain Feature Spaces

In this work, we explore Protein-Protein Interactions (PPI) features to build datasets and predict Anti-/Pro-Longevity Genes. PPIs are defined as physical contacts (or functional interactions) between proteins that occur in a cell or a living organism [6]. There are many different databases describing interactions between proteins [16, 42, 33, 34, 43]. In this work, we use the STRING database [42, 43], which contains a collection of known and predicted protein-protein interactions. These interactions can be either direct (physical interaction) or indirect (functional interaction). The information available in this database comes from the following sources: computational prediction, lab experiments, knowledge transfer between organisms, automated text mining and from interactions observed in other databases. In the STRING dataset, each PPI is associated with a score calculated from the information in the database that indicates the confidence of certain interaction being actually present. I.e., a high confidence score means that there is more support regarding a given interaction in the database.

The uncertain feature spaces addressed in this work is defined as follows. Given an instance $x_i = \{(x_{i1}, x_{i2}, \ldots, x_{id})\}$, each value x_{ij}, where $0 \leq x_{ij} \leq 1$, represents a certainty score defining how likely the the i-th instance is positively associated with the j-th feature. That is, if $x_{i1} > x_{i2}$, this means that the i-th instance is more likely to be positively associated with the first feature than to the second feature.

In Section 2.1, we demonstrate the process of building a dataset for the prediction of ageing-related genes using Gene Ontology features. These previously built datasets were enhanced with Protein-Protein Interactions (PPI) features. So, GO and PPI features are used together in each of the 28 datasets. In each dataset, for each gene (instance), we incorporated PPI features according to the data available in the STRING database. A full description of the datasets can be found in Section 2.4.3 of the article "Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti-Longevity Genes". Figure 3.1 depicts the structure of GO features (hierarchically organised) working alongside with PPI features. Note that, PPI features can be represented as a complete graph, where every pair of proteins (vertices) is linked by an edge. The edge of protein x and protein y contains a numeric value, representing the score of the interaction between proteins x and y.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	A-B	A-C	A-D	A-E	A-F	A-G
Protein A:	0	0	1	0	0	0	1	1	0	1	1	1	1	1	0.2	0.7	0.9	0.8	0.0	0.0

Figure 3.1: Example of hierarchical (GO dataset) and uncertain (PPI dataset) features.

Additionally, it is worth mentioning that the Gene Ontology contains uncertain information describing the confidence of a given annotation. However, in this thesis, the uncertain information of GO annotations is not being explored.

3.2 Related Work

Although uncertain features are present in many different applications (such as sensor data, PPI features, among others), there are very few methods capable of exploring uncertain features in the literature. For instance, in [23], a discriminative feature selection method for graph classification is introduced. It deals with graphs where the linkage of nodes are fundamentally uncertain (i.e., each connection between two nodes holds a likelihood of being a real connection). The graph structure used in that work is similar to those found in the Protein-Protein Interaction network. Note, however, that we are not interested in finding graph subsets, which is a significant difference between their approach and the one reported here.

Another feature selection method for uncertain data was proposed by [8]. They introduced a modified mutual information evaluation measure capable of dealing with uncertain features that is used in a two-step way. First, each feature is evaluated by the modified mutual information measure. Second, a threshold is used to select the x% of features with better mutual information values to build the classifier. However, the uncertain data employed is quite different from the one described in this work, since each feature value is described by a Gaussian distribution. Another significant difference is the fact that the data used to build the classification model is not initially uncertain. The Gaussian distribution is built as follows. First, the real feature values are used as the mean of the distribution, and a user-defined parameter is used as the standard deviation of the distribution (this parameter is equal for every instance/feature in the dataset). Note that, this approach is very different from the method described in our work, where the uncertainty information is given as an input. Also, apart from having a hard to tune user-defined parameter, it has another significant drawback; it cannot handle high-dimensional data since it relies on a Kernel Density Estimation (KDE) to compute the mutual information, which is notably a computationally expensive method [8].

3.3 Contributions

In this section, we briefly describe the contributions and outline the main results of a novel probabilistic Jaccard Distance Measure for Classification and a lazy feature selection for uncertain features (LFSUF). The first method is described in the following article: “A Novel Probabilistic Jaccard Distance Measure for Classification of Sparse and Uncertain Data”, published in the Symposium on Knowledge Discovery, Mining and Learning (KD-

MiLe). The full text and details are provided in Appendix C and a summary of this contribution is presented in Section 3.3.1. The second method is described in the article "A Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/anti-Longevity Genes" submitted to the journal IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). The full text and details of this chapter's second contribution are provided in Appendix D and a summary of this contribution is presented in Section 3.3.2. All datasets and code used in these contributions are available at `https://github.com/pablonsilva/thesis_resources`.

3.3.1 A Novel Probabilistic Jaccard Distance Measure for Classification of Sparse and Uncertain Data

As the third contribution of this thesis, we introduce a novel distance measure that takes advantage of uncertain values to increase the predictive performance of the nearest neighbours classifiers. This distance measure is called Probabilistic Jaccard (ProbJacc).

We presented a novel Jaccard distance measure for nearest-neighbour classification in sparse datasets with probabilistic binary features. We compared both the speed and the predictive performance of the 1-NN classifier using both our novel distance measure and the traditional Jaccard distance by applying internal cross-validation to optimise the cut-off value (Jaccard-ICV), i.e., transforming an uncertain value in a binary value according to a threshold parameter optimised by a internal cross-validation procedure.

The 1-NN classifier using the proposed ProbJaccard distance measure is significantly faster than the Jaccard-ICV method. This is due to the fact that ProbJaccard handles the uncertainty from the data directly, so there is no need to perform internal cross-validation to optimise a cut-off parameter. Additionally, our proposed method has shown an overall improvement in the predictive performance of the 1NN classifier across 28 ageing-related datasets.

3.3.2 Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti- Longevity Genes

The fourth contribution of this thesis is the proposal of a new feature selection method capable of exploiting the uncertainty values (or scores) present in Protein-Protein Interaction (PPI) features, where the higher the feature score, the higher the chance of the current instance (protein) actually interacting with the protein associated with the PPI feature. The novel method is called Lazy Feature Selection for Uncertain Features (LFSUF). The intuition behind this method is as follows. In the handled uncertain data, a feature value with high confidence (i.e., a feature value around one) means that the positive feature value has strong evidence of being actually present in an instance. Conversely, a feature value with low confidence (i.e., a feature value around zero) means that the feature is probably *not* present. Hence, LFSUF aims to select the subset of features whose positive value has the highest confidence (i.e., the highest likelihood of being present) in each test instance (adopting the lazy learning paradigm). Furthermore, the proposed method aims at selecting the subset of features which best correlates with the target class. In summary, the strategy aims at selecting, for each test instance, a subset of features with high confidence that also correlates well with the class.

The proposed LFSUF method obtained overall the best predictive accuracy in the classification of pro-longevity vs anti-longevity genes from four model organisms, when using two different classifiers (Naive Bayes and 1-NN) and two different types of feature sets – first, using both Gene Ontology (GO) and uncertain PPI features; and second, using only uncertain PPI features. Also, LFSUF achieved better predictive accuracy using smaller selected feature sets on average, when compared against other feature selection methods. This is desirable since it improves the interpretability potential of the predictions made by the model. In summary, our results indicate that the application of lazy feature selection on datasets with uncertain features is an effective approach, leading to higher predictive accuracy and better interpretability potential.

Chapter 4

Comparing the Proposed Feature Selection Methods

In this thesis, we propose three different feature selection algorithms tailored to exploit the hierarchical structure of GO features and a feature selection method capable of handling uncertain values. However, these methods were presented separately, in different articles, with no direct comparison between them, making it difficult to select the overall best method. This chapter presents a comparison between the feature selection methods proposed in this thesis. Section 4.1 presents the results of the comparison among the proposed feature selection methods for hierarchical feature spaces. Section 4.2 presents experimental evaluation of a hybrid method that performs feature selection by applying a specific method to each feature type (GO and PPI) in the dataset of ageing genes. It is worth mentioning that the content of this chapter is not present in any other research article.

4.1 Comparing the Predictive Performance of the Proposed Feature Selection Methods for Hierarchical Feature Spaces

In this section, we present a comparison of the proposed feature selection methods for hierarchical feature spaces. The following methods were compared:

- Select Relevant Positive Values (RPV) [Section 2.3.1, Appendix A]
- Genetic Algorithm for Hierarchical Feature Selection with SHE mutation operator (GA-HFS-SHE) [Section 2.3.2, Appendix B] [5]

- Genetic Algorithm for Hierarchical Feature Selection with CbHE mutation operator (GA-HFS-CbHE) [Section 2.3.2, Appendix B] [5]

All methods were implemented within the open-source WEKA data mining tool [14]. The methods were evaluated on the 24 datasets using Gene ontology features, as described earlier. Note that, four datasets were kept out of this evaluation since they were used by the irace [28] procedure to calibrate the parameters of the GA-HFS (population size, number of generations, elitism size, tournament size, crossover probability and mutation probability). The lazy k-NN with $k = 1$ (with Euclidean distance for hierarchical datasets and Probabilistic Jaccard distance [31] for hierarchical plus uncertain datasets) and a lazy version of Naïve Bayes (NB) (both from WEKA) were used as classification algorithms for all evaluated feature selection methods, and the predictive accuracy was measured by 10-fold cross-validation. We evaluated the methods' predictive accuracy by using the Geometric Mean (GM) of sensitivity and specificity. GM takes into account the balance between the sensitivity and specificity of the classifier. Sensitivity (or true positive rate) is the proportion of positive class instances correctly predicted as positive, whereas specificity (or true negative rate) is the proportion of negative class instances correctly predicted as negative [18].

To determine whether the differences in performance are statistically significant, we ran the Friedman test and the Holm post-hoc test [17], as recommended by Demsar [7]. First, the Friedman test was executed with the null hypothesis that the performances of all methods are equivalent. The alternative hypothesis is that there is a difference between the results of all methods as a whole, without identifying specific pairs of methods with significantly different results. If the null hypothesis is rejected, we run the Holm post-hoc test (which corrects for multiple hypothesis testing) to compare the results of the best method overall against each of the other methods. Both the Friedman and Holm test were used at the 0.05 significance level in all our experiments.

Table 4.1 reports the GM results for the proposed methods. Each row shows the results for a different dataset. Columns 3 to 5 show the results of the three methods using Naïve Bayes and the last three columns show the results of the same three methods using 1-NN.

The last two rows of the table show, for each method, the average rank (Avg. Rank) and the number of wins (#Win). The lower the Avg. Rank, the better (higher) the GM value. Note that the Avg. Rank and #Win values for the 3 methods are computed separately for each of the two base classifiers (NB and 1-NN). For each base classifier, the

Table 4.1: Comparing RPV, GA-HFS-SHE and GA-HFS-CbHE methods for feature selection in hierarchical feature spaces with Naïve Bayes and 1-NN as base classifiers in terms of GM (in %).

		Naïve Bayes			1-NN		
			GA-HFS			GA-HFS	
	Dataset	RPV	SHE	CbHE	RPV	SHE	CbHE
C.elegans	BP	65.72	**67.20**	64.30	60.09	66.79	**68.53**
	CC	63.62	66.46	**68.00**	63.04	66.49	**66.88**
	MF	56.69	**62.36**	62.20	44.40	59.70	**63.81**
	BP.MF	65.52	**68.58**	65.60	62.84	68.45	**70.17**
	CC.MF	66.77	**67.57**	65.60	57.81	65.85	**70.76**
	BP.CC.MF	66.50	**68.41**	65.70	60.95	70.96	**71.60**
D.melanogaster	BP	55.45	66.53	**67.60**	59.63	71.94	**74.52**
	CC	**74.44**	71.76	73.70	68.91	72.20	**74.46**
	MF	**67.17**	60.93	60.30	**70.53**	63.55	67.50
	BP.MF	60.15	**63.55**	62.80	63.17	68.80	**71.39**
	CC.MF	**70.81**	65.02	67.40	66.63	**71.55**	70.36
	BP.CC.MF	**66.47**	66.18	66.44	63.13	70.59	**71.31**
M.musculus	BP	68.60	**70.41**	69.60	59.47	72.81	**74.86**
	CC	69.43	67.69	**69.70**	54.58	**69.88**	67.91
	MF	67.93	68.44	**70.30**	66.12	68.41	**69.67**
	BP.MF	69.92	71.06	**71.30**	63.00	74.37	**74.48**
	CC.MF	67.26	67.65	**69.20**	64.02	71.83	**72.38**
	BP.CC.MF	71.22	**73.76**	71.80	62.94	75.25	**75.46**
S.cerevisiae	BP	61.63	70.03	**70.60**	57.17	67.71	**68.52**
	CC	59.89	57.56	**61.40**	38.82	47.22	**61.46**
	MF	**54.61**	41.96	44.90	34.03	45.27	**59.08**
	BP.MF	63.13	69.49	**69.50**	65.76	67.21	**68.09**
	CC.MF	**66.23**	54.95	58.90	46.52	60.46	**71.48**
	BP.CC.MF	58.44	68.38	**68.60**	60.95	**67.67**	67.61
	Avg. Rank	2.29	1.96	**1.75**	2.92	1.92	**1.17**
	#Wins	6.0	8.0	**10.0**	1.0	3.0	**20.0**

Naïve Bayes: {GA-HFS-CbHE} $\succ$ {RPV}
1-NN: {GA-HFS-CbHE} $\succ$ {RPV and GA-HFS-SHE}

highest GM value for each dataset is highlighted in bold type. In the row right below Table 4.1, the symbol $\succ$ represents a statistically significant difference between one or more methods, such that $\{a\} \succ \{b, c\}$ means that a is significantly better than b and c.

In Table 4.1, we present the results of RPV, GA-HFS-SHE and GA-HFS-CbHE. The presented results show that GA-HFS-CbHE achieved the best average rank and the highest number of wins for Naïve Bayes and 1-NN as base classifiers. Friedman test detected a significant difference among the methods for both base classifiers and the Holm test showed the following results. For NB, GA-HFS-CbHE results are statistically significantly better than those presented by RPV. Whilst there is no statistically significant differences between GA-HFS-CbHE and GA-HFS-SHE. For 1-NN, GA-HFS-CbHE achieved the best predictive performance on 20 out of 24 datasets, followed by GA-HFS-SHE with 3 wins and RPV with 1 win. Also, GA-HFS-CbHE obtained the smallest average ranking (1.17). These results are statistically significantly better than both GA-HFS-SHE and RPV. These results demonstrate that, in overall, GA-HFS-CbHE obtained the best predictive performance among the proposed methods for feature selection in hierarchical feature spaces.

Even though requiring to run a parameter tuning procedure in order to find the most suitable set of parameters, we have demonstrated that GA-based methods are robust, obtaining the best predictive performance overall. However, we argue that even though not achieving the best predictive performance, the lazy RPV method should not be put aside. RPV may be helpful when there is not enough data to calibrate the GA's parameters. Also, RPV selects a subset of features to each instance being classified, which could help the domain specialist to analyse each individual prediction.

4.2 Comparing the Predictive Performance of the Best Proposed Feature Selection Methods

In this thesis, we present novel feature selection methods for hierarchical feature spaces (Gene Ontology features) and uncertain feature spaces (PPI features). The novel feature selection methods introduced in Chapter 2 were evaluated on ageing datasets formed by GO features only. While, the novel feature selection method LFSUF introduced in Chapter 3 was evaluated on ageing datasets formed by PPI features alone, and GO and PPI features together. In the second case, the feature selection was employed only on PPI features, while the GO features were kept untouchable. So, in this section, we present an experimental evaluation to understand the performance of applying feature selection methods for both hierarchical GO features and uncertain PPI features at the same time. The goal of this evaluation is to verify whether applying feature selection methods that explores both the hierarchical and uncertain features present in datasets formed by GO and PPI is better than applying feature selection in PPI or GO alone.

In order to do that, we propose a new method called Hybrid, a feature selection method that copes with hierarchical and uncertain feature spaces at the same time. Hybrid uses two feature selection methods, one feature selection method for hierarchical features applied on hierarchical features (e.g., GO features in our case) and another feature selection method for uncertain features applied on uncertain features (e.g., PPI features).

Our experimental evaluation was performed by the comparison of the following feature selection methods:

- Genetic Algorithm for Hierarchical Feature Selection (GA-HFS with CbHE mutation operator) [Section 2.3.2, Appendix B], the best method for hierarchical feature selection on GO features.
- Lazy Feature Selection for Uncertain Features (LFSUF) [Section 3.3.2, Appendix D], it uses all GO features and apply feature selection only on the subset of PPI features.
- Hybrid: a feature selection method applying GA-HFS-CbHE to the subset of hierarchical GO features and LFSUF to the subset of uncertain PPI features.

We followed the same experimental settings described in the previous section. Table 4.2 follows the same format of Table 4.1 and presents the results of the comparison between the GA-HFS-CbHE, LFSUF and Hybrid (a method combining GA-HFS-CbHE and LFSUF) methods.

Results show that, for NB, Hybrid achieved the lower average ranking and the best number of wins, being the best method on 14 out of 24 datasets, followed by LFSUF that is the best method on 9 datasets. The worst method in this comparison is the GA-HFS-CbHE, being the best method only once. Also, for NB, the Hybrid method presented statistically superior results than GA-HFS-CbHE, while there is no statistical difference when the Hybrid method is compared to LFSUF. The Hybrid method was also the best method for the 1-NN classifier. It achieved both the best number of wins and average ranking. Hybrid was the best method in 16 out of 24 datasets. These results are statistically significantly superior to both LFSUF and GA-HFS-CbHE. Overall, these results demonstrate that the Hybrid method achieved the best predictive performance among the proposed methods. So, these results showed that we can achieve good predictive results employing feature selection methods tailored for each type of features (hierarchical and uncertain) present in ageing datasets. Particularly, these results were achieved when a GA-HFS-CbHE was used in the hierarchical part of the dataset (formed by GO features) and LFSUF was applied in the uncertain part of the dataset (PPI features).

The running time of these methods should also be taken into account. GA-HFS-CbHE is an eager feature selection method, which means that it selects a single subset of features in the preprocessing step. On the other hand, LFSUF selects a suitable subset of feature

Table 4.2: Comparing GA-HFS-CbHE, LFSUF and Hybrid methods by using Naïve Bayes and 1-NN as base classifiers in terms of GM (in %).

		Naïve Bayes			1-NN		
	Dataset	GA-HFS-CbHE	LFSUF	Hybrid	GA-HFS-CbHE	LFSUF	Hybrid
C.elegans	BP	64.30	69.20	**70.48**	68.53	67.12	**73.33**
	CC	68.00	71.09	**73.43**	66.88	68.31	**78.66**
	MF	62.20	**70.40**	64.68	63.81	69.13	**70.54**
	BP.MF	65.60	**70.04**	69.71	70.17	68.62	**76.24**
	CC.MF	65.60	**72.17**	70.92	70.76	68.94	**73.28**
	BP.CC.MF	65.70	**70.68**	69.18	71.60	68.59	**73.21**
D.melanogaster	BP	67.60	59.81	**71.92**	**74.52**	63.69	70.47
	CC	73.70	69.90	**76.20**	74.46	**76.29**	73.14
	MF	60.30	62.66	**71.59**	67.50	64.23	**75.66**
	BP.MF	62.80	64.35	**71.35**	71.39	68.74	**71.45**
	CC.MF	67.40	68.90	**71.59**	70.36	66.42	**71.10**
	BP.CC.MF	66.44	65.57	**72.45**	**71.31**	70.95	69.52
M.musculus	BP	69.60	71.18	**73.46**	**74.86**	72.80	73.70
	CC	**69.70**	69.07	69.39	67.91	68.06	**73.16**
	MF	70.30	70.27	**72.28**	69.67	**75.04**	72.55
	BP.MF	71.30	**73.80**	72.35	**74.48**	74.24	71.65
	CC.MF	69.20	71.13	**71.22**	72.38	**77.10**	73.25
	BP.CC.MF	71.80	72.00	**76.17**	**75.46**	72.84	74.92
S.cerevisiae	BP	70.60	**74.57**	74.22	68.52	73.67	**75.41**
	CC	61.40	**73.52**	72.84	61.46	62.27	**64.21**
	MF	44.90	71.31	**71.39**	59.08	67.72	**70.03**
	BP.MF	69.50	73.53	**73.87**	68.09	69.94	**74.97**
	CC.MF	58.90	**73.01**	72.33	71.48	62.89	**72.24**
	BP.CC.MF	68.60	**73.88**	72.29	67.61	71.23	**74.98**
	Avg. Rank	2.75	1.83	**1.42**	2.21	2.33	**1.46**
	#Wins	1	9	**14**	5	3	**16**

Naïve Bayes: {Hybrid} ≻ {GA-HFS-CbHE}
1-NN: {Hybrid} ≻ {GA-HFS-CbHE and LFSUF}

to classify each new instance in a lazy fashion. So, GA-HFS-CbHE takes longer to train in a new dataset and takes a small amount of time in the prediction step. Conversely, LFSUF does the majority of work in the prediction step, taking longer than GA-HFS-CbHE to select a subset of features. However, the time taken is not an issue. Also, note that the time taken to select a feature subset using the Hybrid method is a combination of both GA-HFS-CbHE and LFSUF methods.

We should remember that there could be cases where PPI features are not available since they have not been annotated yet. So, the use of GO features alone should not be discarded, since they can be useful in scenarios with no annotated PPIs. However, when PPI features are available, the use of the Hybrid method (composed of GA-HFS-CbHE and LFSUF) is recommended.

Chapter 5

Conclusions and Future Work

Ageing research focused on understanding the biological mechanisms that contribute to the organisms' ageing process. A better comprehension of these mechanisms could lead to the development of better medicines or medical treatments that improve the lifespan of an organism. In this thesis, we are interested in automatically predicting ageing-related genes into two classes: Pro-longevity and Anti-longevity. Genes are described in terms of Gene Ontology (GO) features and Protein-Protein Interaction (PPI) features. However, the use of such features is not straight-forward. GO features are hierarchically organized, while the information in PPI features is uncertain. Feature selection methods have been successfully used to improve the classification's predictive accuracy. However, only few methods are capable of exploring the information available in hierarchical and uncertain feature spaces.

The main objective of the thesis was to investigate and propose feature selection methods capable of exploring the hierarchical feature structure (in Gene Ontology features) and the uncertain feature values (characteristic found in Protein-Protein Interaction (PPI) features). Additionally, we also present a novel classification method specially designed to handle uncertain feature values.

In order to exploit the hierarchical structure present in GO features, we proposed two novel feature selection methods. The first one, a lazy feature selection method named Select Relevant Positive Feature Values (RPV), is based on the hypothesis that positive feature values, which are present to a small extent in each instance, provide more meaningful and accurate information to the classification process. The second method follows a wrapper approach, and is based on a genetic algorithm (GA) search. We proposed two novel biased mutation operators: (i) Simple Hierarchical Elimination (SHE) mutation, and (ii) Correlation-based Hierarchical Elimination (CbHE) mutation. Both methods were tailored to deal with the redundant features present in hierarchical feature spaces.

The experimental evaluation showed that the proposed hierarchical feature selection methods achieved superior predictive performance when compared against traditional (such as the CFS and ReliefF) and state-of-the-art hierarchical feature selection methods (such as HIP, MR and SHSEL methods), and also selecting a smaller subset of features. Furthermore, we show that GA-HFS with CbHE mutation operator overperformed the RPV method using both Naïve bayes and 1-NN as base classifiers in ageing datasets.

In order to explore datasets containing Protein-Protein Interaction (PPI) features with uncertain values (i.e., feature values represented by a confidence score), we introduced two strategies. Firstly, by employing a lazy feature selection method, named Lazy Feature Selection for Uncertain Features (LFSUF) method, based on the hypothesis that, for a given instance, a feature with both a high confidence score and great correlation with the class has better class-discrimination power, since it has a strong evidence of being present in the current instance. Secondly, we proposed a novel distance measure capable of handle uncertain values, this method is used as the distance measure of a nearest neighbour classifier.

Our experiments showed that LFSUF achieved a good predictive performance in uncertain feature spaces using datasets combining GO and PPI features and on PPI features alone, employing the Naïve Bayes and 1-NN (with the proposed Jaccard distance measure) as base classifiers. We also show, for the first time, that using PPI features along with GO features is beneficial to the classification task of ageing-related genes.

Additionally, in this thesis, we explore the use of feature selection methods for hierarchical feature spaces (GO features) and uncertain feature spaces (PPI features) when they are applied together in the same dataset. So, we proposed a Hybrid method that uses the best feature selection method for each type of feature. More specifically, the Hybrid method applies a GA-HFS-CbHE in the subset of GO features and LFSUF in the subset of PPI features. Our experiments showed that the Hybrid method achieved the best predictive performance, overall, using the Naïve Bayes and 1-NN as classifiers. So, when both GO and PPI features are available, the use of a specific feature selection method for each type of feature is recommended.

It is worth mentioning that the feature selection methods for hierarchical and uncertain feature spaces proposed in this thesis can be used in any application domain. Neither the number of features nor the number of instances are limitations in the application of such methods. So, these methods can be applied to any dataset with the following characteristics: (i) binary features and hierarchical feature structures organised in directed acyclic graphs (DAG) for the hierarchical feature selection methods; and, (ii) features with numeric values ranging from 0 to 1, representing the certainty of a given feature annotation. Figure 5.1 illustrates the decision process of selecting the adequate feature selection method for a given problem.

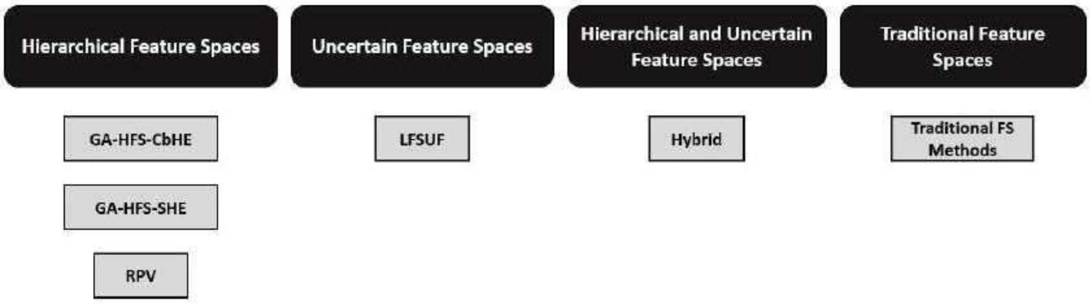

Hierarchical Feature Spaces
Uncertain Feature Spaces
Hierarchical and Uncertain Feature Spaces
Traditional Feature Spaces
GA-HFS-CbHE
GA-HFS-SHE
RPV
LFSUF
Hybrid
Traditional FS Methods

Figure 5.1: Figure describing which feature selection method should be used according to the feature space type.

In this thesis, we have proposed novel feature selection methods for exploiting the structures of hierarchical feature spaces and the uncertain values in uncertain feature spaces. We have done that by employing agnostic feature selection methods, in the sense that a method designed to handle hierarchical features does not share any information with the method capable of exploring uncertain features. So, as a feature work, we would like to understand how feature selection for these two types of feature can work together to improve even more the predictive performance of the ageing problem.

We have demonstrated the effectiveness of the proposed feature section methods in the prediction of ageing-related genes. However, the analysis of the selected features by a domain specialist is an essential task that should be performed. This task would guide us toward understanding the real biological meaning of the features currently being used in the classification task. We intend to address this in future work.

Also, we are interested in exploring the structure present in other types of features. For example, we have demonstrated that using Gene Ontology (GO) and Protein-Protein Interaction (PPI) features improve the predictive accuracy of classifiers for ageing-related genes. We are interested in including more feature types such as Kyoto Encyclopedia of Genes and Genomes (KEGG) [21] features and Motif features [9], and exploring the structure of these features to improve the predictive accuracy of ageing-related genes.

References

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

[2] Alberts, B.; Johnson, A.; Lewis, J. *Molecular Biology of the Cell*, 6nd ed. Garland Science, 2014.

[3] Consortium, T. G. Gene ontology: Tool for the unification of biology. *Nature Genetics 25*, 1 (2000), 25–29.

[4] Cover, T. M.; Thomas, J. A. *Elements of Information Theory*, second ed. Wiley-Interscience, 2006.

[5] da Silva, P. N.; Plastino, A.; Freitas, A. A. A novel genetic algorithm for feature selection in hierarchical feature spaces. In *Proceedings of the 2018 SIAM International Conference on Data Mining* (2018), SIAM, pp. 738–746.

[6] De Las Rivas, J.; Fontanillo, C. Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Computational Biology 6*, 6 (2010).

[7] Demsar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7* (2006), 1–30.

[8] Doquire, G.; Verleysen, M. Feature selection with mutual information for uncertain data. In *Proceedins of DaWaK* (2011), pp. 330–341.

[9] Drawid, A.; Gerstein, M. A bayesian system integrating expression data with sequence patterns for localizing proteins: Comprehensive application to the yeast genome. *Journal of Molecular Biology 201* (2000), 1059–1075.

[10] Fabris, F.; aes, J. P. M.; Freitas, A. A. A review of supervised machine learning applied to ageing research. *Biogerontology 18*, 2 (2017), 171–188.

[11] Fellenberg, M.; Albermann, K.; Zollner, A.; Mewes, H. W.; Hani, J. Integrative analysis of protein interaction data. In *International Int. Conf. Intell. Syst. Mol. Biol.* (2000), pp. 152–161.

[12] Fields, S.; Johnston, M. Whither model organism research? *Science 307* (2005).

[13] Hall, M. Correlation-based feature selection for discrete and numeric class machine learning. In *17th International Conference on Machine Learning (ICML)* (2000), pp. 359–366.

[14] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. The weka data mining software: an update. *ACM SIGKDD Exploration Newsletter 11*, 1 (2009), 10–18.

[15] Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2011.

[16] Hermjakob, H.; Montecchi-Palazzi, L.; Lewington, C.; Mudali, S.; Kerrien, S.; Orchard, S.; Vingron, M.; Roechert, B.; Roepstorff, P.; Valencia, A.; Margalit, H.; amd A Bairoch, J. A.; Cesareni, G.; Sherman, D.; Apweiler, R. Intact - an open source molecular interaction database. *Nucleic Acids Res 32* (2004), 452–455.

[17] Holm, S. A simple sequential rejective method procedure. *Scandinavian Journal of Statistics 6* (1979), 65–70.

[18] Japkowicz, N.; Shah, M. *Evaluating Learning Algorithms: A Classification Perspective.* Cambridge University Press, 2011.

[19] Jeong, Y.; Myaeng, S.-H. Feature selection using a semantic hierarchy for event recognition and type classification. In *6th Intl. Joint Conf. on NLP (IJCNLP)* (2013), pp. 136–144.

[20] Kaletsky, R.; Murphy, C. T. The role of insulin igf-like signaling in c. elegans longevity and aging. *Disease Models and Mechanisms 3*, 7 (2010), 415–419.

[21] Kanehisa, M.; Goto, S. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research 28* (2000), 27–30.

[22] Kirkwood, T. B. L.; Austad, S. N. Why do we age? *Nature 408*, 6809 (2000).

[23] Kong, X.; Yu, P. S.; Wang, X.; Ragin, A. B. Discriminative feature selection for uncertain graph classification. In *2013 SIAM International Conference on Data Mining (SDM)* (2013), pp. 82–93.

[24] Kononenko, I. Estimating attributes: Analysis and extensions of relief. In *ECML* (1994), F. Bergadano and L. D. Raedt, Eds., vol. 784, Springer, pp. 171–182.

[25] Liu, H.; Motoda, H. *Computational Methods of Feature Selection.* Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2008.

[26] Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining.* Springer, 2012.

[27] Liu, H.; Setiono, R. A probabilistic approach to feature selection: A filter solution. In *13th International Conference on Machine Learning (ICML)* (1996), pp. 319–327.

[28] Lopes-Ibanez, M.; Dubois-Lacoste, J.; Caceres, L. P.; Birattari, M.; Stutzle, T. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives 3* (2016), 43–58.

[29] Lu, S.; Ye, Y.; Tsui, R.; Su, H.; Rexit, R.; Wesaratchakit, S.; Liu, X.; Hwa, R. Domain ontology-based feature reduction for high dimensional drug data and its application to 30-day heart failure readmission prediction. In *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing* (2013), pp. 478–484.

[30] Magalhães, J. P.; Bukovsky, A.; Lehmann, G.; Costa, J.; Li, Y.; Fraifeld, V.; Church, G. M. The human ageing genomic resources: Online databases and tools for biogerontologistis. *Ageing Cell 8*, 1 (2009), 65–72.

[31] Martire, I.; da Silva, P. N.; Plastino, A.; Fabris, F.; Freitas, A. A. A novel probabilistic jaccard distance measure for classification of sparse and uncertain data. In *Proceedings of Symposium on Knowledge Discovery, Mining and Learning (KDMiLe)* (2017), pp. 81–88.

[32] McDonald, R. B. *Biology of Aging*, 2nd ed. Garland Science, 2019.

[33] Orii, N.; Ganapathiraju, M. K. Wiki-pi: A web-server of annotated human protein-protein interactions to aid in discovery of protein function. *PLoS ONE 7*, 1 (2012).

[34] Oughtred, R.; Stark, C.; Breitkreutz, B. J.; Rust, J.; Boucher, L.; Chang, C.; Kolas, N.; O'Donnell, L.; Leung, G.; McAdam, R.; Zhang, F.; Dolma, S.; Willems, A.; Coulombe-Huntington, J.; Chatr-Aryamontri, A.; Dolinski, K.; Tyers, M. The biogrid interaction database: 2019 update. *Nucleic Acids Res 47*, D1 (2019), 529–541.

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

[36] Promislow, D. E. L. Protein networks, pleiotropy and the evolution of senescence. *Proceedings. Biological Sciences 271*, 1545 (2004), 1225–1234.

[37] Raamsdonk, J. M. V. Mechanisms underlying longevity: A genetic switch model of aging. *Experimental Gerontology 107* (2018), 136–139.

[38] Reeves, C. R. Genetic algorithms. *Handbook of Metaheuristics 146* (2010), 109–139.

[39] Ristoski, P.; Paulheim, H. Feature selection in hierarchical feature spaces. In *DS 2014* (2014), S. Dzeroski, P. Panov, D. Docev, and L. Todorovski, Eds., vol. 8777 of *LNCS*, Springer, pp. 288–300.

[40] Schwanhausser, B.; Busse, D.; Li, N.; Dittmar, G.; Schuchhardt, J.; Wolf, J.; Chen, W.; Selbach, M. Global quantification of mammalian gene expression control. *Nature 473* (2011), 337–342.

[41] Stencil, C.; Waltz, D. Toward memory-based reasoning. *Commun ACM 29*, 12 (1986), 1213–1228.

[42] Szklarczyk, D.; Franceschini, A.; Wyder, S.; Forslund, K.; Heller, D.; Huerta-Cepas, J.; Simonovic, M.; Roth, A.; Santos, A.; Tsafou, K. P.; Kuhn, M.; Bork, P.; Jensen, L. J.; von Mering, C. String v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res 43* (2015), 47–52.

[43] Szklarczyk, D.; Gable, A. L.; Lyon, D.; Wyder, S.; Huerta-Cepas, J.; Simonovic, M.; Doncheva, N. T.; Morris, J. H.; Bork, P.; Jensen, L. J.; von Mering, C. String v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res 47* (2095), 607–613.

[44] Wan, C. *Hierarchical Feature Selection for Knowledge Discovery: application of data mining to the biology of ageing.* Springer, 2019.

[45] Wan, C.; Freitas, A. A. Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegant genes based on bayesian classification methods. In *International Conference on Bioinformatics and Biomedicine (BIBM)* (2013), pp. 373–380.

[46] Wan, C.; Freitas, A. A. Two methods for constructing a gene ontology-based feature network for a bayesian network classifier and applications to datasets of ageing-related genes. In *6th ACM Conf. on Bioinfo., Comp. Biology and Health Informatics (BCB)* (2015), pp. 27–36.

[47] Wan, C.; Freitas, A. A. An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features. *Artif. Intel. Review 50*, 2 (2018), 201–240.

[48] Wan, C.; Freitas, A. A.; Magalhães, J. P. 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 (2015), 262–275.

[49] Wieser, D.; Papatheodorou, I.; Ziehm, M.; Thornton, J. M. Computational biology for ageing. *Philosophical Transactions of the Royal Society B: Biological Sciences 366*, 1561 (2011), 51–63.

[50] Zaki, M. J.; Meira Jr., W. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* Cambridge University Press, 2014.

APPENDIX A – da Silva, P.N., Plastino, A., Freitas, A.A. “Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method” - Submitted to Applied Intelligence

Noname manuscript No.
(will be inserted by the editor)

Prioritizing Positive Feature Values: a New Hierarchical Feature Selection Method

Pablo Nascimento da Silva · Alexandre Plastino · Alex A. Freitas

Received: date / Accepted: date

Abstract In this work, we address the problem of feature selection for the classification task in hierarchical and sparse feature spaces, which characterise many real-world applications nowadays. A binary feature space is deemed hierarchical when its binary features are related via generalization-specialization relationships, and is considered sparse when in general the instances contain much fewer "positive" than "negative" feature values. In any given instance, a feature value is deemed positive (negative) when the property associated with the feature has been (has not been) observed for that instance. Although there are many methods for the traditional feature selection problem in the literature, the proper treatment to hierarchical feature structures, is still a challenge. Hence, we introduce a novel hierarchical feature selection method that follows the lazy learning paradigm – selecting a feature subset tailored for each instance in the test set. Our strategy prioritizes the selection of features with positive values, since they tend to be more informative – the presence of a relatively rare property is usually a piece of more relevant information than the absence of that property. Experiments on different application domains have shown that the proposed method outperforms previous hierarchical feature selection methods and also traditional methods in terms of predictive accuracy, selecting smaller feature subsets in general.

Keywords Hierarchical Feature Spaces · Feature Selection · Classification

P. N. da Silva (✉)
Institute of Computing, Fluminense Federal University (UFF), Niterói-RJ, Brazil
E-mail: psilva@ic.uff.br

A. Plastino
Institute of Computing, Fluminense Federal University (UFF), Niterói-RJ, Brazil
E-mail: plastino@ic.uff.br

A. A. Freitas
School of Computing, University of Kent, UK
E-mail: a.a.freitas@kent.ac.uk

1 Introduction

The classification task is one of the most relevant types of supervised learning in the knowledge discovery scenario [9]. A previously trained classification model automatically assigns a class label to an instance, based on the values of its features. In many important real-world problems, each instance in the dataset can be described as a binary feature vector, such that each feature takes either a "positive" or a "negative" value, indicating the presence or the absence of a property, respectively, in the object being classified. It should be noted that in this scenario, intuitively positive values are more informative than negative values in general. After all, a positive feature value has a clear and well-defined meaning, whilst the negative value of a feature represents very vague information, in the sense that it just tell us that the object being classified does not have a certain property, without providing any clue about the object's properties. Therefore, in this work we prioritize the selection of positive feature values over negative feature values, when learning classification models.

More specifically, this work addresses hierarchical feature spaces, where binary features are related via generalization-specialization relationships. In addition, the addressed feature spaces are sparse, i.e., in general the instances contain much fewer positive than negative feature values. In a generalization-specialization hierarchy, also known as "IS-A" hierarchy, for any given instance t, if a feature x has positive value in t, denoted $(x = 1)$, then all ancestors of x in the feature hierarchy also have positive value in t. In contrast, if a feature x has negative value in t, denoted $(x = 0)$, then all descendants of x in the feature hierarchy also have negative value in t.

Some examples of data commonly characterized by hierarchical and sparse feature spaces, where positive feature values are in general more informative than negative values, are text [12] and biological data [28, 29, 31], which are two of the most investigated types of machine learning applications.

For example, in the text classification problem, an article may be characterized by a set of tags describing its content. In this case, one general feature (e.g., News) may be associated with one or more specialized features (e.g., Economy, Politics and Sports). In addition, knowing that a document contains a certain word like Economics (positive feature value) provides us with clear information about the document's contents, whilst knowing that the document does not contain a certain word like Economics (negative feature value) provides us with much less information about the document's contents.

Similarly, in bioinformatics problems where each instance represents a gene, each gene may be associated with terms derived from an ontology of biological processes or functions. Hence, a general feature (e.g., biological process) would be the ancestor of more specific features (e.g., reproduction, metabolic process and biological regulation). In addition, a gene annotation indicating that the gene is involved in DNA repair (positive feature value) provides us with much more information than a lack of DNA repair annotation (negative feature value) for that gene.

Many important real-world datasets have a large number of features, many of which are not crucial for predicting the correct class. Some features can be redundant (highly correlated with each other) or irrelevant for predicting the class variable, decreasing the classifier's predictive accuracy, making the learning process slower, and reducing the comprehensibility of the results.

Feature selection methods have been successfully employed to cope with these problems. They aim at selecting a reduced subset of features to predict the target class, yet increasing the predictive accuracy of the classifier [16, 17]. Although many methods address this problem [7, 13, 15, 16, 17, 18, 22, 32], only few of them explore the hierarchical information in order to improve their effectiveness [12, 20, 24, 28, 29, 30, 31]. Existing hierarchical feature selection methods usually find a suitable subset of features by keeping those features with higher values of relevance and removing redundancy among hierarchically related features.

In this work, we focus on hierarchical and sparse feature spaces from different domains that share a singular characteristic; a positive feature value is always much more informative than a negative feature value, as briefly discussed earlier and discussed in more detail later. Despite this interesting characteristic of positive feature values, none of the previously proposed feature selection methods for hierarchical and sparse feature spaces has prioritized the selection of positive feature values. Hence, in this work, we hypothesize that the selection of positive feature values tends to increase the predictive accuracy of the classifier, and we propose feature selection methods prioritizing positive feature values.

The main contributions of this work are twofold. Firstly, we propose a novel lazy feature selection method for hierarchical and sparse feature spaces which relies on the higher relevance of features with positive values for the classification task. The basic idea of this method is to select, for each test instance, a subset with the most specific positive features in the hierarchy as well as its relevant ancestors. Secondly, we introduce a new lazy version of a relevance measure that evaluates the predictive relevance of a feature value for the current test instance.

The proposed feature feature selection method prioritizing positive feature values is evaluated in experiments with 33 datasets. These datasets are mainly from the area of bioinformatics, but they also include other types of application domains, in particular two datasets involving the classification of sports tweets, one dataset involving the classification of news headings, one dataset involving the classification of URLs, and finally one dataset classifying cities into categories of "liveability". The results of experiments with these diverse application domains show that the proposed hierarchical feature selection method outperforms both traditional feature selection methods and recent state-of-the-art hierarchical feature selection methods regarding the predictive accuracy, whilst also selecting smaller feature subsets.

This paper is organized as follows. Section 2 defines the hierarchical feature selection problem and briefly discusses feature selection methods. Section 3 reviews related work. Section 4 introduces the new relevance measure and the novel lazy restrictive hierarchical feature selection method. Section 5 presents experimental results. Section 6 presents conclusions and research directions.

2 Hierarchical Feature Selection for Classification

The classification problem can be defined as follows. Let $X = \{X_1, \ldots, X_d\}$ be a set of d predictive features and $L = \{l_1, \ldots, l_q\}$ be a set of q class labels, where $q \geq 2$. Let $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be a dataset with N instances, where

x_i corresponds to a vector $(x_{i1}, x_{i2}, \ldots, x_{id})$, for the i-th instance, which stores values for the d features in X and each $y_i \in L$ corresponds to a single target class. The goal of the classification task is to learn a classifier from D that, given an unlabelled instance $t = (x, ?)$, predicts its class label y.

The quality of the feature set has a huge impact on the predictive performance of classification algorithms [16, 17]. Feature selection methods aim at improving the predictive performance of the classifier by selecting a subset containing relevant and non-redundant features. Relevant features are those that are useful for predicting the target class variable, and non-redundant features are those that are not highly correlated with other features.

Feature selection methods can be categorized into embedded, wrapper and filter methods [16, 17]. Embedded methods are incorporated into the classification algorithm, selecting features during the construction of a classification model. Wrapper and filter methods are instead used in a data pre-processing step. Wrapper methods measure the relevance of a feature subset by evaluating the predictive accuracy of a classifier built using that subset. Hence, they select features tailored for the target classification algorithm, but they tend to be very time-consuming. By contrast, filter methods evaluate the predictive power of features in a generic way, by using a relevance measure that is independent of the target classification algorithm. Filter methods tend to be much faster and more scalable than wrapper methods. We focus on filter methods in this work. For an example of a wrapper approach see [3].

In some scenarios, the i-th instance is defined as a d-dimensional binary feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})$ with $x_{ij} \in \{0, 1\}$ for all $1 \leq j \leq d$. When the feature set X is hierarchically structured, we call it a hierarchical feature space, which can be represented as a Direct Acyclic Graph (DAG). In this DAG a vertex (node) represents a feature and an edge represents a generalization-specialization relationship between features. In this sense, an edge $(X_a \rightarrow X_b)$ indicates that X_a is a parent (immediate ancestor) of X_b and X_b is a child (immediate descendant) of X_a. More generally, a feature X_a is an ancestor (descendant) of a different feature X_b if and only if there is a sequence of edges leading from X_a to X_b (from X_b to X_a) in the feature DAG – or in the feature tree, if each node has either zero or one parent. The root node is the most general feature, while the leaf nodes are the most specific ones. Note that this structure produces a hierarchical redundancy among features, since a specific feature value logically implies the values of all its ancestors or descendants: all ancestors of positive-valued features have positive values and all descendants of negative-valued features have negative values.

For example, to classify an instance where the feature set is formed by Gene Ontology (GO) terms, if the instance is annotated with the GO term "multicellular organism reproduction", then that instance is considered annotated with the more general GO terms "reproduction" and "multicellular organism process". Conversely, if an instance is not annotated with the GO term "reproduction" (i.e., the feature "reproduction" has a negative value), then the instance is considered not to be annotated with the GO term "multicellular organism reproduction" (i.e., the child feature is guaranteed to have a negative value too, in that instance).

Hierarchical feature selection methods exploit characteristics of the feature DAG to improve the predictive accuracy. This is typically done by removing hierarchically redundant features [24, 31]. Technically speaking, in this work, note that, even though we are dealing with hierarchical features, the information about the hierarchical structure (represented by a feature DAG) is only used by the hierarchical feature selection methods. I.e., the hierarchical structure is used to enhance the feature selection process, helping to identify a better set of features to be selected. After the feature selection step, the data is treated as a "flat" dataset (i.e., the hierarchical structure is not considered anymore), then we can use traditional classification methods to make predictions.

Feature selection methods (as well as classification methods) are categorized as eager or lazy. Eager methods select a subset of features based on the training instances. Then, a model trained with the selected features is used to predict the class of any test instance. By contrast, lazy methods select a feature subset tailored for each test instance [1, 22], by observing the feature values (but not the class, of course) in that test instance. In this work, the main motivation to adopt the lazy learning approach is the ability to select a set of relevant positive feature values specifically tailored for each testing instance.

Based on these definitions, our proposed hierarchical method can be categorized as a filter feature selection method which follows the lazy learning paradigm.

3 Related Work

Traditional (non-hierarchical) feature selection methods, like the well-known eager Correlation-based Feature selection (CFS) [6, 7] and ReliefF [13] methods, can be used in hierarchical feature spaces by ignoring the hierarchical relationships among features. However, this is intuitively a sub-optimal approach. Hence, a few methods that directly exploit such hierarchical relationships to improve performance have been recently proposed, as follows.

SHSEL [24] is a hierarchical feature selection method that performs eager learning. SHSEL assumes that, if two features are directly hierarchically related (one is a parent of the other), they are usually highly correlated and tend to be similarly relevant for classification. Hence, for each pair of directly hierarchically related features, SHSEL removes the most specific feature if the correlation between them is higher than a user-defined threshold. Then, using only the remaining features, it keeps for each path in the hierarchy the features whose relevance is higher than the average relevance of features in that path. Moreover, Lu et al. proposed the Greedy Top-Down (GTD) search strategy [20], which selects the most relevant features in each path from each leaf to the root node in the hierarchy. Likewise, an eager learning hierarchical method called Tree-Based Feature Selection (TSEL) [12] has been used in the special case of tree-structured features. Previous work showed that SHSEL achieves better performance than TSEL and GTD [24]. For this reason, TSEL and GTD are no longer considered in this work.

Some hierarchical methods proposed in the literature are based on the lazy learning paradigm, such as the Select Hierarchical Information-Preserving Features (HIP) method [31], the Select Most Relevant Features (MR) method [31], and the hybrid Select Hierarchical Information-Preserving and Most Relevant Features (HIP-MR) method [28, 31]. Since the hybrid HIP-MR obtained worse results than its base methods HIP and MR in [28, 31], it is no longer considered. Next, we briefly describe HIP and MR.

The HIP method eliminates hierarchical redundancy by selecting only the "core" features in the current test instance – i.e., features whose values are non-redundant since they cannot be inferred from the values of other features. In other words, HIP selects the subset of the most specific positive-valued features (which imply their ancestors) and the most general negative-valued features (which imply their descendants). The values of the features selected by HIP for an instance imply the values of all other features for that instance, so it ensures that hierarchical redundancy is completely eliminated. However, HIP does not take into account the relevance of the selected features.

In a similar vein, the MR method not only eliminates hierarchical redundancy but also selects features with higher relevance. For each feature in the DAG, MR considers all paths between the feature and the root (for positive feature values) or between the feature and the leaves (for negative values). Then, the most relevant feature in each path is kept. However, unlike HIP, in general MR does not select all "core" features, i.e., it removes some hierarchically non-redundant features.

The proposed RPV method (described in Section 4) shares with HIP a certain focus on more specific positive feature values, but there are three important differences between these methods. First, HIP selects both positive and negative feature values, whereas RPV only selects positive feature values. Second, among positive feature values, HIP selects only the most specific ones; whilst RPV selects not only the most specific feature values, but also some of their relevant ancestors in the feature hierarchy. Third, RPV uses a new measure of feature value relevance (introduced in this paper), whilst HIP does not use any such relevance measure.

In this work, we compare our proposed method against the state-of-the-art hierarchical feature selection methods HIP, MR and SHSEL, as well as against the traditional (non-hierarchical) feature selection methods CFS and ReliefF.

It is worth noting that there are also other types of hierarchical feature selection methods, often discussed in the literature under the name of structured feature selection, as reviewed in [5, 19]. However, in general, those methods have been proposed for the regression task (using a variation of the Lasso method that produces a sparse linear model), rather than for the classification task addressed in this paper.

It is also important to highlight that the hierarchical feature selection task addressed in this paper should not be confused with the kind of hierarchical feature learning performed in deep learning processes. Deep neural networks involve hierarchical feature construction, where, during the training of the neural net, features are hierarchically learnt across the layers of the network [23]. On the other hand, in the problem discussed in this work, the hierarchy of features is predefined, and it is provided as an input to the feature selection algorithm. The point is not to learn or construct new features; the point is to select the best possible subset of features, among the original feature set, exploiting generalization-specialization information associated with the predefined feature hierarchy.

4 The Proposed Hierarchical Feature Selection Method

This section presents our new relevance measure and the new feature selection method for hierarchical and sparse feature spaces.

4.1 Lazy Feature Relevance Measure

In general, how to assess the relevance (or predictive power) of a feature plays an important role in the design of a good feature selection method. Many different functions have been proposed to cope with this issue, such as the Information Gain [2], the Mutual Information [27], the R measure [25, 28], etc.

The R measure, first proposed by [25], was adjusted by [28] to assess the predictive power of features in hierarchical feature selection. As shown in Equation 1, where k is the number of classes, the R measure calculates the relevance of a binary feature X based on the differences between the conditional probabilities of each class c_i given feature values x_1 and x_2.

$$R(X) = \sum_{i=1}^{k}[P(c_i|x_1) - P(c_i|x_2)]^2 \tag{1}$$

Note that Equation 1 is an eager relevance measure, but features may be useful or not depending on the feature values of the test instance being currently classified [22]. Our proposed feature selection method considers that taking into account the feature values (specifically positive values) of the current test instance may contribute to identifying a subset of high-quality features for that particular instance, in the spirit of lazy learning. For this reason, we propose a new feature relevance measure, named Lazy Relevance Measure (LazyR), which assesses the predictive power of a given feature X taking a specific value x in the current test instance. Defined in Equation 2, LazyR calculates the relevance of X with value x as a function of the sum of differences in the conditional probabilities of each class (c_i) given the specific feature value x and the class probability $\frac{1}{k}$ associated with a uniform distribution – ignoring the other values of X, since they do not occur in the current test instance. This measure has the highest value when the feature value x is perfectly correlated with one of the k classes, and presents the lowest value when the conditional probability of each class c_i is exactly $\frac{1}{k}$.

$$LazyR(X = x) = \sum_{i=1}^{k}\left[P(c_i|x) - \frac{1}{k}\right]^2 \tag{2}$$

The LazyR measure has some benefits over eager measures. Eager relevance measures (e.g., R and Information Gain) assess the relevance of all values of a feature to discriminate among class labels. In contrast, LazyR assesses the relevance of a specific feature value. For example, consider a feature A with positive and negative values, where the positive value discriminates well among class labels and the negative value does not. An eager relevance measure could assign a low score to feature A, leading to its removal. In contrast, our lazy relevance measure would keep A in the model if the instance being classified has a positive value for A, and remove it if the instance has a negative value, a principled data-driven decision.

4.2 The Proposed Lazy and Restrictive Hierarchical Feature Selection Method

We designed a new feature selection method for hierarchical and sparse feature spaces called Select Relevant Positive Feature Values (RPV). The intuition for this method is twofold. First, in sparse feature spaces, positive feature values are more informative and easier to interpret than negative values. That is, since positive feature values are quite rare, they provide more relevant and more meaningful information than negative values. For instance, in text mining, typically a document is described by features representing the presence (positive value) or absence (negative value) of words in that document, and the class indicates a document's subject. The presence of the word "teacher" is relevant for predicting that the document's class is "Education", but the absence of the word "teacher" is not relevant for classification nor meaningful, it is too broad information. Second, the generalization-specialization structure of hierarchical feature spaces creates hierarchical redundancy among features, which intuitively reduces predictive accuracy. RPV exploits generalization-specialization relationships in order to eliminate hierarchical redundancy, which should improve predictive accuracy.

More specifically, our method adopts the following ideas: (i) it relies on a restrictive selection approach, where selecting only positive feature values might increase the accuracy of the classifier; (ii) it tries to identify a specific subset of relevant positive features for each instance t in the test set – using the lazy paradigm; (iii) taking into account the hierarchy, it selects the most specialized positive feature values as well as those positive feature values whose relevance value is higher than (or equal to) the relevance of all its positive descendants.

We now show, theoretically, that Naïve Bayes – a classification algorithm used in related work [24, 28, 29, 31] and also employed in our experiments – tends to give larger influence to positive feature values than to negative feature values in sparse datasets, which is in agreement with the ideas behind the proposed feature selection method.

Consider the log-odds ratio form of Naïve Bayes (for binary classes c_1 and c_2):

$$ln\frac{P(c_1|X)}{P(c_2|X)} = ln\frac{P(c_1)}{P(c_2)} + \sum_{i=1}^{d} ln\frac{P(x_i|c_1)}{P(x_i|c_2)}, \tag{3}$$

which predicts class c_1 for the current instance if $ln\frac{P(c_1|X)}{P(c_2|X)} > 0$, and predicts class c_2 otherwise. The summation term of this formula can be divided into two parts: $\sum_{i+=1}^{d+} ln\frac{P(x_{i+}|c_1)}{P(x_{i+}|c_2)}$ and $\sum_{i-=1}^{d-} ln\frac{P(x_{i-}|c_1)}{P(x_{i-}|c_2)}$, where $i+$ and $i-$ index the set of positive and negative feature values in the current instance, respectively; $d+$ and $d-$ are the number of positive and negative feature values in the current instance, respectively; and $d- + d+ = d$. In the case of very sparse features, each term in the second summation (over the $d-$ negative feature values) will tend to zero. This is because, since the vast majority of instances take the negative value for a highly sparse feature, both the terms $P(x_{i-}|c_1)$ and $P(x_{i-}|c_2)$ will tend to have similar values (both will tend to be close to 1), and therefore each term $ln\frac{P(x_{i-}|c_1)}{P(x_{i-}|c_2)}$ will tend to be close to zero. I.e., negative feature values will have little influence in the Naïve Bayes formula. On the other hand, for positive feature values, the terms $P(x_{i+}|c_1)$ and $P(x_{i+}|c_2)$ will have quite different values in general, and so the summation of the terms $ln\frac{P(x_{i+}|c_1)}{P(x_{i+}|c_2)}$ over the $d+$ positive feature values will tend to be a large number, rather than close to zero. I.e., positive feature values tend to have a larger influence than negative feature values in the Naïve Bayes formula.

The RPV method works as follows. Given a test instance t, first, it evaluates the relevance of each feature in t. Then, it identifies the list of ancestors for each positive feature value, using the feature DAG. After that, RPV marks every negative feature value in t for removal. For each positive feature X_i in t, RPV evaluates each of its ancestors and marks for removal those whose relevance is lower than the relevance of X_i. At the end of the process, RPV removes every feature marked for removal, and the remaining features are used in the lazy classification of the current test instance.

Algorithm 1 describes how RPV works in detail. This algorithm produces as output a subset of features named *SelectedFeatSubSet*. In the initialization phase (lines 1 to 5), the ancestors and the relevance value (measured by LazyR) for each feature in the DAG are computed and stored into the respective *Ancestors* and *Relevance* arrays (indexed by the features' ids). Also, the *Status* array is initialized with the "Selected" value for all features.

Algorithm 1 Select Relevant Positive Feature Values (RPV)

```
Input : $D$ (training dataset), $t$ (test instance) and $DAG$ (feature hierarchy)
Output: a subset of features $SelectedFeatSubSet$
 1: for each feature $X_i$ in $DAG$ do
 2:     $Ancestors[X_i] \leftarrow$ list of ancestors of $X_i$ in the DAG
 3:     $Relevance[X_i] \leftarrow LazyR(X_i = positive)$ {computed using the training set $D$}
 4:     $Status[X_i] \leftarrow$ "Selected"
 5: end for
 6: for each feature $X_i$ in $DAG$ do
 7:     if $Value(X_i, t)$ is positive then
 8:         for each feature $A_j \in Ancestors[X_i]$ do
 9:             if $Relevance[A_j] < Relevance[X_i]$ then
10:                 $Status[A_j] \leftarrow$ "Removed"
11:             end if
12:         end for
13:     else
14:         $Status[X_i] \leftarrow$ "Removed" {since $Value(X_i, t)$ is negative}
15:     end if
16: end for
17: $SelectedFeatSubSet \leftarrow$ features with $Status$ set to "Selected"
18: return $SelectedFeatSubSet$
```

The main phase of RPV works as follows. In line 7, for each feature X_i in DAG, the function $Value(X_i, t)$ returns the value of X_i in the test instance t. If the returned value is positive, RPV looks at each ancestor A_j of X_i in the DAG and marks for removal (setting the *Status* flag) those with relevance value lower than the relevance of X_i (lines 8 to 12). In line 14, every feature with negative value in t is marked for removal, since negative values are much less informative than positive values, as discussed earlier. In lines 17 and 18, the feature subset *SelectedFeatSubSet* receives all features whose *Status* is still "Selected" and this subset is returned by the algorithm. Then, a lazy classifier is executed for test instance t using only the selected features. Note that, after initializing each feature's *Status* with "Selected", the *Status* of a feature can only be changed to "Removed" in lines 10 and 14, and once this change is made, that feature's *Status* is never set back to "Selected" by the algorithm. Hence, the result of the algorithm does not depend on the order in which the features are processed.

The RPV algorithm is executed for each test instance in a lazy learning fashion, but note that, in order to save time, the values of the *Ancestors* and *Relevance* arrays can be pre-computed in an eager fashion and stored to be accessed whenever a new instance needs to be classified.

Figure 1 illustrates how RPV works. In this figure, each vertex represents a feature, and the numbers on the right and left side of each node represent, respectively, the feature's value (1 for positive, 0 for negative) and the relevance of that feature value. After RPV's initialization phase, each feature in the DAG (denoted by letters A to N) is processed in turn. When A, B, D, E, F and I are processed, their *Status* will be set to "Removed", since their values are "0". When C (with value "1") is processed, RPV sets to "Removed" the *Status* of C's ancestors in the DAG whose relevance (LazyR) value is lower than C's relevance – i.e., G and N are marked for removal. When H is processed, L and N (ancestors with lower relevance than H) are marked for removal, and M will also be marked for removal when J is processed. After processing all features, the only ones selected (never marked for removal) are features C, K, H and J.

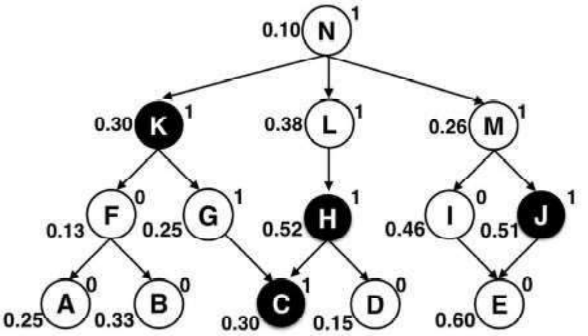


Fig. 1: Example of a feature DAG showing the subset of features selected by RPV.

The RPV method presents some appealing characteristics: (i) it selects only positive feature values, which are more informative than negative values; (ii) it uses a lazy relevance measure specifically adapted to assess the relevance of a feature value in the current test instance; (iii) since it selects only positive values, it tends to select fewer features than the other methods used in our experiments (as shown later); (iv) it selects the most specific positive feature values and some of their most relevant ancestors.

5 Computational Experiments

5.1 Datasets

In this work, the proposed method was evaluated on 33 distinct datasets, 28 from the bioinformatics domain and 5 from distinct classification domains.

Following the same methodology described in previous work [29, 31], we generated 28 datasets of ageing-related genes, involving the effect of genes on an organism's longevity. These datasets were created by integrating data from the Human

Ageing Genomic Resources (HAGR) GenAge database (version: Build 17) [21] and the Gene Ontology (GO) database (version: 2015-10-10) [26]. HAGR is a database of ageing- and longevity-associated genes in model organisms which provides ageing information for genes from four model organisms: *C. elegans* (worm), *D. melanogaster* (fly), *M. musculus* (mouse) and *S. cerevisiae* (yeast). The GO database provides information about three ontology types: biological process (BP), molecular function (MF) and cellular component (CC). Each ontology contains a separate set of GO terms (features), i.e., a distinct feature hierarchy (a DAG). So, for each of the 4 model organisms, we created 7 datasets, with 7 combinations of feature types (feature hierarchies), denoted by BP, CC, MF, BP.CC, BP.MF, CC.MF and BP.CC.MF.

Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term in the GO hierarchy and a binary class variable indicating if the instance is either positive ("pro-longevity" gene) or negative ("anti-longevity" gene) according to the HAGR database. In order to avoid overfitting, GO terms which occurred in less than three genes were discarded.

It is worth mentioning that the annotation of GO terms available in these data resources is error-prone, because it is based on a combination of biologists' expert knowledge and the results of biological experiments (which are inherently noisy).

The remaining 5 datasets, previously used in a related work [24], represent different classification tasks with features and hierarchies extracted from either the Open Directory Project[1] or DBpedia [14]. These datasets are described below.

- Tweets T and Tweets C: in these datasets, the task is to identify sports-related tweets, where each tweet can be either related to sports (positive class) or not related to sports (negative class). The hierarchy and features were generated by extracting types (in Tweets T) and categories (in Tweets C) from DBpedia.
- NY Daily: this dataset is a set of news headings augmented with DBpedia's types, where the classification task is to identify a sentiment variable (positive/negative).
- Stumbleupon (Std.upon): this dataset is formed by a set of URLs whose features are derived from the Open Directory Project. The classification task is to predict if a URL is either positive ("evergreen") or negative ("ephemeral").
- Cities: this dataset was generated from a list of the most and the least liveable cities augmented with DBpedia types. The task is to classify each city into low, medium and high liveability.

Information about the datasets is shown in Table 1. For each of the 4 model organisms, each of the 7 rows shows information about a specific dataset. The first column identifies the group of datasets for each of the four organism or the general-domain datasets. The second column shows the feature hierarchies used to build the bioinformatics datasets or, in the last five rows, the names of the datasets from general (non-bioinformatics) domains. The other columns show, respectively, the number of features (#features), the number of edges in the feature DAGs (#edges), the number of instances (#instances), the percentage of positive-class instances (% Pos), the percentage of negative-class instances (% Neg), and the percentage of positive feature values (% Pos values). In the last row, the class distribution (low, medium and high) for the dataset Cities is shown within columns 6 and 7.

[1] http://www.dmoz.com

Table 1: Detailed information about the datasets used in the experiments.

Group	Dataset	#features	#edges	#instances	% Pos	% Neg	%Pos values
C.elegans	BP	991	1707	657	34.40	65.60	4.50
	CC	178	277	484	36.36	63.64	6.49
	MF	263	331	504	37.70	62.30	5.07
	BP.CC	1169	1984	664	34.34	65.66	4.49
	BP.MF	1254	2038	663	34.24	65.76	4.35
	CC.MF	441	608	566	36.22	63.78	4.99
	BP.CC.MF	1432	2315	667	34.33	65.67	4.38
D.melanogaster	BP	800	1355	132	71.97	28.03	8.32
	CC	89	130	122	70.49	29.51	12.02
	MF	146	182	126	70.63	29.37	7.72
	BP.CC	889	1485	133	71.43	28.57	8.62
	BP.MF	945	1536	133	71.43	28.57	8.11
	CC.MF	234	311	130	70.77	29.23	9.28
	BP.CC.MF	1034	1666	133	71.43	28.57	8.44
M.musculus	BP	1333	2406	109	68.81	31.78	10.65
	CC	143	214	107	68.22	31.78	16.41
	MF	240	289	106	67.92	32.08	9.73
	BP.CC	1475	2619	109	68.81	31.19	11.21
	BP.MF	1572	2694	109	68.81	31.19	10.47
	CC.MF	382	501	109	68.81	31.19	12.07
	BP.CC.MF	1714	2906	109	68.81	31.19	10.97
S.cerevisae	BP	844	1511	331	13.29	86.71	5.35
	CC	145	230	331	13.29	86.71	9.04
	MF	221	277	331	13.29	86.71	5.73
	BP.CC	989	1741	331	13.29	86.71	6.04
	BP.MF	1065	1788	331	13.29	86.71	5.43
	CC.MF	366	507	331	13.29	86.71	7.44
	BP.CC.MF	1210	2018	331	13.29	86.71	5.98
General	Tweets T	4082	36019	1179	55.64	44.36	1.14
	Tweets C	10883	15189	1179	55.64	44.36	1.02
	NY Daily	5145	44152	1016	57.09	42.91	1.21
	Stb.upon	3976	12354	3020	45.36	54.64	1.17
	Cities	727	7051	212	18.40/50.00/31.60		3.31

5.2 Experimental Methodology

We implemented our RPV method and other methods used in this work within the open-source WEKA data mining tool [8]. The datasets used in the experiments and the program code of the RPV method are freely available on the web at (will be freely available on the web after the publication of the paper). The methods were evaluated on the 33 datasets described earlier. The lazy k-NN with Euclidean distance (with $k = 1$) and a lazy version of Naïve Bayes (NB) (both from WEKA) were used as classification algorithms for all evaluated feature selection methods, and the predictive accuracy was measured by 10-fold cross-validation. It is worth mentioning that Naïve Bayes has also been used in previous work on hierarchical feature selection [20, 24, 28, 30, 31], as well as k-NN [30, 31]. Besides, after conducting some preliminary experiments with the datasets used in this work and without employing any feature selection method, Naïve Bayes achieved the best classification performance, followed by 1-NN, when compared with other traditional classification algorithms, namely SVM, Random Forest and Decision Trees (C4.5).

As shown in Table 1, the majority of the datasets have imbalanced class distributions, so we evaluated the methods' predictive accuracy by using the Geometric Mean (GM) of sensitivity and specificity as well as the Area Under Precision-Recall

Curve (AUCPR) measures. The GM is defined in Equation 4, which was also used in [29, 30, 31].

$$GM = \sqrt{Sensitivity * Specificity} \tag{4}$$

GM takes into account the balance between the sensitivity and specificity of the classifier. Sensitivity (or true positive rate) is the proportion of positive class instances correctly predicted as positive, whereas specificity (or true negative rate) is the proportion of negative class instances correctly predicted as negative [11]. The AUCPR plots the precision of the classifier as a function of its recall, then the area under this curve is used to evaluate the classifier (the higher the better) [11].

To determine whether the differences in performance are statistically significant, we ran the Friedman test and the Holm post-hoc test [10], as recommended by Demsar [4]. First, the Friedman test was executed with the null hypothesis that the performances of all methods are equivalent. The alternative hypothesis is that there is a difference between the results of all methods as a whole, without identifying specific pairs of methods with significantly different results. If the null hypothesis is rejected, we run the Holm post-hoc test (which corrects for multiple hypothesis testing) to compare the results of the best method overall (RPV with the LazyR measure) against each of the other methods. Both the Friedman and Holm test were used at the 0.05 significance level in all our experiments.

5.3 Results

Tables 2, 3, 4 and 5 report the predictive accuracy results for two experiments: the first one (Tables 2 and 3) compares our proposed RPV method to baselines approaches, and the second one (Tables 4 and 5) evaluates RPV against state-of-the-art features selection methods. In these tables, each row shows the results for a different dataset. Columns 3 to 8 show the results of six evaluated methods (defined below) using Naïve Bayes and the last six columns show the results of the same six methods using 1-NN. Tables 2 and 4 report the AUCPR results, whilst Tables 3 and 5 report the GM results. For the Cities dataset (with 3 classes), these measures were computed by considering each class in turn as the positive class and averaging the results over the 3 classes.

The last two rows of each table show, for each method, the average rank (Avg. Rank) and the number of wins (#Win). The lower the Avg. Rank, the better (higher) the GM or AUCPR value. Note that the Avg. Rank and #Win values for the 6 methods are computed separately for each of the two base classifiers (NB and 1-NN). For each base classifier, the highest GM or AUCPR value for each dataset is highlighted in bold type. In the row right below Tables 2 to 5, the symbol $\succ$ represents a statistically significant difference between one or more methods, such that $\{a\} \succ \{b, c\}$ means that a is significantly better than b and c.

5.3.1 Comparison against baseline feature selection approaches

This first experiment aims to evaluate the effectiveness of the two main characteristics of the proposed RPV method, i.e., its focus on selecting only a subset of positive feature values and its new lazy relevance measure. Tables 2 and 3 report

Table 2: Comparing RPV with different feature relevance measures against baseline methods in terms of AUCPR – in %.

		Naïve Bayes						1-NN					
		Baseline			RPV			Baseline			RPV		
	Datasets	No FS	All-Neg	All-Pos	IG	R	LazyR	No FS	All-Neg	All-Pos	IG	R	LazyR
C.elegans	BP	55.1	53.1	54.6	55.7	55.3	**56.4**	48.0	41.8	48.3	49.4	47.4	**50.4**
	CC	56.3	**56.5**	55.1	53.1	54.5	54.3	49.4	53.5	54.5	53.2	53.9	**54.6**
	MF	50.2	47.8	50.5	50.6	51.4	**51.7**	47.6	49.9	**51.7**	**51.7**	50.5	50.1
	BP.CC	55.6	55.4	55.8	**56.8**	55.4	56.3	47.7	46.0	**50.9**	48.9	49.8	50.2
	BP.MF	53.6	53.0	53.2	53.8	54.0	**55.4**	45.8	46.7	48.7	49.4	**51.3**	50.1
	CC.MF	**54.8**	53.7	54.5	52.3	54.1	53.3	47.7	**57.0**	54.8	51.7	51.9	54.6
	BP.CC.MF	54.0	53.0	54.0	55.1	55.7	**56.6**	46.4	49.1	49.2	**50.3**	48.8	48.6
D.melanogaster	BP	83.1	80.2	83.4	**83.5**	82.2	82.5	78.2	76.2	80.0	**80.7**	80.2	79.3
	CC	87.6	84.0	87.8	88.5	89.8	**90.0**	79.6	81.9	85.2	83.5	**84.0**	83.4
	MF	81.9	79.7	81.9	82.0	**82.8**	81.1	79.3	80.0	81.9	82.1	82.2	**82.4**
	BP.CC	85.2	80.8	84.7	84.0	**86.5**	83.4	76.8	74.2	79.9	81.7	**82.7**	80.8
	BP.MF	84.7	80.7	85.1	84.8	**85.7**	83.3	76.7	77.0	80.2	81.4	**81.5**	80.4
	CC.MF	88.1	85.8	88.2	89.4	89.1	**89.7**	78.3	76.8	83.6	82.5	83.4	**84.7**
	BP.CC.MF	85.4	82.9	85.7	86.4	**87.1**	84.8	77.2	73.2	79.6	81.4	**81.7**	81.3
M.musculus	BP	82.5	82.1	84.4	**85.5**	85.1	85.2	77.1	75.9	76.0	**78.2**	77.7	77.6
	CC	84.5	82.2	86.0	**86.5**	85.9	84.4	74.1	69.6	75.1	77.8	**78.9**	76.6
	MF	**87.1**	83.9	86.5	85.8	86.3	85.6	77.6	74.1	**81.8**	78.7	79.2	79.1
	BP.CC	84.2	83.1	85.2	86.3	85.6	**88.6**	77.1	**78.4**	74.9	76.2	75.7	75.6
	BP.MF	81.7	82.3	84.4	85.6	85.4	**86.8**	77.2	73.3	78.8	**80.0**	78.8	78.2
	CC.MF	85.7	82.4	86.7	87.1	**87.2**	86.2	72.6	72.9	77.7	77.3	**80.6**	79.3
	BP.CC.MF	83.0	82.9	85.2	**87.6**	86.8	87.0	78.5	74.5	78.3	78.5	**78.6**	78.2
S.cerevisae	BP	45.6	43.2	40.2	42.8	38.3	**46.4**	29.2	25.9	32.8	**39.5**	35.5	36.4
	CC	34.0	32.5	33.9	34.6	31.6	**35.3**	30.6	31.7	34.5	36.9	37.3	**37.6**
	MF	26.8	20.4	**27.0**	20.8	20.7	25.2	28.9	26.3	32.0	**36.0**	34.3	35.8
	BP.CC	47.4	**48.0**	40.4	44.6	39.6	46.5	25.2	17.0	28.0	32.6	33.3	**34.5**
	BP.MF	41.8	42.8	35.8	41.1	36.2	**46.1**	25.3	21.6	35.3	40.4	35.2	**40.5**
	CC.MF	**33.9**	27.4	31.5	33.3	30.2	32.7	26.5	26.0	**36.8**	34.1	36.4	36.6
	BP.CC.MF	44.4	45.6	35.9	44.0	37.9	**46.2**	23.4	16.3	31.9	33.3	33.4	**34.3**
General	Tweets T	81.6	71.1	82.2	82.7	82.8	**83.3**	76.5	58.8	81.0	81.6	82.4	**82.6**
	Tweets C	98.3	95.4	98.3	98.4	**98.5**	**98.5**	94.5	90.2	96.6	96.9	97.0	**97.5**
	NY Daily	64.1	60.7	64.1	64.1	63.5	**64.8**	**60.4**	57.9	59.5	58.9	58.8	60.1
	Std.upon	77.8	75.7	78.0	77.9	77.7	**78.7**	71.6	**74.8**	74.1	74.1	74.2	74.4
	Cities	70.7	64.8	71.7	71.2	70.9	**74.1**	60.2	61.2	**69.5**	69.0	69.2	69.4
	Avg. Rank	3.7	5.2	3.5	2.8	3.3	**2.5**	5.0	5.2	3.2	2.8	2.5	**2.4**
	#Win	3.0	2.0	1.0	5.0	5.5	**16.5**	1.0	3.0	4.5	6.5	7.0	**10.0**

Naïve Bayes: {RPV-LazyR} ≻ {No FS, All-Pos, All-Neg and RPV-R}

1-NN: {RPV-LazyR} ≻ {No FS and All-Neg}

the AUCPR and GM results, respectively. We compare our RPV method using the LazyR relevance measure against two RPV versions with different relevance measures and three baseline methods. The first baseline is the base classifier using no feature selection method (No FS). I.e., it uses the full set of predictive features. We also implemented two baseline lazy non-hierarchical feature selection methods: one that selects all features with positive value in the current test instance (All-Pos); and another one that selects all features with negative values in the current test instance (All-Neg). Moreover, in order to evaluate the benefit of our proposed feature relevance measure (LazyR), we compare our RPV (which uses the LazyR measure) against two RPV versions. The first version uses the original eager relevance measure R (RPV-R), defined in Equation 1, and the second version uses the traditional Information Gain measure (RPV-IG).

Considering the results for the AUCPR measure in Table 2, RPV-LazyR obtained the best #Win and the best Avg. Rank values for both Naïve Bayes and

Table 3: Comparing RPV with different feature relevance measures against baseline methods in terms of GM – in %.

		Naïve Bayes						1-NN					
		Baseline			RPV			Baseline			RPV		
	Datasets	No FS	All-Neg	All-Pos	IG	R	LazyR	No FS	All-Neg	All-Pos	IG	R	LazyR
C.elegans	BP	62.0	0.0	59.4	60.5	60.9	**65.7**	58.4	20.6	**61.0**	60.9	59.3	60.1
	CC	**65.7**	39.9	65.5	63.6	65.3	63.6	59.9	62.9	**64.0**	62.1	63.2	63.0
	MF	57.6	44.2	**59.1**	57.0	54.1	56.7	**53.4**	24.4	45.4	46.2	44.6	44.4
	BP.CC	61.9	0.0	60.9	62.9	64.0	**67.7**	59.9	14.6	**62.4**	60.3	62.0	61.5
	BP.MF	61.9	0.0	59.8	61.7	61.8	**65.5**	58.0	11.5	62.7	63.0	**63.5**	62.8
	CC.MF	64.2	24.3	64.4	64.1	66.1	**66.8**	58.6	52.0	**60.7**	56.8	54.6	57.8
	BP.CC.MF	62.4	0.0	61.6	62.8	63.4	**66.5**	60.0	19.6	62.1	**64.4**	63.7	61.0
D.melanogaster	BP	**59.4**	0.0	51.4	53.8	54.6	55.5	58.8	18.1	**62.4**	58.6	56.8	59.6
	CC	66.7	0.0	75.6	**76.2**	75.7	74.4	**71.9**	50.3	71.0	69.5	69.5	68.9
	MF	58.0	0.0	67.0	63.8	63.3	**67.2**	51.3	0.0	70.1	70.5	70.5	**70.9**
	BP.CC	57.7	0.0	56.1	60.5	62.3	**64.5**	56.1	22.1	57.5	61.6	63.1	**64.1**
	BP.MF	57.3	0.0	58.8	56.4	**61.0**	60.2	54.5	0.0	61.8	59.9	**63.6**	63.2
	CC.MF	65.8	0.0	69.1	69.5	70.3	**70.8**	61.0	32.8	**68.8**	68.5	67.4	66.6
	BP.CC.MF	59.4	0.0	61.0	61.8	63.8	**66.5**	56.8	0.0	60.8	**63.9**	62.5	63.1
M.musculus	BP	59.1	25.7	66.3	**69.9**	68.4	67.9	**65.1**	33.8	57.9	61.6	62.1	59.5
	CC	64.1	28.1	68.3	69.0	66.4	**69.4**	55.0	37.0	54.1	56.1	**56.8**	54.6
	MF	63.5	57.6	65.9	65.3	64.7	**67.9**	61.3	42.1	**68.1**	63.9	61.8	66.1
	BP.CC	67.4	25.7	67.0	**69.3**	67.9	69.2	**64.3**	44.9	58.0	57.1	54.5	53.3
	BP.MF	64.9	25.7	69.6	70.5	**70.9**	69.9	62.8	48.7	61.0	62.5	**63.8**	63.0
	CC.MF	61.6	36.4	**68.6**	68.2	68.2	67.3	48.9	36.3	63.2	66.4	**67.2**	64.0
	BP.CC.MF	70.2	25.7	69.0	70.6	69.8	**71.2**	**65.5**	56.3	62.7	61.1	63.8	62.9
S.cerevisae	BP	61.5	0.0	54.7	56.2	52.4	**61.6**	54.7	0.0	57.3	**67.1**	60.1	57.2
	CC	57.6	0.0	59.3	58.9	59.2	**59.9**	**52.5**	0.0	38.7	39.0	39.0	38.8
	MF	34.2	0.0	**57.8**	54.2	44.1	54.6	**43.5**	0.0	34.0	34.2	34.3	34.0
	BP.CC	**63.1**	0.0	51.3	54.0	50.0	58.1	54.5	0.0	56.3	**57.8**	53.6	49.5
	BP.MF	62.1	0.0	53.5	55.4	55.7	**63.1**	49.8	0.0	**67.6**	67.4	62.3	65.8
	CC.MF	59.9	0.0	60.6	59.9	56.5	**66.2**	48.2	0.0	**50.5**	44.5	45.1	46.5
	BP.CC.MF	**62.8**	0.0	50.8	53.0	52.3	58.4	45.6	0.0	**62.2**	61.2	61.8	62.0
General	Tweets T	68.2	8.8	72.2	73.0	73.1	**73.6**	73.0	0.0	74.2	74.7	74.8	**75.1**
	Tweets C	87.6	0.0	94.5	94.8	**95.0**	94.8	91.2	59.1	**95.0**	94.5	94.4	**95.0**
	NY Daily	50.3	0.0	56.7	**57.1**	55.9	56.6	**53.0**	0.0	51.3	51.1	51.5	52.6
	Std.upon	68.7	15.1	70.6	70.6	**71.0**	70.7	70.6	**71.8**	71.0	70.9	70.7	71.1
	Cities	73.5	0.0	**73.6**	63.9	71.0	71.8	59.3	24.7	61.2	61.2	60.8	**61.4**
	Avg. Rank	3.7	6.0	3.4	3.1	3.0	**1.9**	3.7	5.8	**2.8**	3.0	2.9	2.9
	#Win	4.0	0.0	4.0	4.0	4.0	**17.0**	8.0	1.0	**10.5**	4.0	5.0	4.5

Naïve Bayes: {RPV-LazyR} $\succ$ {No FS, All-Pos, All-Neg, RPV-IG and RPV-R}

1-NN: {RPV-LazyR} $\succ$ {No FS and All-Neg}

1-NN. For Naïve Bayes, the Holm post-hoc test indicated that RPV-LazyR is significantly better than No FS, All-Pos, All-Neg and RPV-R. For 1-NN, the Holm post-hoc test indicated that RPV-LazyR is significantly better than No FS and All-Neg.

The GM results in Table 3 show that, for Naïve Bayes, RPV-LazyR obtained the smallest (best) Avg. Rank among all six methods. It also obtained the highest GM value in 17 out of the 33 datasets. The Holm post-hoc test indicated that RPV-LazyR is significantly better than No FS, All-Pos, All-Neg, RPV-IG and RPV-R. For 1-NN, the baseline All-Pos (which selects all positive features) achieved the highest GM in 10.5 (one tied win) datasets, but its Avg. Rank (2.8) was very close to the Avg. Rank of both RPV-R and RPV-LazyR (2.9), with no significant difference among them. Moreover, the Holm post-hoc test indicated that RPV-LazyR is significantly better than No FS and All-Neg.

Table 4: Comparing the proposed RPV against state-of-the-art feature selection methods in terms of AUCPR – in % values.

		Naïve Bayes						1-NN					
	Datasets	CFS	ReliefF	SHSEL	HIP	MR	RPV	CFS	ReliefF	SHSEL	HIP	MR	RPV
C.elegans	BP	47.4	47.4	56.5	**58.4**	55.6	56.4	47.2	46.2	52.5	**55.6**	52.6	50.4
	CC	48.6	50.7	49.6	**57.1**	54.5	54.3	50.8	50.3	48.4	50.6	52.8	**54.6**
	MF	41.8	40.6	45.6	50.6	50.0	**51.7**	40.2	42.1	45.5	49.2	47.3	**50.1**
	BP.CC	42.0	49.3	**60.0**	59.8	56.7	56.3	49.3	48.4	52.9	52.3	**54.2**	50.2
	BP.MF	44.9	48.1	**57.2**	57.0	53.9	55.4	48.1	46.2	48.6	**52.7**	49.0	50.1
	CC.MF	49.3	47.3	51.1	**54.2**	52.0	53.3	44.3	48.3	49.2	51.1	52.6	**54.6**
	BP.CC.MF	44.3	43.4	57.8	**58.1**	55.6	56.6	41.1	46.9	52.4	**52.9**	51.1	48.6
D.melanogaster	BP	82.8	81.5	84.1	**87.6**	82.0	82.5	**83.2**	82.8	82.2	78.3	79.3	79.3
	CC	82.3	82.1	88.3	88.6	89.6	**90.0**	81.3	80.0	**87.6**	83.7	82.4	83.4
	MF	76.9	76.8	79.3	**82.9**	80.4	81.1	76.2	78.4	81.5	80.0	80.6	**82.4**
	BP.CC	83.4	80.3	83.9	**88.5**	86.2	83.4	**83.1**	79.2	81.4	78.5	81.7	80.8
	BP.MF	81.1	80.5	83.1	**84.5**	84.2	83.3	**86.1**	77.9	76.4	75.8	77.3	80.4
	CC.MF	82.8	76.9	88.3	88.5	88.4	**89.7**	84.5	82.6	**85.5**	85.2	83.8	84.7
	BP.CC.MF	81.7	81.4	84.8	**87.2**	85.1	84.8	**87.7**	79.8	79.8	77.9	79.8	81.3
M.musculus	BP	72.6	70.7	85.7	**86.0**	85.3	85.2	75.3	72.7	**79.7**	73.7	75.6	77.6
	CC	71.5	70.4	82.8	80.0	**85.0**	84.4	68.7	72.9	**78.9**	70.0	78.3	76.6
	MF	74.3	70.2	**85.9**	85.5	82.7	85.6	80.0	69.0	**82.9**	81.2	82.8	79.1
	BP.CC	70.0	73.3	87.2	86.0	87.5	**88.6**	70.6	71.5	74.2	71.7	**76.5**	75.6
	BP.MF	73.4	72.4	**87.9**	87.2	86.5	86.8	74.2	72.8	78.0	75.9	76.6	**78.2**
	CC.MF	70.5	71.8	84.8	84.9	83.5	**86.2**	73.7	72.8	**84.4**	76.5	74.0	79.3
	BP.CC.MF	72.8	71.4	**88.3**	88.0	88.1	87.0	72.7	74.3	77.1	75.0	75.8	**78.2**
S.cerevisae	BP	28.1	25.4	41.1	46.3	41.2	**46.4**	27.9	30.6	**41.9**	31.8	28.0	36.4
	CC	13.3	13.3	26.3	30.4	29.9	**35.3**	11.2	14.9	30.2	31.7	28.3	**37.6**
	MF	19.4	19.4	24.0	**27.0**	25.8	25.2	15.7	19.4	23.6	**36.8**	36.0	35.8
	BP.CC	27.6	22.9	43.0	**48.5**	41.6	46.5	34.9	26.6	**35.3**	28.6	32.9	34.5
	BP.MF	19.6	19.9	40.5	46.0	42.8	**46.1**	32.9	26.6	**41.0**	27.2	35.1	40.5
	CC.MF	18.4	18.4	28.2	31.9	31.1	**32.7**	17.1	21.8	30.6	32.3	34.1	**36.6**
	BP.CC.MF	20.9	20.3	41.7	46.0	42.0	**46.2**	20.3	18.2	32.9	24.8	26.9	**34.3**
General	Tweets T	74.9	78.1	73.7	77.2	82.1	**83.3**	74.9	72.1	75.4	77.7	79.4	**82.6**
	Tweets C	90.0	87.8	82.3	85.3	87.8	**98.5**	87.8	86.4	84.2	94.1	87.8	**97.5**
	NY Daily	59.2	59.0	60.8	64.2	64.5	**64.8**	59.0	57.3	55.9	59.4	59.4	**60.1**
	Std.upon	70.5	66.1	76.7	75.9	76.2	**78.7**	66.4	65.8	73.6	73.4	72.6	**74.4**
	Cities	57.1	56.9	62.8	73.2	69.8	**74.1**	57.1	58.4	62.6	64.8	64.5	**69.4**
	Avg. Rank	5.0	5.5	3.2	2.2	3.0	**2.1**	4.5	5.2	2.8	3.3	3.0	**2.1**
	#Win	0.0	0.0	5.0	12.0	1.0	**15.0**	4.0	0.0	9.0	4.0	2.0	**14.0**

Naïve Bayes: {RPV} ≻ {CFS, ReliefF and SHSEL}

1-NN: {RPV} ≻ {CFS, ReliefF and HIP}

Note that the full set of positive feature values has much higher predictive power than the full set of negative values and the full set of features, since All-Pos obtained much better AUCPR and GM Avg. Rank values than All-Neg and No FS. This suggests that positive feature values are more informative than negative feature values. Also, RPV-LazyR has the best average rank and the highest number of wins for Naïve Bayes using AUCPR and GM, and for 1-NN using AUCPR. So, selecting the most relevant features using the LazyR relevance measure increases the predictive power when compared with the baselines methods. Also, this result indicates the benefit of using our proposed LazyR measure, rather than the R or IG measures. Hence, the RPV method using LazyR was chosen to be used in the next experiment due to its highest predictive accuracy.

Table 5: Comparing the proposed RPV against state-of-the-art feature selection methods in terms of GM – in % values.

		Naïve Bayes						1-NN					
	Datasets	CFS	ReliefF	SHSEL	HIP	MR	RPV	CFS	ReliefF	SHSEL	HIP	MR	RPV
C.elegans	BP	61.6	54.4	61.5	61.4	63.4	**65.7**	59.3	59.1	50.8	52.8	57.9	**60.1**
	CC	63.2	65.5	62.6	**68.6**	65.3	63.6	63.1	**63.6**	62.3	60.2	63.4	63.0
	MF	53.2	38.8	48.4	50.9	54.1	**56.7**	45.9	45.9	30.5	**51.3**	48.3	44.4
	BP.CC	63.9	55.2	62.4	64.1	63.9	**67.7**	55.8	61.2	56.5	54.7	60.1	**61.5**
	BP.MF	63.9	55.3	63.3	63.2	65.0	**65.5**	53.6	57.5	52.3	52.2	58.7	**62.8**
	CC.MF	61.5	61.6	57.9	61.4	63.1	**66.8**	**58.8**	57.4	56.4	54.8	57.1	57.8
	BP.CC.MF	62.0	51.0	63.3	63.0	65.2	**66.5**	57.4	56.8	57.6	55.3	59.6	**61.0**
D.melanogaster	BP	60.6	**69.3**	58.4	66.1	57.8	55.5	**66.2**	62.5	49.7	53.6	60.9	59.6
	CC	64.8	69.8	61.6	68.4	61.4	**74.4**	70.9	62.4	**75.0**	68.1	69.0	68.9
	MF	51.9	54.7	53.3	57.1	51.6	**67.2**	54.3	57.8	48.5	48.4	46.8	**70.9**
	BP.CC	49.5	65.2	55.2	**71.8**	50.0	64.5	**73.6**	62.1	62.4	49.0	58.1	64.1
	BP.MF	61.4	66.9	54.8	**68.1**	56.5	60.2	**67.4**	54.6	53.6	48.4	57.3	63.2
	CC.MF	59.4	53.8	65.1	65.7	65.8	**70.8**	65.6	64.8	**69.3**	60.0	61.6	66.6
	BP.CC.MF	60.7	69.2	57.7	**75.1**	56.7	66.5	**69.3**	63.3	58.2	50.8	54.5	63.1
M.musculus	BP	54.4	44.1	**68.8**	67.3	59.1	67.9	47.5	47.1	**60.3**	44.2	56.0	59.5
	CC	50.4	44.8	60.6	58.3	61.9	**69.4**	45.2	42.7	48.1	45.2	**56.7**	54.6
	MF	62.7	47.4	62.8	65.8	61.1	**67.9**	47.6	38.8	65.4	53.0	65.9	**66.1**
	BP.CC	55.0	43.6	67.7	69.1	67.4	**69.2**	45.6	50.0	57.4	52.7	**57.5**	53.3
	BP.MF	63.6	50.0	66.6	68.4	54.4	**69.9**	51.8	44.7	60.4	45.7	58.2	**63.0**
	CC.MF	54.0	46.3	62.2	**68.1**	62.7	67.3	44.0	50.4	63.9	51.7	53.9	**64.0**
	BP.CC.MF	63.1	47.1	65.5	70.5	60.6	**71.2**	49.6	53.8	61.8	52.4	56.2	**62.9**
S.cerevisae	BP	64.8	50.2	52.1	68.8	**69.1**	61.6	50.9	50.2	51.4	44.2	38.4	**57.2**
	CC	46.2	0.0	33.1	47.8	42.2	**59.9**	0.0	15.6	33.3	36.8	**38.9**	38.8
	MF	26.1	26.3	26.3	42.8	31.9	**54.6**	26.3	26.3	21.1	**38.5**	36.2	34.0
	BP.CC	65.2	47.7	59.0	**67.4**	62.5	58.1	52.3	**55.7**	45.9	41.7	43.5	49.5
	BP.MF	61.8	43.6	49.8	**68.3**	58.1	63.1	38.4	43.6	49.6	41.4	44.5	**65.8**
	CC.MF	36.7	26.2	46.4	57.4	46.4	**66.2**	26.2	26.2	21.0	41.0	38.5	**46.5**
	BP.CC.MF	61.4	47.6	57.1	**66.4**	62.4	58.4	40.6	47.3	46.4	35.4	43.8	**62.0**
General	Tweets T	70.6	70.8	63.1	73.0	71.8	**73.6**	74.8	68.8	64.2	72.5	72.2	**75.1**
	Tweets C	87.6	89.4	89.7	72.6	77.8	**94.8**	91.9	88.5	0.0	84.2	84.1	**95.0**
	NY Daily	48.7	43.8	39.1	52.4	49.8	**56.6**	49.0	42.9	41.8	52.1	52.5	**52.6**
	Std.upon	67.1	67.3	68.2	66.4	66.4	**70.7**	**71.9**	66.7	68.8	70.5	71.3	71.1
	Cities	59.5	55.0	66.0	**86.1**	73.9	71.8	56.8	56.3	57.5	61.1	58.2	**61.4**
	Avg. Rank	4.1	4.7	4.2	2.5	3.6	**1.9**	3.4	4.1	3.8	4.5	3.2	**2.0**
	#Win	0.0	1.0	1.0	9.0	1.0	**21.0**	6.0	2.0	3.0	2.0	3.0	**17.0**

Naïve Bayes: $\{\text{RPV}\} \succ \{\text{CFS, ReliefF, SHSEL and MR}\}$

1-NN: $\{\text{RPV}\} \succ \{\text{CFS, ReliefF, SHSEL, HIP and MR.}\}$

5.3.2 Comparison against state-of-the-art feature selection methods

In this experiment, the RPV method is evaluated against two non-hierarchical feature selection methods: CFS (using WEKA's default parameters, with best first search) and ReliefF (with a threshold set to 0.01); and three recent hierarchical feature selection methods: SHSEL (using Information Gain, with a threshold set to 0.99), HIP and MR, which were reviewed in Section 3. The MR method could also employ the LazyR instead of the original R measure. However, previous experiments (not reported here) have shown that the R measure is the best relevance measure to MR. So, we use the original MR with R in the following experiments.

Table 4 shows that RPV achieves both the best average rank and the highest number of wins for both NB and 1-NN, in terms of AUCPR. For NB, the Holm post-hoc test indicates that RPV is statistically better than CFS, ReliefF and SHSEL. There was no statistical difference between RPV and HIP, nor between RPV and MR, however RPV was the winner in 15 datasets, while HIP was the winner in 12 datasets and MR in just one dataset. For 1-NN, the Holm test indicates that RPV is significantly better than CFS, ReliefF and HIP.

Table 5 shows that RPV achieves both the best average rank and the highest number of wins for both Naïve Bayes and 1-NN, in terms of GM. For NB, the Holm post-hoc test indicates that RPV is statistically superior to two hierarchical methods, SHSEL and MR, as well as superior to the non-hierarchical CFS and ReliefF. There was no significant difference between RPV and HIP methods, but RPV was the winner (highest GM) in 21 datasets, whilst HIP was the winner in 9 datasets. For 1-NN, the post-hoc test indicates that RPV is statistically superior to all other methods in terms of GM.

Note that there is a large difference of performance between RPV and HIP. RPV achieves a higher number of wins and a better average rank than HIP in all experiments reported in Tables 4 and 5.

Summarizing, using both AUCPR and GM measures, and both Naïve Bayes and 1-NN classifiers, RPV achieved better average rank and higher number of wins than the traditional feature selection methods CFS and ReliefF, and also than the hierarchical methods SHSEL, HIP and MR. In all comparisons RPV is statistically superior to CFS and ReliefF. When compared to SHSEL, HIP and MR, in some cases, the differences were not statistically significant, however, in all these cases, RPV clearly outperformed the hierarchical methods in terms of average rank and number of wins.

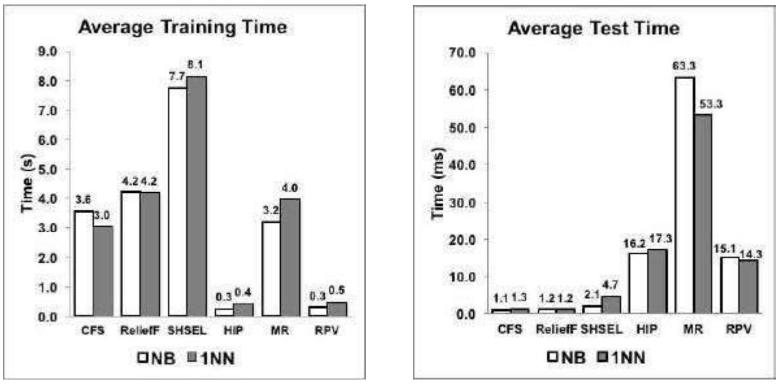
5.3.3 Running time performance

Figure 2 shows the results of the experiments which measure the runtime of the methods on a computer with 4 GB of RAM and an Intel Core i5 1.6GHz CPU. Subfigure 2(a) shows the average training time, i.e., the time required for selecting features and building the Naïve Bayes model for eager methods; and for lazy methods, the time required for pre-computing the probabilities used by Naïve Bayes and computing for each feature in the dataset: the ancestors and descendants (for HIP and RPV), the paths from a feature to root and leafs (for MR) and the relevance value (for MR and RPV). Subfigure 2(b) shows the average testing time, i.e., the time required for classifying one instance with Naïve Bayes or 1-NN for the eager methods, and the time required for feature selection and classifying one instance for the lazy methods. The runtimes reported in Figure 2 were averaged over the 33 datasets.

CFS, ReliefF and SHSEL are eager methods and find a unique subset of features during the training step which will be used to classify all test instances, while the lazy methods HIP, MR and RPV find a subset of features for each test instance in the classification step. Hence, as can be observed in Figure 2, CFS, ReliefF and SHSEL have the worst training times and the best test times. On the other hand, HIP, MR and RPV have the lowest training times and the highest test times. Note that our proposed RPV method is, in general, the fastest among the lazy methods.

5.3.4 An analysis of the worst-case time complexity of RPV

The worst-case time complexity of RPV can be calculated as follows. First of all, note that, in the worst case scenario, the feature DAG has N nodes (features) and



(a) Average training time in seconds (s).

(b) Average test time in milliseconds (ms)

Fig. 2: Average training and test times for CFS, ReliefF, SHSEL, HIP, MR and RPV methods.

$\frac{N(N-1)}{2}$ edges. I.e., each feature is linked to all features but its own descendants. Hence, line 2 of Algorithm 1 has worst-case time complexity $O(N)$, but this line is executed N times since it is within a *for* loop, which takes $O(N^2)$. Line 3 of Algorithm 1 requires the value of the LazyR (Lazy Relevance) measure for each feature, which is precomputed before Algorithm 1 is run, with a time complexity of $O(N.M)$, where N is the number of features and M is the number of training instances. The time taken by lines 3 and 4 are constant (simple assignment), and therefore they can be ignored in the analysis, since their time complexity is dominated by the one of line 2. During RPV's feature elimination procedure, performed by the nested loop starting in line 6, in the worst case, in line 9 the relevance value of the most specific feature is compared to the relevance of $N-1$ ancestors, the second most specific feature is compared to $N-2$ ancestors and so on, until all features have been evaluated. The other lines in the nested loop, lines 10 and 14, do not change the time complexity associated with this loop. In total, in the worst case, the nested loop starting at line 6 of Algorithm 1 performs $\frac{N(N-1)}{2}$ comparisons, i.e., $O(N^2)$.

Hence, in addition to the time $O(N.M)$ to pre-compute all LazyR values, Algorithm 1 takes $O(N^2)+O(N^2)$, which in total is $O(NM+N^2)$. Note that Algorithm 1 is executed once for each test instance to be classified. Hence, the total worst-case time complexity of the RPV feature selection method is $O(N.M+t.N^2)$, where N is the number of features, M is the number of training instances, and t is the number of test instances to be classified.

Note, however, that in practice the time taken by RPV tends to be much smaller than suggested by this worst-case analysis, because in practice the number of ancestors of each feature is usually much smaller than the theoretical maximum of $N-1$ (a key assumption in the above analysis). For instance, in dataset DM-BP.CC.MF, which has 1714 features, the feature with the largest number of ancestors has only 6 ancestors.

Table 6: Average percentage of features selected by CFS, ReliefF, SHSEL, HIP, MR, All-Neg, All-Pos, RPV-R, RPV-IG and RPV.

	Dataset	CFS	ReliefF	SHSEL	HIP	MR	All-Neg	All-Pos	RPV-R	RPV-IG	RPV
C. elegans	BP	4.64	12.30	**1.64**	7.02	17.78	95.58	4.42	1.69	1.91	1.68
	CC	8.88	9.95	4.61	16.29	33.56	93.91	6.09	2.80	2.98	**2.75**
	MF	10.23	8.33	2.97	11.27	23.68	95.31	4.69	2.55	2.86	**2.48**
	BP.CC	4.00	12.22	1.91	8.00	20.20	95.61	4.39	2.01	1.92	**1.71**
	BP.MF	4.67	11.00	1.88	7.50	18.92	98.44	4.21	**1.56**	1.89	1.69
	CC.MF	7.77	9.95	3.34	12.28	28.03	95.49	4.60	2.66	2.30	**2.25**
	BP.CC.MF	3.87	11.24	2.03	8.25	20.91	95.78	4.22	**1.56**	1.93	1.75
D. melanogaster	BP	**3.71**	18.98	12.46	11.88	22.68	91.80	8.20	5.20	4.77	4.23
	CC	13.71	38.89	15.28	21.67	35.08	88.09	11.91	6.50	6.29	**5.78**
	MF	8.84	38.36	14.11	15.42	22.58	93.01	6.99	4.50	**4.44**	4.75
	BP.CC	3.54	18.54	12.84	12.73	23.78	91.57	8.43	**3.53**	3.90	4.23
	BP.MF	9.28	20.82	12.60	12.33	22.52	92.07	7.93	**3.47**	3.77	4.17
	CC.MF	7.77	42.13	14.68	17.52	27.71	91.49	8.51	**4.36**	4.90	4.92
	BP.CC.MF	6.14	22.71	12.89	13.07	23.59	91.81	8.19	**3.60**	3.93	4.27
M. musculus	BP	12.75	25.81	15.18	11.67	21.47	89.44	10.56	**4.00**	4.70	4.51
	CC	10.91	36.11	18.11	28.36	32.65	83.98	16.02	8.99	9.45	**8.43**
	MF	7.50	28.22	16.00	20.10	25.18	90.65	9.35	6.55	**5.71**	6.15
	BP.CC	12.65	27.32	15.52	13.27	22.53	88.93	11.07	4.94	4.95	**4.86**
	BP.MF	12.53	25.87	15.54	12.90	22.09	89.66	10.34	4.88	**4.63**	4.72
	CC.MF	16.25	33.42	17.90	22.88	27.84	88.41	11.59	6.99	6.78	**6.76**
	BP.CC.MF	12.52	26.76	15.80	14.18	22.91	89.20	10.80	5.55	5.05	**5.01**
S. cerevisae	BP	3.58	35.86	2.91	6.98	17.85	94.77	5.23	**1.27**	1.93	2.26
	CC	15.17	36.30	**4.90**	19.70	33.31	90.64	9.36	6.05	5.17	5.92
	MF	12.94	43.24	3.12	10.84	25.98	94.73	5.27	3.80	**2.95**	3.37
	BP.CC	3.44	35.56	3.21	8.85	20.14	94.15	5.85	2.87	**2.36**	2.80
	BP.MF	3.37	37.15	2.96	7.78	19.56	94.75	5.25	2.56	**2.14**	2.49
	CC.MF	8.60	38.96	**3.84**	14.39	28.96	93.09	6.91	4.60	4.72	4.39
	BP.CC.MF	3.13	37.41	3.20	9.22	21.22	94.25	5.75	2.98	3.05	**2.91**
General	Tweets T	4.26	8.00	4.05	10.05	36.83	98.19	1.81	0.90	1.07	**0.89**
	Tweets C	2.06	13.27	48.56	39.01	43.11	98.98	1.02	0.83	0.85	**0.82**
	NY Daily	8.80	1.92	5.31	14.68	20.31	97.79	2.21	1.24	1.15	**1.04**
	Stb.upon	2.36	5.07	26.16	30.39	63.80	98.83	1.17	0.92	0.92	**0.79**
	Cities	17.96	24.27	29.03	45.91	59.68	76.26	23.84	10.65	10.62	**10.57**
	#Win	1.0	0.0	3.0	0.0	0.0	0.0	0.0	8.0	6.0	**15.0**
	Avg.	8.12	23.72	11.05	15.65	27.46	92.62	7.46	3.83	3.81	**3.80**

5.3.5 Evaluating the feature space compression

The selection of a small subset of relevant features for each instance may improve the interpretability of the model's predictions, since only relevant features are used to justify each prediction. So, we report the percentage of features selected by each method in Table 6. Again, the table's first two columns show the feature hierarchies (GO term types) used to build the datasets and the name of the general domain datasets. The following columns show the percentage of features selected by each method. The last two rows show the number of wins (#Win) and the average percentage of features selected (Avg.) across all 33 datasets.

The results show that, in general, the three RPV versions (with three distinct relevance measures) select a feature subset smaller than the one selected by all other seven methods. RPV (with the LazyR measure) selects, on average, only 3.80% of all features, whilst achieving the highest predictive accuracy in general. Note that the percentage of features selected by RPV is about a half of those selected by All-Pos. It means that not all positive feature values are relevant and that the correct identification of relevant positive feature values increases the predictive accuracy of Naïve Bayes and 1-NN. In this work, we introduced the LazyR measure to identify such features. Although the HIP method (which selects the set of the most specific positive-valued features and the most general negative-valued features) achieved a good predictive performance, the RPV method obtained both a higher predictive performance and a much smaller selected feature subset.

6 Conclusion and Future Work

This paper presented a novel lazy method for hierarchical and sparse feature selection based on the hypothesis that positive feature values provide more meaningful and accurate information, even though they are present to a small extent in each instance. Our method, named Select Relevant Positive Feature Values (RPV), has some interesting properties: (i) it selects rare but informative and relevant positive features; (ii) it selects smaller feature subsets; and (iii) it is based on a new lazy feature relevance measure (LazyR) which assesses the predictive power of a feature value specifically in the current test instance being classified.

The computational experiments involved 33 real-world datasets and four different classification scenarios, namely four combinations of two different classification algorithms (Naïve Bayes and 1-NN) times two different predictive accuracy measures (AUCPR and the Geometric Mean of Sensitivity and Specificity). The results of these experiments have shown that the proposed RPV method obtained in general the best predictive accuracy across those four classification scenarios. More precisely, a statistical significance test was used to compare RPV against each of 10 other feature selection approaches, in each of the above four classification scenarios; and the results of that test have shown that RPV obtained predictive accuracies statistically significant better than another feature selection approach in 28 out of the 40 cases. In addition, in none of those cases RPV's predictive accuracy was significantly worse than the accuracy of any other feature selection approach. Furthermore, RPV selected in general the smallest subset of features, among all evaluated feature selection methods. This is also desirable, since each instance is classified using its own small set of relevant features; hence each instance's classification is justified by a more specific feature subset, improving prediction interpretability.

In addition, the hypothesis that selecting positive feature values might increase the predictive accuracy of the classifier (mentioned earlier) is supported by two types of results. First, the fact that RPV, which selects only a subset of positive feature values (i.e., it never selects negative feature values) obtained by far the best predictive accuracy results. Second, the fact that the results of the All-Pos baseline method, which selects all positive feature values (and no negative values) were clearly better than the results of the All-Neg method, which selects all negative feature values (and no positive values), as discussed in Section 5.3.1.

In future work, we plan to evaluate the behaviour of the proposed RPV method in each application domain analysing the usefulness of the selected features with the assistance of specialists from those domains.

7 Acknowledgements

We thank Dr. Cen Wan for kindly providing us with the program code of the HIP and MR methods. This work was supported by the Brazilian government's agencies CAPES and CNPq.

References

1. D. Aha. *Lazy Learning.* Kluwer Academic Publishers, 1997.
2. T. Cover and J. Thomas. *Elements of Information Theory.* Wiley-Interscience, second edition, 2006.
3. P. da Silva, A. Plastino, and A. Freitas. A novel genetic algorithm for feature selection in hierarchical feature spaces. In *Proc. of the 2018 SIAM International Conference on Data Mining*, pages 738–746. SIAM, 2018.
4. J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
5. J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan. Feature selection based on structured sparsity: a comprehensive study. *IEEE Transactions on Neural Networks and Learning Systems*, 28(7):1490 – 1507, 2016.
6. M. Hall. *Correlationbased Feature Selection for Machine Learning.* Phd thesis, University of Waikato, New Zealand, 1999.
7. M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th International Conference on Machine Learning (ICML)*, pages 359–366, 2000.
8. M. Hall, E. Frank, G. Holmes, B. Pfahringer, and P. Reutemann. The weka data mining software: an update. *ACM SIGKDD Exploration Newsletter*, 11(1):10–18, 2009.
9. J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2011.
10. S. Holm. A simple sequential rejective method procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.
11. N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Perspective.* Cambridge University Press, 2011.
12. Y. Jeong and S.-H. Myaeng. Feature selection using a semantic hierarchy for event recognition and type classification. In *Proc. 6th Intl. Joint Conf. on NLP (IJCNLP)*, pages 136–144, 2013.
13. I. Kononenko. Estimating attributes: Analysis and extensions of relief. In *Proc. 7th European Conference on Machine Learning*, 1994.
14. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.
15. J. Li, K. Cheng, S. Wang, F. Morstatter, R. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *arXiv preprint arXiv:1601.07996*, 2016.
16. H. Liu and H. Motoda. *Computational Methods of Feature Selection.* CRC Press, 2008.
17. H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining.* Springer, 2012.

18. H. Liu and R. Setiono. A probabilistic approach to feature selection: A filter solution. In *Proc. 13th International Conference on Machine Learning (ICML)*, pages 319–327, 1996.
19. J. Liu and J. He. Moreau-yosida regularization for grouped tree structure learning. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1459–1467. Curran Associates, Inc., 2010.
20. S. Lu, Y. Ye, R. Tsui, H. Su, R. Rexit, S. Wesaratchakit, X. Liu, and R. Hwa. Domain ontology-based feature reduction for high dimensional drug data and its application to 30-day heart failure readmission prediction. In *Proc. 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 478–484, 2013.
21. J. Magalhães, A. Bukovsky, G. Lehmann, J. Costa, Y. Li, V. Fraifeld, and G. Church. The human ageing genomic resources: Online databases and tools for biogerontologistis. *Ageing Cell*, 8(1):65–72, 2009.
22. R. Pereira, A. Plastino, B. Zadrozny, L. Merschmann, and A. Freitas. Lazy attribute selection: Choosing attributes at classification time. *Intelligent Data Analysis*, 15(5):715–732, 2011.
23. C. Qi, L. Yi, H. Su, and L. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 5099–5108. Curran Associates, Inc., 2017.
24. P. Ristoski and H. Paulheim. Feature selection in hierarchical feature spaces. In S. Dzeroski, P. Panov, D. Docev, and L. Todorovski, editors, *Proc. Discovery Science 2014*, volume 8777 of *LNCS*, pages 288–300. Springer, 2014.
25. C. Stencil and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.
26. The GO Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
27. J. Vergara and P. Estévez. A review of feature selection methods based on mutual information. *Neural Computing and Applications*, 24(1):175–186, 2014.
28. C. Wan and A. Freitas. Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegant genes based on bayesian classification methods. In *Proc. 2013 International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 373–380. IEEE Press, 2013.
29. C. Wan and A. Freitas. Two methods for constructing a gene ontology-based feature network for a bayesian network classifier and applications to datasets of ageing-related genes. In *Proc. 6th ACM Conf. on Bioinfo., Comp. Biology and Health Informatics (BCB)*, pages 27–36, 2015.
30. C. Wan and A. Freitas. An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features. *Artificial Intelligence Review*, 50(2):201–240, 2017.
31. C. Wan, A. Freitas, and J. 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, 2015.
32. L. Wang, Y. Wang, and Q. Chang. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111(1):21–31, 2016.

Table A.1: Results from the statistical analysis of the experiments presented in Section 5.3.1 and Section 5.3.2.

AUCPR (Results for Table 2)									
Naïve Bayes					1-NN				
Methods	Avg. Rank	p-value	adjusted α	Sig?	Methods	Avg. Rank	p-value	adjusted α	Sig?
RPV	2.5				RPV	2.4			
RPV-IG	2.8	5.15E-01	0.017	No	RPV-R	2.4	8.28E-01	0.017	No
RPV-R	3.3	1.65E-02	0.017	Yes	RPV-IG	2.8	7.70E-01	0.017	No
All-Pos	3.5	1.11E-03	0.017	Yes	All-Pos	3.2	2.47E-01	0.017	No
No FS	3.7	1.22E-02	0.013	Yes	No FS	5	5.00E-05	0.013	Yes
All-Neg	5.2	5.00E-05	0.010	Yes	All-Neg	5.2	5.00E-05	0.010	Yes
Friedman's $X_f^2 = 42.05$ (Yes)					Friedman's $X_f^2 = 76.75$ (Yes)				
GM (Results for Table 3)									
Naïve Bayes					1-NN				
Methods	Avg. Rank	p-value	adjusted α	Sig?	Methods	Avg. Rank	p-value	adjusted α	Sig?
RPV	1.9				RPV	2.9			
RPV-R	3	1.69E-02	0.050	Yes	All-Pos	2.8	1.00E+00	0.017	No
RPV-IG	3.1	9.18E-03	0.025	Yes	RPV-R	2.9	1.00E+00	0.017	No
All-Pos	3.4	1.13E-03	0.017	Yes	RPV-IG	3	1.00E+00	0.017	No
No FS	3.7	9.30E-05	0.013	Yes	No FS	3.7	1.27E-02	0.013	Yes
All-Neg	6	1.00E-05	0.010	Yes	All-Neg	5.8	5.00E-05	0.010	Yes
Friedman's $X_f^2 = 94.00$ (Yes)					Friedman's $X_f^2 = 70.62$ (Yes)				
AUCPR (Results for Table 4)									
Naïve Bayes					1-NN				
Methods	Avg. Rank	p-value	adjusted α	Sig?	Methods	Avg. Rank	p-value	adjusted α	Sig?
RPV	2.1				RPV	2.1			
HIP	2.2	8.34E-01	0.017	No	SHSEL	2.8	1.29E-01	0.025	No
MR	3	1.00E-01	0.017	No	MR	3	1.02E-01	0.025	No
SHSEL	3.2	1.62E-02	0.017	Yes	HIP	3.3	1.40E-02	0.017	Yes
CFS	5	5.00E-05	0.013	Yes	CFS	4.5	5.00E-05	0.013	Yes
ReliefF	5.5	5.00E-05	0.010	Yes	ReliefF	5.2	5.00E-05	0.010	Yes
Friedman's $X_f^2 = 96.55$ (Yes)					Friedman's $X_f^2 = 55.91$ (Yes)				
GM (Results for Table 5)									
Naïve Bayes					1-NN				
Methods	Avg. Rank	p-value	adjusted α	Sig?	Methods	Avg. Rank	p-value	adjusted α	Sig?
RPV	1.9				RPV	2			
HIP	2.5	1.94E-01	0.050	No	MR	3.2	9.32E-03	0.050	Yes
MR	3.6	4.48E-04	0.025	Yes	CFS	3.4	4.89E-03	0.025	Yes
CFS	4.1	5.00E-05	0.017	Yes	SHSEL	3.8	2.88E-04	0.017	Yes
SHSEL	4.2	5.00E-05	0.013	Yes	ReliefF	4.1	5.00E-05	0.013	Yes
ReliefF	4.6	5.00E-05	0.010	Yes	HIP	4.5	5.00E-05	0.010	Yes
Friedman's $X_f^2 = 46.48$ (Yes)					Friedman's $X_f^2 = 35.83$ (Yes)				

Appendix A - Statistical Analysis

This appendix shows more detailed results of the statistical analysis for the experiments reported in Sections 5.3.1 and 5.3.2. Table A.1 shows the detailed results of the Friedman test and the Holm post-hoc test. This table is organized as follows. Each one of the four parts of the table shows the results of the statistical test from a different table in the results Section (5.3.1 or 5.3.2). The left-handed side of each part shows the statistical results for the Naïve Bayes classifier, while the right-handed side shows the results for the 1-NN classifier. The five columns in the left and right parts of the Table represent, respectively, the method's name, its average rank, the p-value obtained by the Holm test, the adjusted α and whether or not the comparison between RPV and the given method is statistically significant (Sig?) according to the Holm post-hoc test. The last row in each of the four parts of the table shows the value of the computed Friedman's statistic (X_f^2) and whether or not the test's result is statistically significant.

For the Friedman test, a significant difference is found when the value of X_f^2 is greater than the critical value of 12.83 (this number is defined for the comparison with $k = 6$ methods,

$n = 33$ datasets and 5% of significance level). We first run the Friedman test to verify if there is a significant difference among those k methods' results. After that, if the result of the Friedman test is statistically significant, we execute the Holm test to identify the pair of methods with statistically different results. Since the value of each one of the eight comparisons shown is greater than the critical value, we can say that all experiments have detected significant differences among at least one pair of methods. So, we applied the Holm test to all scenarios. Likewise, the results presented for the Holm test are statistically significant when the p-value is lower than the adjusted α. Both the p-value and the adjusted α were internally calculated by the Holm test.

APPENDIX B – da Silva, P.N., Plastino, A., Freitas, A.A. “A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces”. In Proc. of the 2018 SIAM International Conference on Data Mining (SDM) (2018), pages 738-746. SIAM

A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces

Pablo Nascimento da Silva[*] Alexandre Plastino[*] Alex A. Freitas[†]

Abstract

Feature selection methods have been widely adopted to prepare high-dimensional feature spaces for the classification task of data mining. However, in many real-world datasets, the feature space is formed by binary features related via generalization-specialization relationships, also known as hierarchical feature spaces. Although there are many methods for the traditional feature selection problem, methods which properly consider hierarchical features are still very underexplored. In this work, we propose a novel genetic algorithm (GA) for hierarchical feature selection. The proposed GA has two novel hierarchical mutation operators tailored to deal with redundant features in hierarchical feature spaces. The computational experiments show that our proposed approach exhibited better predictive performance than two state-of-the-art hierarchical feature selection methods (SHSEL and HIP) and also than two traditional feature selection methods (ReliefF and CFS).

Keywords: Hierarchical Feature Spaces, Feature Selection, Genetic Algorithms, Classification, Bioinformatics.

1 Introduction

Classification is one of the most important tasks in the data mining field [23]. In this task a previously trained classification model automatically assigns a class label to a new instance, based on the values of its features. In many interesting real-world problems, each instance is formed by a set of hierarchically organized binary features. In other words, in each instance, a feature value is deemed positive (negative) when the property associated with the feature has been (has not been) observed for that instance. Besides, such features are related via generalization-specialization relationships, characterizing hierarchical feature spaces [7, 12, 16, 18, 19, 20, 21]. In a generalization-specialization hierarchy, for any given instance t, if a feature x has positive value in t, denoted ($x = 1$), then all ancestors of x in the feature hierarchy also have positive value in t. In contrast, if a feature x has negative value in t, denoted ($x = 0$), then all descendants of x in the feature hierarchy also have negative value in t.

One example of data often characterized by hierarchical feature spaces is biological data [18, 19, 20]. For instance, in this work we address the problem of hierarchical feature selection on datasets of ageing-related genes [2]. Ageing is a complex biological process that affects nearly all animal species, even though it is still poorly understood [14]. However, the increasing amount of available ageing-related data allows the use of data mining methods to discover novel patterns that could improve the understanding of the biological ageing process. In the datasets explored in this work, each instance represents a gene, and each gene is associated with terms derived from the Gene Ontology [17], as described later. In these datasets, a general feature (e.g., reproduction) would be the ancestor of more specific features (e.g., asexual reproduction).

Real-world datasets often have a large number of features, many of which can be redundant (highly correlated with other features) or irrelevant for classification (having no significant correlation with the class variable). The problem of redundant features, in particular, is a recurrent issue in hierarchical feature spaces, since hierarchically related features (i.e., features on the same path within the hierarchy) tend to be highly correlated (redundant) with each other, as will be seen later. Hence, by removing hierarchically redundant features in a data preprocessing phase one can improve the classifier's predictive accuracy, speed up the learning process and improve the interpretability of the classifier.

Existing hierarchical feature selection methods [7, 12, 16, 18, 21] usually find a suitable feature subset by keeping features with good relevance values and removing redundancy among hierarchically related features. However, none of the existing methods employs an effective search method; they just use a simple criterion for removing hierarchically redundant features.

In this work, we introduce a new genetic algorithm (GA) with two mutation operators specifically designed to cope with hierarchically redundant features, in order to increase predictive accuracy. These mutation operators are based on two principles: (i) exploiting generalization-specialization relationships among

[*] Universidade Federal Fluminense, {psilva, plastino}@ic.uff.br

[†] University of Kent, a.a.freitas@kent.ac.uk

the features; and (ii) removing hierarchical redundancy among the features. In essence, the proposed mutation operators attempt to reduce the number of hierarchically redundant features by assigning to each feature in a candidate feature subset a different probability of mutation. This probability is determined by the degree of correlation among hierarchically redundant features.

Experiments showed that the proposed GA achieved better predictive accuracy than two traditional and two hierarchical feature selection methods.

The remainder of this work is organized as follows. Section 2 reviews essential concepts of feature selection for classification, hierarchical feature spaces and genetic algorithms. Section 3 describes the related work. Our novel genetic algorithm is introduced in Section 4. Section 5 introduces the two novel mutation operators based on the hierarchy of features. In Section 6, we report the computational results. Finally, Section 7 presents the conclusions.

2 Background

2.1 Feature Selection for Classification The predictive accuracy of classifiers is significantly influenced by the quality of the input features [10, 11]. The main goal of feature selection methods is to increase predictive accuracy by selecting a subset of features that are relevant for classification and non-redundant (with little or no correlation among the features).

Feature selection methods can be divided into embedded, filter and wrapper methods [10, 11]. Embedded methods select features during the training of the classifier; whilst wrapper and filter methods are used in a data preprocessing phase. Filter methods evaluate the quality of a feature subset using specific measures, without using the target classification algorithm. By contrast, wrapper methods evaluate the quality of a feature subset by measuring the predictive accuracy of a classifier built using that subset. Hence, wrapper methods select a feature subset tailored specifically for the target classification algorithm, which increases the chances of maximizing predictive accuracy for that algorithm. We follow the wrapper approach in this work.

2.2 Feature Selection and Hierarchical Spaces We use the following notation. The i-th instance of a dataset D consists of a d-dimensional vector of binary features $(x_{i1}, x_{i2}, \dots x_{id})$, $x_{ij} \in \{0, 1\}$ for all $1 \leq j \leq d$. In this work the feature set X of D is a hierarchical feature space, more precisely a Direct Acyclic Graph (DAG), where each vertex (node) represents a feature and each edge represents a generalization-specialization relationship between two features. An edge $(X_a \rightarrow X_b)$ shows that feature X_a is a parent of feature X_b, and conversely X_b is a child of X_a. More generally, a feature X_a is an ancestor (descendant) of a different feature X_b if and only if there is a sequence of edges leading from X_a to X_b (from X_b to X_a) in the feature DAG. In generalization-specialization hierarchies ("IS-A hierarchies"), for each instance t, if a feature x has positive value in t $(x = 1)$, then all ancestors of x in the hierarchy also have positive values in t. In contrast, if a feature x has negative value in t $(x = 0)$, then all descendants of x in the hierarchy also have negative values in t. Note that IS-A hierarchies lead to hierarchical redundancy among features, since a specific feature value logically implies the values of all its ancestors or descendants, as explained above.

Hierarchical feature selection methods are a special case of feature selection methods that exploit characteristics of the feature DAG to improve the predictive accuracy of classifiers. This is typically done by removing hierarchically redundant features [16, 19].

2.3 Genetic Algorithms (GAs) A GA is a stochastic search method inspired by Charles Darwin's natural evolution theory [15]. A GA works with a population of individuals (candidate feature subsets in this work) that iteratively undergo selection and modification, evolving towards a good solution for a given problem. In essence, a GA works as follows. First, an initial population of individuals is randomly created. Then, the quality of each individual is evaluated by a fitness function. At each generation (iteration), the best individuals (those with the highest fitness values) are selected more often for reproduction. The selected individuals undergo genetic operations, like crossover (which combines parts of two individuals to create a new individual) and mutation (where a small part of an individual is replaced according to a randomly generated value). The reproduction process produces offspring which will replace the parents, creating a new generation of individuals which are expected to be better than the previous generation's individuals. This process is repeated until a stopping criterion (e.g., a fixed number of generations) is satisfied.

3 Related Work

Traditional (non-hierarchical) feature selection methods – e.g., Correlation-based Feature Selection (CFS) [4] and ReliefF [8] – can be employed in hierarchical feature spaces by completely ignoring the structure of the feature hierarchy, i.e., treating the features as a flat set of features. However, this is a naive approach to cope with hierarchically redundant features. When the feature space is hierarchical, intuitively the use of hierarchical feature selection methods is more likely to effectively cope with hierarchically redundant features, by exploiting the generalization-specialization relationships in the feature hierarchy. Next, we briefly review existing hierarchical feature selection methods.

SHSEL [16] is based on the principle that, if there is an edge between two features in the hierarchy (i.e., one is a parent of the other), in general they are highly correlated and tend to be redundant for classification. Hence, for each pair of features connected by an edge in the hierarchy, SHSEL removes the most specific (child) feature if the correlation between those two features is above a user-defined threshold. Next, using only the remaining features, for each path in the feature hierarchy, SHSEL keeps the features with relevance higher than the average relevance of features in that path. A related method, Greedy Top-Down search strategy (GTD) [12], selects the features with the highest relevance value in each path from each leaf to the root node in the hierarchy. Moreover, Tree-Based Feature Selection (TSEL) [7] has been used in the special case of tree-structured (rather than DAG-structured) features. A recent work showed that SHSEL achieved better results than TSEL and GTD [16]. Thus, TSEL and GTD are no longer considered in this work.

Some hierarchical feature selection methods follow the lazy learning paradigm, selecting a different feature subset for each new test instance to be classified. Such lazy methods are the Select Hierarchical Information-Preserving Features (HIP) method [19], the Select Most Relevant Features (MR) method [19], and the hybrid HIP-MR method [18, 19]. Since the HIP method obtained better results than MR and HIP-MR in [21], hereafter we only consider the HIP method.

For each new instance to be classified, HIP selects the subset of the most specific positive-valued features (which imply their ancestors) and the most general negative-valued features (which imply their descendants). As a result, the values of the features selected by HIP for an instance imply the values of all other features for that instance, so that this method completely removes the hierarchical redundancy in the original feature set. Actually, HIP selects only features whose values are non-hierarchically redundant, i.e., features whose values cannot be inferred from the values of other features. However, HIP has the limitation of not explicitly taking into account the relevance (the degree of correlation with the class variable) of the features.

Note that all above methods follow the filter approach. Our proposed GA seems to be the first hierarchical feature selection method based on the wrapper approach.

In this work, we compare our proposed methods against the state-of-the-art hierarchical feature selection methods HIP and SHSEL, as well as against the traditional feature selection methods CFS and ReliefF.

4 A Novel Genetic Algorithm for Feature Selection in Hierarchical Feature Spaces

The problem of redundant features is a recurrent issue in the classification task. In datasets with hierarchical features, the structure of the feature space can be used to mitigate this matter through the elimination of hierarchically connected features. In fact, this removal is performed by all hierarchical feature selection methods proposed so far. However, none of these methods employ an effective heuristic search to deal with this problem. So, in this work, we propose a new genetic algorithm (GA), named Genetic Algorithm for Hierarchical Feature Selection (GA-HFS). We focus on GAs because they perform a global search, less likely to get trapped in local optima than local search methods [3, 15]; and they have been successfully employed in non-hierarchical feature selection [3, 22].

GA-HFS uses new mutation operators to guide the search towards feature subsets with few hierarchically redundant features. These mutation operators use the hierarchical structure of the feature space to determine the value of a biased mutation probability for each feature – i.e., the probability of changing a feature's status from selected to non-selected.

4.1 Individual Representation Individuals are represented by d-dimensional binary vectors (using the value 1 for selected features and 0 for non-selected features), where d is the number of features in the dataset. Figure 1 shows an example of an individual representation. Each letter from A to N represents a feature in the hierarchy, and an edge represents a generalization-specialization relationship – e.g., the edge from A to B shows that A is a parent of B. Nodes in black (white) represent selected (non-selected) features.

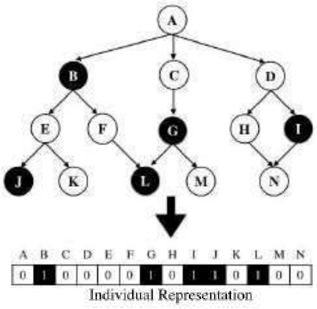


Figure 1: Example of an individual representation used by the proposed GA-HFS.

4.2 Fitness Function This function evaluates the quality of a feature subset. We employ a lexicographic multi-objective fitness function. In this approach, two or more objectives are taken into account to measure the fitness of each individual, where each objective has a distinct predefined priority. GA-HFS' lexicographic fitness function has two objectives: to maximize predictive accuracy (higher priority), measured by the Geometric Mean (GM) of Sensitivity and Specificity [6] of the classifier; and to minimize the number of selected features (lower priority). This latter objective is used only as a tie-breaking criterion in tournament selection – described below. The GM is computed by an internal 5-fold cross-validation procedure on the training set, since we follow the wrapper approach [3].

4.3 Elitism This procedure preserves the ϵ (a parameter) best individuals from the previous generation.

4.4 Selection Procedure At each generation (iteration), tournament selection [15] is used to select individuals to act as parents for the crossover and mutation operators. Tournament selection randomly samples k individuals from the population, where k (the tournament size) is a user-defined parameter. These individuals play a tournament based on the lexicographic multi-objective approach. That is, if an individual has a GM value higher than the others in the tournament, the former is selected as the tournament winner. Otherwise (i.e., the individuals in the tournament have the same GM value), to break the tie, the tournament winner is the individual with the smallest number of selected features. Tournament selection is performed as many times as needed to produce new individuals (for the next generation), until reaching the fixed population size.

4.5 Crossover Operation Each pair of individuals (parents) selected by a tournament can undergo crossover to create two child individuals. GA-HFS uses uniform crossover. For each position in the feature vector of the two parents, this operator randomly decides if the binary values in that position remain the same in each parent or are swapped between the two parents. The crossover operator is performed with a given user-defined probability.

4.6 Mutation Operation GA-HFS has two new biased mutation operators (introduced in the following section), which exploit the hierarchy of features to generate new individuals.

4.7 Stopping criteria GA-HFS runs until a fixed number of iterations has been performed or until the algorithm converges (i.e., there is no difference between the population's highest and lowest fitness values).

5 One Standard Mutation and Two Novel Hierarchical Mutation Operators

Mutation operators randomly replace the value of a gene (indicating whether or not a feature is selected) in an individual. Mutation contributes to more diversity in the population, because it can introduce new gene values that do not occur in any individual of the population. The mutation probability is a user-defined parameter (in general a relatively small value) defining the probability of mutating each gene in an individual. We propose three versions of GA-HFS, using three mutation operators (one operator per GA-HFS version): a standard mutation and two novel ones, as described next.

5.1 Bitwise Mutation This is the most common mutation operator for binary representation. This operator simply flips the bit value of a gene in an individual with a user-defined mutation probability. I.e., it flips a gene value from a selected feature (1) to a non-selected feature (0) or vice-versa.

5.2 Simple Hierarchical Elimination (SHE) Mutation The new Simple Hierarchical Elimination (SHE) mutation operator is a modified version of bitwise mutation with biased mutation probabilities, as explained below. It relies on the assumption that a feature subset with a large amount of hierarchical redundancy often decreases predictive accuracy [12, 16]. The SHE mutation aggressively removes hierarchically redundant features from a feature set. Hence, after applying this operator, the reduced feature subset is expected to have fewer hierarchically redundant features and consequently a higher fitness value.

The SHE mutation is described in Algorithm 1. It works by assigning to each feature f in the input individual a mutation probability value that depends on the selection status of that feature and its ancestors/descendants in the individual. If f is marked as selected and any of f's ancestors/descendants is also selected (involving hierarchical redundancy), then f will mutate with a biased probability (bp) – lines 2 and 3 of the algorithm. If the mutation is applied, it will remove feature f, changing its status from selected to non-selected. If the condition in line 2 is not satisfied, then f will mutate with a standard probability (sp) – lines 4 and 5. Both sp and bp are user-defined parameters, where bp should be higher than sp. Hence, the probability of removing a currently selected feature with at least one currently selected ancestor/descendant is greater than the probability of changing the status of other features in an individual. During the GA run, the SHE operator is applied many times, reducing the number of redundant features in the individuals at each generation. Therefore, in the long run, it is expected that individuals with relatively few hierarchically redundant features and higher values of fitness will be present in the last generation's population.

Algorithm 1 Simple Hierarchical Elimination (SHE) Mutation

```
Input : an $individual$, $bp$ (biased probability value) and $sp$ (standard probability value)
Output: a mutated individual
for each gene (feature) $f \in individual$ do
    if $f$ is selected and $f$ has selected ancestors/descendants then
        Mutate $f$ with biased probability $bp$
    else
        Mutate $f$ with standard probability $sp$
    end if
end for
```

5.3 Correlation-based Hierarchical Elimination (CbHE) Mutation The second new mutation operator follows the same basic principle of SHE, i.e., it sets biased mutation probabilities in order to reduce the number of hierarchically redundant selected features and to try to achieve a higher predictive accuracy. Note, however, that although SHE favors feature elimination from a candidate feature subset, it does not consider the actual degree of correlation between features. Hence, the second new operator, named Correlation-based Hierarchical Elimination (CbHE) mutation, attempts to guide the search towards a good candidate solution by setting biased mutation probabilities for the individual's selected features based on the correlation between features in the hierarchy.

CbHE assigns mutation probabilities as follows. First, it assigns to each non-selected feature in the individual a standard probability value. Second, it assigns to each gene marked as selected in the individual a biased mutation probability based on the correlation level between the feature and its ancestors/descendants also marked as selected in the individual.

The basic idea is that if a feature marked as selected in an individual is strongly (weakly) correlated with its ancestors/descendants also marked as selected in that individual, then the probability of mutating the feature's status to non-selected should be high (low). Note that, like in the SHE mutation, the decision to use a biased or standard mutation probability in the CbHE mutation varies not only across features, but also across individuals. The goal is evolving the population towards candidate solutions with few hierarchically redundant features. However, in SHE the biased mutation probability value is fixed throughout the GA run, whereas CbHE dynamically computes the biased mutation probability values in a data-driven way.

CbHE is described in Algorithm 2. For each feature f marked as selected in the individual, CbHE sets a mutation probability value based on the correlation between f and its selected ancestors/descendants. If f is selected, then CbHE calculates the average correlation between f and its selected ancestor/descendant features in the individual (lines 3 to 7). As a measure of correlation, CbHE uses the symmetrical uncertainty coefficient [10], which takes normalized values in the [0,1] range. Then, f undergoes mutation with a biased mutation probability given by the average correlation between f and its selected ancestors/descendants. In other words, a strong (weak) correlation means that there is a high (low) probability that the status of f in the individual will mutate from selected to non-selected – line 8. Note that, if f is selected and none of its ancestors/descendants is selected, then the status of f remains the same (no mutation). In contrast, if f is not selected in the individual, then CbHE assigns to f a standard mutation probability value (line 10).

Algorithm 2 Correlation-based Hierarchical Elimination (CbHE) Mutation

```
Input : an $individual$ and $sp$ (standard probability value)
Output: a mutated individual
 1: for each gene (feature) $f \in individual$ do
 2:     if $f$ is selected then
 3:         $corr \leftarrow 0$
 4:         $AD \leftarrow$ selected ancestors/descendants
 5:         for each feature $v \in AD$ do
 6:             $corr \leftarrow corr + \frac{Correlation(f,v)}{|AD|}$
 7:         end for
 8:         Mutate $f$ with biased probability $corr$
 9:     else
10:         Mutate $f$ with standard probability $sp$
11:     end if
12: end for
```

6 Computational Experiments

6.1 Datasets Following the methodology used in [19, 20], we built 28 datasets of ageing-related genes, involving the effect of genes on an organism's longevity. These datasets were built by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (version: Build 17) [13] and the Gene Ontology (GO) database (version: 2015-10-10) [17]. HAGR is a database with information about ageing- and longevity-related genes in four model organisms: *C. elegans* (worm), *D. melanogaster* (fly), *M. musculus* (mouse) and *S. cerevisiae* (yeast). The GO database provides three ontology types: biological process (BP), molecular function (MF) and cellular component (CC). Each ontology has a separate set of GO terms (features), i.e., a distinct feature hierarchy (a DAG). So, for each of the 4 model organisms, we built 7 datasets, with 7 combinations of feature types (feature hierarchies), denoted: BP, CC, MF, BP.CC, BP.MF, CC.MF, BP.CC.MF.

Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term in the GO hierarchy and a binary class variable indicating if the instance is either positive ("pro-longevity" gene) or negative ("anti-longevity" gene) according to the HAGR database. In order to avoid overfitting, GO terms annotated for less than three genes were discarded.

Information about the datasets is shown in Table 1. For each of the 4 model organisms, each of the 7 rows describes a specific dataset. The first and second columns show the organism name and the feature hierarchies used in each dataset. The other columns show the number of features (#F), the number of edges in the feature DAGs (#E), the number of instances (#I), the percentage of positive-class instances (% Pos) and the percentage of negative-class instances (% Neg).

6.2 Experimental Methodology We implemented all feature selection methods used in this work within the open-source WEKA data mining tool [5]. The Naïve Bayes (NB) from WEKA was used as the classification algorithm to evaluate the quality of the feature subsets selected by each feature selection method. NB was chosen due to its good performance on related work [16, 21] and its fast speed. The predictive accuracy was measured by 10-fold cross validation. The methods were evaluated on 24 datasets, since the 4 datasets with the BP.CC feature hierarchies were used only for tuning the parameters of all methods. Since GAs are stochastic search methods, we run GA-HFS using 10 different random seeds for each of the 10 cross-validation folds (i.e., 100 GA-HFS runs in total), and the reported results are averaged over all those 100 runs.

As shown in Table 1, the majority of the datasets have imbalanced class distributions, so we evaluated the methods' predictive accuracy by using the Geometric Mean (GM) of Sensitivity and Specificity as well as the Area Under the Precision-Recall Curve (AUCPR) measures. GM is defined as follows: $GM = \sqrt{Sensitivity * Specificity}$. Sensitivity is the proportion of positive class instances correctly predicted as positive, whereas Specificity is the proportion of negative class instances correctly predicted as negative [6]. The AUCPR plots the precision of the classifier as a function of its recall, then the area under this curve is used to evaluate the classifier's predictive accuracy (the higher the area, the better) [6].

Table 1: Detailed information about the datasets used in the experiments.

Group	Dataset	#F	#E	#I	% Pos	% Neg
C. elegans	BP	991	1707	657	34.40	65.60
	CC	178	277	484	36.36	63.64
	MF	263	331	504	37.70	62.30
	BP.CC	1169	1984	664	34.34	65.66
	BP.MF	1254	2038	663	34.24	65.76
	CC.MF	441	608	566	36.22	63.78
	BP.CC.MF	1432	2315	667	34.33	65.67
D. melanogaster	BP	800	1355	132	71.97	28.03
	CC	89	130	122	70.49	29.51
	MF	146	182	126	70.63	29.37
	BP.CC	889	1485	133	71.43	28.57
	BP.MF	945	1536	133	71.43	28.57
	CC.MF	234	311	130	70.77	29.23
	BP.CC.MF	1034	1666	133	71.43	28.57
M. musculus	BP	1333	2406	109	68.81	31.78
	CC	143	214	107	68.22	31.78
	MF	240	289	106	67.92	32.08
	BP.CC	1475	2619	109	68.81	31.19
	BP.MF	1572	2694	109	68.81	31.19
	CC.MF	382	501	109	68.81	31.19
	BP.CC.MF	1714	2906	109	68.81	31.19
S. cerevisae	BP	844	1511	331	13.29	86.71
	CC	145	230	331	13.29	86.71
	MF	221	277	331	13.29	86.71
	BP.CC	989	1741	331	13.29	86.71
	BP.MF	1065	1788	331	13.29	86.71
	CC.MF	366	507	331	13.29	86.71
	BP.CC.MF	1210	2018	331	13.29	86.71

To determine whether the differences in predictive accuracy are statistically significant, as recommended by Demsar [1], we ran the Friedman test followed by the Nemenyi post-hoc test. First, the Friedman test was executed with the null hypothesis that the accuracies of all methods are equivalent. The alternative hypothesis is that there is a difference between the accuracies of all methods as a whole. If the null hypothesis is rejected, we run the Nemenyi post-hoc test to identify pairs of methods with significantly different accuracies. Both the Friedman and Nemenyi tests were used at the 0.05 significance level.

6.3 Parameter Tuning To tune the parameter settings of all feature selection methods we used the irace tool [9]. To use irace, we selected 4 out of the 28 datasets (one from each model organism). We selected the 4 datasets with the BP.CC feature hierarchies, since they have a medium number of features. Irace was run with default parameters and a maximum budget of 250. For each feature selection method, the best parameter setting found by irace was used in the experiments to measure predictive accuracy, using the other 24 datasets. Table 2 shows the ranges of parameter settings used by the irace tool. The last three columns show the best parameter setting found by irace for each of the three GA-HFS versions, each with a different mutation operator (Bitwise, SHE and CbHE).

Table 2: Ranges of parameter settings used by irace and the best parameter setting found, for the three GA-HFS versions (each with a distinct mutation type).

GA-HFS				
Parameter	Range	Bitwise	SHE	CbHE
# Population	[50, 150]	138	62	146
# Generations	[50, 150]	80	96	149
Elitism Size	[2,10]	6	2	4
Tourn. Size	[2,10]	4	5	6
Crossover Prob.	[0.70,1.00]	0.93	0.98	0.95
Mutation Prob.	[0.01,0.10]	0.02	0.03	0.06
SHE's Prob.	[0.01,0.30]	–	0.19	–
CbHE's Prob.	auto	–	–	auto

Irace was also used to tune the parameters of ReliefF and SHSEL. These methods have only one parameter to be tuned, making their parameter tuning easier than for GA-HFS. ReliefF's parameter and SHSEL's parameter are thresholds that calibrate the number of features to be removed, and irace considered both parameters' values in the range [0.00,1.00]. The best parameter values found by irace were 0.04 and 0.98 for ReliefF and SHSEL, respectively. HIP and CFS have no parameters to be tuned.

Table 3: GM values (%) obtained by Naïve Bayes with the 8 feature selection approaches.

	Datasets	NoFS	CFS	HIP	ReliefF	SHSEL	GA-HFS	GA-HFS-SHE	GA-HFS-CbHE
C. elegans	BP	62.0	61.3	61.4	51.8	57.9	65.4	**67.2**	64.3
	CC	65.7	63.0	**68.6**	60.3	62.9	66.2	66.5	68.0
	MF	57.6	49.6	50.9	47.9	41.6	61.0	**62.4**	62.2
	BP.MF	61.9	63.5	63.2	61.5	60.1	66.8	**68.6**	65.6
	CC.MF	64.2	61.1	61.4	55.6	56.9	66.2	**67.6**	65.6
	BP.CC.MF	62.4	61.6	63.0	58.3	61.6	66.5	**68.4**	65.7
D. melanogaster	BP	59.4	58.3	66.1	39.8	53.7	64.1	66.5	**67.6**
	CC	66.7	68.1	68.4	51.2	59.1	69.7	71.8	**73.7**
	MF	58.0	52.2	57.5	55.4	46.0	54.7	**60.9**	60.3
	BP.MF	57.3	61.4	**68.1**	43.7	59.0	59.3	63.6	62.8
	CC.MF	65.8	59.1	65.7	57.4	57.1	**68.9**	65.0	67.4
	BP.CC.MF	59.4	59.5	**75.1**	63.8	60.9	60.0	66.2	66.4
M. musculus	BP	59.1	52.8	67.3	59.7	66.1	67.7	**70.4**	69.6
	CC	64.1	50.4	58.3	61.6	55.5	67.7	67.7	**69.7**
	MF	63.5	59.6	65.8	62.1	64.0	66.9	68.4	**70.3**
	BP.MF	64.9	63.6	68.4	59.5	65.1	71.4	71.1	**71.3**
	CC.MF	61.6	55.4	68.1	56.4	61.2	66.8	67.7	**69.2**
	BP.CC.MF	70.2	63.1	70.5	57.4	69.2	**74.4**	73.8	71.8
S. cerevisae	BP	61.5	63.4	68.8	68.2	49.6	68.1	70.0	**70.6**
	CC	57.6	40.9	47.8	57.1	0.00	**61.4**	57.6	**61.4**
	MF	34.2	26.1	42.8	35.6	14.1	31.8	42.0	**44.9**
	BP.MF	62.1	60.0	68.3	66.6	49.8	67.5	**69.5**	**69.5**
	CC.MF	**59.9**	36.8	57.4	51.6	25.1	58.3	55.0	58.9
	BP.CC.MF	62.8	61.4	66.4	**69.6**	53.5	67.7	68.4	68.6
	Avg. Rank	5.0	6.5	3.7	6.3	6.9	3.3	2.3	**1.9**
	#Wins	1.0	0.0	3.0	1.0	0.0	3.0	7.5	**9.5**

{GA-HFS-CbHE,GA-HFS-SHE} $\succ$ {SHSEL,CFS, ReliefF, NoFS} and {GA-HFS,HIP} $\succ$ {SHSEL,CFS,ReliefF}

6.4 Results This section compares the predictive accuracies obtained by Naïve Bayes with 8 feature selection approaches: 3 GA-HFS versions (one with standard bitwise mutation and the two others with a new mutation operator (SHE and CbHE)); two traditional (non-hierarchical) feature selection methods (CFS and ReliefF), two state-of-the-art hierarchical methods (SHSEL and HIP); and, as a baseline, Naïve Bayes using the whole feature set (NoFS).

The results for the measures GM and AUCPR are shown in Tables 3 and 4, where the first column shows the organism name and the feature hierarchies of each dataset. The other columns show the GM or AUCPR values obtained by Naïve Bayes with the aforementioned 8 feature selection approaches. The best results for each dataset are highlighted in bold type.

The last two rows of each table show, for each method, its average rank (Avg. Rank) and number of wins (#Wins). The lower the Avg. Rank, the better the performance of the method. In the row right below Table 3 and Table 4, the symbol $\succ$ means a statistically significant difference between some methods, such that $\{a\} \succ \{b, c\}$ means that a is significantly better than b and c.

Table 3 shows that GA-HFS-CbHE achieved the best average rank and the highest number of wins in terms of GM; whilst GA-HFS-SHE achieved the second best average rank and number of wins. The Friedman test detected a significant difference among the methods, and the Nemenyi test showed that the two GA-HFS versions with novel mutation operators (GA-HFS-CbHE and GA-HFS-SHE) are significantly better than SHSEL, CFS, ReliefF and NoFS. There was no significant difference between those two best GA-HFS versions and GA-HFS using bitwise mutation and the HIP method; but GA-HFS-CbHE and GA-HFS-SHE

Table 4: AUCPR values (%) obtained by Naïve Bayes with the 8 feature selection approaches.

	Datasets	NoFS	CFS	HIP	ReliefF	SHSEL	GA-HFS	GA-HFS-SHE	GA-HFS-CbHE
C. elegans	BP	55.1	55.9	58.4	50.5	56.4	57.9	59.7	**60.1**
	CC	56.3	54.0	57.2	56.5	49.8	56.8	56.6	**59.6**
	MF	50.2	48.0	50.7	50.3	45.6	53.0	**54.7**	54.4
	BP.MF	53.6	55.4	57.0	50.8	54.8	56.9	**59.1**	58.2
	CC.MF	54.8	51.2	54.2	55.7	53.1	56.1	**57.9**	56.4
	BP.CC.MF	54.0	58.1	58.1	54.7	54.7	59.2	**61.3**	60.3
D. melanogaster	BP	83.1	83.3	**87.6**	82.8	82.8	84.7	85.8	85.4
	CC	87.6	87.9	**88.6**	84.6	86.6	86.5	88.0	88.5
	MF	81.9	78.1	82.9	81.9	79.3	81.4	83.3	**84.5**
	BP.MF	84.7	82.8	84.5	79.7	84.2	86.3	**86.5**	85.9
	CC.MF	88.1	87.3	**88.6**	85.2	86.5	87.5	88.2	87.6
	BP.CC.MF	85.4	86.0	87.2	82.2	85.7	86.4	**88.4**	87.5
M. musculus	BP	82.5	82.6	86.0	81.4	85.1	87.1	**89.3**	89.1
	CC	84.5	79.5	80.0	81.2	82.3	**86.7**	**86.7**	86.0
	MF	87.1	86.0	85.5	83.8	86.0	86.7	87.5	**87.9**
	BP.MF	81.7	86.7	87.2	83.7	87.4	88.3	89.7	**90.0**
	CC.MF	85.7	82.5	84.9	82.3	84.9	87.6	88.3	**88.6**
	BP.CC.MF	83.0	86.1	88.0	83.4	87.6	88.6	89.9	**90.0**
S. cerevisae	BP	45.6	48.4	46.3	45.6	45.1	51.2	53.6	**55.1**
	CC	34.0	27.0	30.4	28.6	26.5	**38.8**	38.3	38.1
	MF	26.8	27.8	27.0	31.7	28.2	25.8	34.6	**36.9**
	BP.MF	41.8	47.3	46.0	45.0	45.3	49.3	54.5	**54.9**
	CC.MF	33.9	26.6	32.0	33.1	35.0	38.4	37.9	**39.4**
	BP.CC.MF	44.4	46.1	46.0	49.9	44.3	50.5	**53.9**	53.3
	Avg. Rank	5.8	6.0	4.3	6.6	6.1	3.5	**1.8**	**1.8**
	#Wins	0.0	0.0	3.0	0.0	0.0	1.5	8.5	**11.0**

$\{$GA-HFS-CbHE,GA-HFS-SHE$\} \succ \{$HIP,SHSEL,CFS, ReliefF, NoFS$\}$; $\{$GA-HFS$\} \succ \{$SHSEL,CFS, ReliefF, NoFS$\}$ and $\{$HIP$\} \succ \{$ReliefF$\}$

had a much better average rank and number of wins. Moreover, GA-HFS with bitwise mutation and HIP were significantly better than SHSEL, CFS and ReliefF.

Table 4 reports the AUCPR results. GA-HFS-CbHE and GA-HFS-SHE achieved the joint best average rank, but GA-HFS-CbHE had the highest number of wins followed by GA-HFS-SHE. The Friedman test detected a significant difference among the methods. The Nemenyi test showed that both GA-HFS-CbHE and GA-HFS-SHE were significantly better than HIP, SHSEL, CFS, ReliefF and NoFS. Besides, GA-HFS was significantly better than SHSEL, CFS, ReliefF and NoFS. Also, HIP was significantly better than ReliefF. There was no significant difference between GA-HFS with bitwise mutation and the two GA-HFS versions using the hierarchical mutation operators. However, the latter two methods obtained a much better number of wins and average rank than the former.

Figure 2 presents the average percentage of features selected by each method – averaged over the 24 datasets and the 10 cross-validation folds. As expected, comparing the three GA-HFS versions, the percentage of selected features is reduced when a hierarchical mutation operator (SHE or CbHE) is applied. Since GA-HFS-SHE and GA-HFS-CbHE were the two best methods regarding GM and AUCPR, these results broadly support the hypothesis that reducing the number of hierarchically redundant features increases predictive accuracy. Note that, among the three GA mutation types, SHE tends to remove more hierarchically redundant features. Indeed, GA-HFS-SHE selects on average less than half of the features selected by GA-HFS (with traditional bitwise mutation) and about two thirds of the features selected by GA-HFS-CbHE.

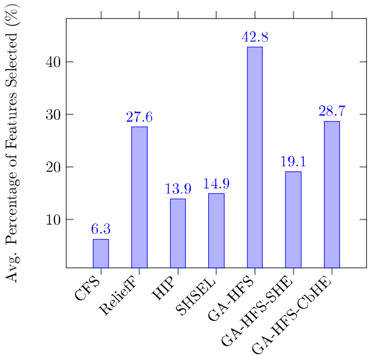


Figure 2: Average percentage (%) of features selected by the 7 feature selection methods.

Finally, we computed the relative frequency (%) of selection of each feature, out of all runs of GA-HFS-CbHE (the best method overall) on all datasets which originally included that feature. We computed these results per organism and for all organisms as a whole. To identify the most relevant features (Gene Ontology (GO) terms) for the biology of ageing, we focus on very frequently selected GO terms, but ignoring very generic, non-informative terms. Briefly, the most frequently selected GO terms include: (a) GO:0016209, "Antioxidant activity", with selection frequency over 90% for organisms yeast and worm; (b) GO:0000003, "Reproduction", with selection frequency over 92% for yeast, worm and fly; (c) GO:0045202, "Synapse" (a structure connecting neurons), with selection frequency over 86% for worm, fly and mouse. These GO terms were ranked 2nd, 4th and 6th, respectively, in terms of relative selection frequencies across all organisms.

7 Conclusions

This work has introduced three versions of a genetic algorithm (GA) for feature selection, including two novel mutation operators tailored for feature selection in hierarchical feature spaces. These two operators are based on the principle that reducing the number of hierarchically redundant features often leads to higher predictive accuracy. The first operator, Simple Hierarchical Elimination (SHE) mutation, sets a fixed biased mutation probability to each feature with hierarchical redundancy, where the probability of removing such features is greater than the probability of changing the selection status of other features. The second mutation operator, Correlation-based Hierarchical Elimination (CbHE), sets the probability of removing a hierarchically redundant feature in a data-driven way, based on the correlation among hierarchically related features.

The experiments compared the predictive accuracy of Naïve Bayes with features selected by 8 different approaches. In summary, the two proposed GAs using the two novel hierarchical mutation operators achieved better predictive accuracies than traditional and state-of-the-art hierarchical feature selection methods. Actually, those two best GAs obtained significantly higher predictive accuracy than 4 or 5 other approaches, depending on the accuracy measure. Also, those two best GAs, using new hierarchical mutation operators, selected overall substantially fewer features than the GA using a non-hierarchical mutation operator.

Acknowledgements This work was supported by CAPES research grant PDSE 88881.132229/2016-01 (P.N. da Silva) and CNPq (A. Plastino). We also acknowledge the support of IC-UFF for access to the "OSCAR" computer cluster.

References

[1] J. Demsar, *Statistical comparisons of classifiers over multiple data sets*, J Mach. Learn. Res. 7, 1–30, 2006.

[2] F. Fabris, J.P. Magalhães and A.A. Freitas, *A review of supervised machine learning applied to ageing research*, Biogerontology 17 (2), 1–8, 2017.

[3] A.A. Freitas, *Data mining and knowledge discovery with evolutionary algorithms*, Springer, 2002.

[4] M.A. Hall, *Correlation-based feature selection for discrete and numeric class machine learning*, In: Intl. Conf. on Mach. Learn. (ICML), pp. 359–366, 2000.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, *The WEKA data mining software: an update*, ACM SIGKDD Exploration Newsletter 11(1), 10–18, 2009.

[6] N. Japkowicz and M. Shah, *Evaluating learning algorithms: A classification perspective*, Cambridge University Press, 2011.

[7] Y. Jeong and S-H. Myaeng, *Feature selection using a semantic hierarchy for event recognition and type classification*, In: Intl. Joint Conf. on Natural Language Processing (IJCNLP), pp. 136–144, 2013.

[8] I. Kononenko, *Estimating attributes: analysis and extensions of RELIEF*, In: ECML, pp. 171–182., 1994.

[9] M. Lopes-Ibanez, J. Dubois-Lacoste, L.P. Caceres, M. Birattari and T. Stutzle, *The irace package: Iterated racing for automatic algorithm configuration*, Operations Research Perspectives 3, 43–58, 2016.

[10] H. Liu and H. Motoda, *Computational methods of feature selection*, CRC Press, 2008.

[11] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*, Springer, 2012.

[12] S. Lu, Y. Ye, R. Tsui, H. Su, R. Rexit, S. Wesaratchakit, X. Liu and R. Hwa, *Domain ontology-based feature reduction for high dimensional drug data and its application to 30-day heart failure readmission prediction*, In: Intl. Conf. on Collaborative Computing (Collaboratecom), pp. 478–484, 2013.

[13] J.P. 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 biogerontologistis*, Aging Cell 8(1), 65–72, 2009.

[14] J.P. Magalhães, *The biology of ageing: a primer. An introduction to gerontology*, Cambridge University Press, pp. 22–47, 2011.

[15] C.R. Reeves, *Genetic Algorithms*, In: Handbook of Metaheuristics, Springer, 2010.

[16] P. Ristoski and H. Paulheim, *Feature selection in hierarchical feature spaces*, In: DS, pp. 288–300, 2014.

[17] The GO Consortium, *Gene ontology: tool for the unification of biology*, Nat. Gen. 25(1), 25–29, 2000.

[18] 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: IEEE Intl. Conf. on Bioinformatics and Biomedicine (BIBM), pp. 373–380, 2013.

[19] C. Wan, A.A. Freitas and J.P. Magalhães, *Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods*, IEEE/ACM Trans. Comput. Biol. Bioinform. 12(2), 262–275, 2015.

[20] C. Wan and A.A. Freitas, *Two methods for constructing a gene ontology-based feature network for a bayesian network classifier and applications to datasets of ageing-related genes*, In: ACM BCB, pp. 27–36, 2015.

[21] 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.

[22] B. Xue, M. Zhang, W.N. Browne and X. Yao, *A survey on evolutionary computation approaches to feature selection*, IEEE Trans. Evol. Comp. 20(4), 606–626, 2016.

[23] M.J. Zaki and W. Meira Jr., *Data mining and analysis: fundamental concepts and algorithms*, Cambridge University Press, 2014.

APPENDIX C – Martire, I., da Silva, P.N., Plastino, A., Fabris, F., Freitas, A.A. “A Novel Probabilistic Jaccard Distance Measure for Classification of Sparse and Uncertain Data”. In Proc. of Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) (2017), pages 81-88

A novel probabilistic Jaccard distance measure for classification of sparse and uncertain data

I. Martire[1], P. N. da Silva[1], A. Plastino[1], F. Fabris[2], A. A. Freitas[2]

[1] Universidade Federal Fluminense, Brazil
igormartire@id.uff.br, {psilva, plastino}@ic.uff.br
[2] School of Computing, University of Kent, Canterbury, Kent, CT2 7NF, UK
{F.Fabris, A.A.Freitas}@kent.ac.uk

Abstract. Classification is one of the most important tasks in the data mining field, allowing patterns to be leveraged from data in order to try to properly classify unseen instances. Also, more and more often, the classification task has to be performed on datasets containing uncertain data. Although an increasing number of studies have been developed to handle uncertainty in classification in the last decade, there are still many underexplored scenarios — such as sparse data, usual in the bioinformatics field. Thus, in this work, we propose a novel distance measure for sparse and uncertain binary data based on the widely used Jaccard distance, testing its performance using the 1NN classifier. We evaluate the classification performance of our proposed method on 28 biological aging-related datasets with sparse and probabilistic binary features and compare it with a common technique to handle uncertainty by employing data transformation and traditional classification. The experimental results show that our proposed distance measure has both a smaller runtime and a better predictive performance than the traditional transformation approach.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.5.4 [**Pattern Recognition**]: Applications; J.3 [**Computer Applications**]: Life and Medical Sciences

Keywords: aging-related genes, classification, data mining, Jaccard distance, uncertain data

1. INTRODUCTION

The classification task is one of the most relevant tasks in the data mining field [Han et al. 2011]. Given a dataset of pre-labeled instances, the classification task comprises the induction of a classification model that is capable of predicting the class of an unseen instance based solely on its features. These features can have a numerical or categorical domain, with certain or uncertain values. Naturally, because of their higher prevalence, the majority of the techniques that have been developed so far focus on the handling of certain data [Aggarwal 2014].

In this work, we focus on the classification of uncertain data, specifically in sparse datasets. This kind of data can originate from many sources due to various factors, such as measurement precision limits, measurement errors, approximations or even lack of information. Even though the number of studies on classifying uncertain data has significantly increased in the last decade [Aggarwal 2014], there are still many underexplored areas, as is the case of sparse datasets.

We are particularly interested in the study of aging-related genes (represented as instances in our datasets) in order to identify the effect of genes on the longevity of an organism. These datasets commonly use binary features extracted from the Gene Ontology (GO) database [Ashburner et al. 2000], but another important type of feature are protein-protein interactions (PPIs) [Stojanova et al. 2013]. PPI features indicate whether or not an aging-related protein interacts with each of a set of other proteins (which may or may not be aging-related proteins). For that purpose, we can use the STRING database [Szklarczyk et al. 2014], a popular source of PPI datasets in the bioinformatics literature. Note, however, that instead of providing binary values for the PPIs, the STRING database provides confidence scores for each interaction. This allows the dataset to present more PPI data, but adds uncertainty to it.

When not working with uncertain data, an approach to classify genes described by binary features in the aging literature is to use the k-Nearest Neighbors classifier with the Jaccard distance [Wan et al. 2015] [Wan and Freitas 2017]. Since the Jaccard distance is not able to directly handle uncertain binary values, a data transformation procedure would be required to "remove" the uncertainty, which, of course, could cause loss of valuable data. A simple and common transformation is applying a cut-off on the PPI values, so that when the confidence score is over (below) a certain value it is converted to 1 (0). The problem would then lie on how to choose an appropriate cut-off value. However, in the bioinformatics literature there is usually no concern on optimizing this value and not even an explanation about the reasons behind its choice.

Thus, the main contribution of this work is to provide an intuitive, fast and accurate method to handle uncertain PPI data in distance-based classification. For that purpose, we propose a novel Jaccard distance measure able to handle uncertain binary features, without requiring any data transformation procedure or parameter optimization. Also, it allows the algorithm to benefit from the uncertain information available, removing the need to rely on arbitrary cut-off values or to spend much time on optimizing its value.

The remainder of this article is organized as follows. Section 2 describes the related work. In Section 3, we introduce the novel distance measure for classification in sparse datasets with probabilistic binary features. Section 4 presents the datasets used in this work. Computational results are presented in Section 5. Lastly, in Section 6, we present the conclusions and future research directions.

2. RELATED WORK

The classification of uncertain data has been extensively studied in the last two decades. Many different techniques have been adapted to handle uncertain data, such as Bayesian approaches [Ren et al. 2009], Neural Networks [Ge et al. 2010], Decision Trees [Tsang et al. 2011], k-Nearest Neighbors [Yang et al. 2015] and Support Vector Machines [Yang and Li 2009]. Most of them focus on uncertain numerical features, not specifically on binary features. Notwithstanding, very few uncertain data mining studies focus on sparse datasets, and they are usually related to other tasks, such as Frequent Itemset Mining [Xu et al. 2014].

As mentioned in the previous section, a lot of the research done so far in the bioinformatics field has simply ignored the uncertain information provided by the STRING database about PPIs. This has been done by applying *ad-hoc* cut-off values such as 0.4 [Shi et al. 2017], 0.7 [Gao et al. 2017], and 0.9 [Lin et al. 2016].

3. A NOVEL PROBABILISTIC JACCARD DISTANCE MEASURE

Distance-based classifiers use the intuitive idea that instances of the same class are more similar among themselves than among instances of other classes [Han et al. 2011]. Similarity (distance) measures, like the Jaccard index (distance), are functions that calculate how similar (distant) two objects are to (from) each other, and thus are the basis of supervised distance-based classification algorithms.

Next, we present the definitions of the traditional Jaccard measure (which cannot directly handle uncertainty, since a data transformation is needed to handle it) and of our proposed distance measure (which handles uncertainty directly).

Let s_j and $s_{j'}$ be the sets of binary features with positive value (the least frequent value for each feature) in instances j and j' respectively. The Jaccard index is defined as in Equation (1). In the special case when both s_j and $s_{j'}$ are empty, the Jaccard index is defined to be equal to 1.

$$\text{Jaccard}(s_j, s_{j'}) = \frac{|s_j \cap s_{j'}|}{|s_j \cup s_{j'}|} \tag{1}$$

And the Jaccard distance between j and j' is simply defined as:

$$\delta_{\text{Jaccard}}(j, j') = 1 - \text{Jaccard}(s_j, s_{j'}). \tag{2}$$

Note that Equation (1), and consequently Equation (2), are limited to scenarios with binary feature values without uncertainty. We then propose an extension of the Jaccard index to take into account the probability $p_i(s_j)$ of a binary feature i (of a total of n features in the dataset) belonging to s_j, i.e., having positive value in instance j. Equation (3) defines this new similarity coefficient, here called ProbJaccard (Probabilistic Jaccard measure). Again, we define ProbJaccard(s_j,$s_{j'}$) = 1 when the denominator evaluates to zero, which happens when both sets are certainly empty.

$$\text{ProbJaccard}(s_j, s_{j'}) = \frac{\sum_{i=1}^{n}[p_i(s_j) \times p_i(s_{j'})]}{\sum_{i=1}^{n}[p_i(s_j) + p_i(s_{j'}) - p_i(s_j) \times p_i(s_{j'})]} \tag{3}$$

Like Equation (1), the numerator of Equation (3) measures the degree of *intersection* between the two instances, while the denominator measures the degree of *union* between the two instances. Note however, that these degrees of intersection and union are *probabilistic* in Equation (3).

Analogously, we define the Probabilistic Jaccard distance between j and j' as:

$$\delta_{\text{ProbJaccard}}(j, j') = 1 - \text{ProbJaccard}(s_j, s_{j'}). \tag{4}$$

Note that all these indexes and distances take values in the interval [0,1]. Also note that, when working with certain data, Equations (3) and (4) become equivalent to Equations (1) and (2), and, thus, they can be used in datasets with both certain and uncertain binary features.

4. EXPERIMENTAL DATASETS

We use 28 datasets of aging-related genes, where instances are genes and the binary class indicates whether or not the genes are related to longevity. These datasets were created by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (version: 335 Build 17) [de Magalhães et al. 2009] and the Gene Ontology (GO) database (version: 2015-10-10) [Ashburner et al. 2000]. HAGR is a database of aging- and longevity-associated genes in model organisms which provides aging information for genes from four model organisms: *C.elegans* (worm), *D.melanogaster* (fly), *M.musculus* (mouse) and *S.cerevisiae* (yeast). The GO database provides information about three ontology types: biological process (BP), molecular function (MF) and cellular component (CC). Each ontology contains a separate set of GO terms (features). So, for each of the four model organisms, we created seven datasets, with seven combinations of feature types, denoted by BP, CC, MF, BP.CC, BP.MF, CC.MF, and BP.CC.MF.

Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term and a binary class variable indicating if the instance is either positive ("pro-longevity" gene) or negative ("anti-longevity" gene) according to the HAGR database. These GO features values are highly sparse and, in order to avoid overfitting, GO terms which occurred in less than three genes were discarded, avoiding the use of rare features with very little statistical support and virtually no generalization power for our set of genes.

Finally, as a contribution to the aging-related genes classification problem, in order to improve the predictive performance achieved when using only GO terms [Wan et al. 2015], we added protein-protein interactions (PPIs) uncertain data from the STRING database (version: 10) [Szklarczyk et al. 2014] to each of the 28 datasets. The data is also highly sparse and, as we did with the GO features, we also filtered out the PPIs that only occurred in less than three genes.

These PPI features values were obtained, in the STRING database, from the *combined_score* field in the *network.node_node_links* table. Their values $s \in [0, 1]$ indicate the degree of confidence of their correspondent interactions. We use these values under a probabilistic perspective, where the features can be seen as binary ones (with the value 0 (1) indicating absence (presence) of the correspondent PPI in that instance's set of PPIs) and their values are represented by a probability distribution function f, defined as $f(1) = s$ and $f(0) = 1 - s$.

Table I shows statistics for each dataset, including information on their sparsity. For each of the four model organisms, each of the seven rows shows information about a specific dataset. The first column identifies the model organism. The second column shows the selected Gene Ontologies on the dataset. The other columns show, respectively, the number of features, the number (and percentage) of GO features, the number of PPI features, the average percentage of GO features with value 0 in an instance, the average percentage of PPI features with value 0 in an instance, the number of instances, the number (and percentage) of positive-class instances and the number of negative-class instances. For example, for the *C. elegans* dataset with GO terms of the Biological Process (BP) ontology type only (first row), out of the 12,438 features, 991 (7.97%) are GO features and the remaining 11,447 (92.03%) are PPI features. Also, the column "avg. % GO = 0" shows that, on average, an instance of that dataset has 95.48% of its GO features with value 0 and the column "avg. % PPI = 0" shows that, on average, an instance of that dataset has 95.32% of its PPI features with value 0. Finally, the last three columns show that this dataset has 657 instances, from which only 226 (34.40%) are labeled *positive* (Pos) and the remaining 431 (65.60%) are labeled *negative* (Neg).

5. EXPERIMENTS

In our datasets, as shown in Table I, the distribution of instances belonging to the two classes is imbalanced. Then, if the simple accuracy measure (the percentage of correctly classified instances) had been used, it would provide us with misleading performance evaluation since we could trivially obtain a high accuracy (but no useful model) by predicting the majority class for all instances [Japkowicz and Shah 2011]. Hence, we evaluate the predictive performance of the classifiers by using the value of Geometric mean (Gmean), defined as $\mathbf{Gmean} = \sqrt{Sens \times Spec}$, which takes into account the balance of the classifiers's sensitivity (Sens) and specificity (Spec) [Japkowicz and Shah 2011]. Sensitivity (specificity) means the proportion of pro-longevity (anti-longevity) genes that were correctly predicted as pro-longevity (anti-longevity) in the testing dataset [Altman and Bland 1994]. The reported Gmean value for each dataset is the average of all the 10 Gmean values generated by the well-known stratified 10-fold cross-validation procedure [Witten et al. 2016].

In this work, we use the 1-Nearest Neighbor (1NN) classifier since, in previous work, it has been shown effective for classification in aging-related datasets [Wan et al. 2015][Wan and Freitas 2017].

We start by testing the improvement in predictive performance when the PPI features are added to the original database composed of GO terms only. Since this inserted data is uncertain and the Jaccard distance does not handle uncertain values, we decided to use, as a baseline, a 5-fold Internal

Table I: Statistics for each dataset.

Organism	Dataset	# features	# (%) GO features	# PPI features	avg. % GO = 0	avg. % PPI = 0	# instances	# (%) Pos	# Neg
C. elegans	BP	12438	991 (7.97)	11447	95.48	95.32	657	226 (34.40)	431
	CC	11163	178 (1.59)	10985	93.35	94.63	484	176 (36.36)	308
	MF	11151	263 (2.36)	10888	94.93	94.58	504	190 (37.70)	314
	BP.CC	12626	1169 (9.26)	11457	95.47	95.35	664	228 (34.34)	436
	BP.MF	12733	1254 (9.85)	11479	95.65	95.35	663	227 (34.24)	436
	CC.MF	11731	441 (3.76)	11290	95.01	94.87	566	205 (36.22)	361
	BP.CC.MF	12912	1432 (11.09)	11480	95.62	95.37	667	229 (34.33)	438
D. melanogaster	BP	7359	800 (10.87)	6559	91.68	91.11	132	95 (71.97)	37
	CC	6549	89 (1.36)	6460	86.98	90.85	122	86 (70.49)	36
	MF	6698	145 (2.16)	6553	92.28	90.92	126	89 (70.63)	37
	BP.CC	7503	889 (11.85)	6614	91.38	91.20	133	95 (71.43)	38
	BP.MF	7559	945 (12.50)	6614	91.89	91.20	133	95 (71.43)	38
	CC.MF	6817	234 (3.43)	6583	90.72	91.17	130	92 (70.77)	38
	BP.CC.MF	7648	1034 (13.52)	6614	91.56	91.20	133	95 (71.43)	38
M. musculus	BP	11513	1332 (11.57)	10181	89.35	90.04	109	75 (68.81)	34
	CC	10236	142 (1.39)	10094	83.20	90.11	107	73 (68.22)	34
	MF	10323	240 (2.32)	10083	90.27	89.86	106	72 (67.92)	34
	BP.CC	11655	1474 (12.65)	10181	88.79	90.04	109	75 (68.81)	34
	BP.MF	11753	1572 (13.38)	10181	89.53	90.04	109	75 (68.81)	34
	CC.MF	10563	382 (3.62)	10181	87.93	90.04	109	75 (68.81)	34
	BP.CC.MF	11895	1714 (14.41)	10181	89.03	90.04	109	75 (68.81)	34
S. cerevisiae	BP	6305	844 (13.39)	5461	94.65	92.25	331	44 (13.29)	287
	CC	5606	145 (2.59)	5461	89.96	92.25	331	44 (13.29)	287
	MF	5682	221 (3.89)	5461	94.27	92.25	331	44 (13.29)	287
	BP.CC	6450	989 (15.33)	5461	93.96	92.25	331	44 (13.29)	287
	BP.MF	6526	1065 (16.32)	5461	94.57	92.25	331	44 (13.29)	287
	CC.MF	5827	366 (6.28)	5461	92.56	92.25	331	44 (13.29)	287
	BP.CC.MF	6671	1210 (18.14)	5461	94.02	92.25	331	44 (13.29)	287

Cross-Validation (ICV) method (accessing the training set only) to automatically choose a cut-off value to discretize the feature (feature values greater or equal than the cut-off are set to 1 and set to 0 otherwise). This ICV is performed in each iteration of the external cross-validation procedure. This baseline method is here called Jaccard-ICV. Applying this cut-off on the uncertain data allows us to convert it to certain binary values and then use it with the 1NN classifier using the traditional Jaccard distance.

The STRING database online search interface suggests four cut-off values: 0.15, 0.40, 0.70 and 0.90, meaning, respectively, low, medium, high and highest confidence. These values have also been extensively employed in the related literature [Lin et al. 2016] [Shi et al. 2017] [Gao et al. 2017]. For these two reasons, the ICV focused on choosing the best out of these four cut-off values.

The results are shown in Table II, where the boldface numbers denote the highest Gmean value obtained for each dataset. The first two columns are the same as in Table I, explained in the previous section. The third column shows the Gmean values obtained by the 1NN classifier in datasets with GO features only and using the traditional Jaccard distance metric. The fourth and fifth columns show the values obtained with the classification on the datasets composed of both GO and PPI data. While the fourth column shows the results with the internal cross-validation approach explained above, the fifth column shows the results obtained when using 1NN with our new proposed distance measure. Each row represents a different dataset in the same way as in Table I. Table II, however, has two additional rows. The second to last row, Average Rank, shows the average rank obtained by each method over the 28 datasets. For each dataset, the best method receives the ranking value of 1; conversely, the worst method receives the ranking value of 3. So, the smaller the average rank of a method, the better its overall predictive performance. Finally, the last row, #Wins, shows the number of datasets where each method has obtained the best predictive performance. Again, the boldface numbers denote the best result in each of these two rows.

Table II: Comparison of predictive performance using Gmean as evaluation measure.

		GO	GO + PPI	
Group	Dataset	Jaccard	Jaccard-ICV	Prob-Jaccard
C. elegans	BP	55.91	**65.30**	64.13
	CC	59.73	61.01	**63.21**
	MF	53.47	64.86	**66.41**
	BP.CC	61.14	65.25	**66.44**
	BP.MF	58.07	**67.15**	65.19
	CC.MF	60.33	**63.12**	62.30
	BP.CC.MF	58.11	**68.49**	66.25
D. melanogaster	BP	**64.17**	52.39	61.13
	CC	70.44	**72.08**	68.03
	MF	50.65	**60.52**	58.13
	BP.CC	61.87	55.05	**65.19**
	BP.MF	62.88	63.24	**63.81**
	CC.MF	58.69	64.14	**64.93**
	BP.CC.MF	62.57	**65.49**	63.30
M. musculus	BP	62.98	**68.31**	63.07
	CC	50.74	56.27	**63.95**
	MF	53.94	65.64	**69.18**
	BP.CC	**61.84**	55.56	56.81
	BP.MF	63.81	**66.29**	65.30
	CC.MF	56.61	67.23	**68.89**
	BP.CC.MF	62.27	**63.49**	58.51
S. cerevisiae	BP	53.69	57.34	**58.26**
	CC	50.61	53.56	**61.45**
	MF	40.34	58.69	**58.99**
	BP.CC	58.32	55.88	**65.39**
	BP.MF	51.03	57.83	**58.29**
	CC.MF	41.56	**63.74**	60.73
	BP.CC.MF	53.60	57.32	**62.88**
Average Rank		2.71	1.75	**1.54**
# Wins		2	11	**15**

The results in Table II show that Prob-Jaccard, which uses our proposed distance measure, achieves the best predictive performance on 15 datasets, followed by Jaccard-ICV (best results on 11 datasets) and Jaccard (2 datasets). To determine whether the differences in performance are statistically significant, we ran the non-parametric Friedman test followed by the Nemenyi test [Japkowicz and Shah 2011]. Both tests were used at the 0.05 significance level. The Friedman test indicated that there was at least one pair of classifiers with a statistical difference in the predictive performance. Hence, we employed the post-hoc Nemenyi test to discover in which pairs this difference occurs. The Nemenyi test showed that both Prob-Jaccard and Jaccard-ICV are significantly superior to Jaccard-GO, which does not include PPI features. However, even though Prob-Jaccard achieves both a better average rank and a higher number of wins than Jaccard-ICV, the difference in the performance between Prob-Jaccard and Jaccard-ICV was not statistically significant.

One could think of using the Euclidean distance with the 1NN classifier by using the probability values as features values, thus leading to a scenario with "certain" numerical features instead of uncertain binary ones. A preliminary experiment using this strategy has been performed, obtaining very poor results when compared to the other two methods explored in this article. These results are somewhat intuitive, since the Euclidean distance is known to be weakly discriminant for multidimensional and sparse data, and also because treating a probability as just a numeric value can lead to wrong assumptions. As an example, think of the case when comparing the distance between two instances with a single uncertain binary feature, and assume this feature's values for both instances are represented by the same probability distribution function f, for which $f(0) = f(1) = 0.5$. The Euclidean distance between these two instances would be zero, even though, if we assume that the (unknown) true value of a feature is binary (an assumption that may or may not be appropriate depending on the application domain), there is a 50% chance that these two instances have the opposite binary values for their single feature.

Based on the conducted experiments, we can notice a great improvement by simply adding the PPI features and optimizing the choice of cut-off value for each fold via internal cross-validation. However, this approach is slow, which could be a big problem when working with larger datasets. We then compared the runtime performance of the Jaccard-ICV method with our proposed Prob-Jaccard method to demonstrate how much faster this proposed method can be in comparison to the internal cross-validation one, without losing in overall predictive performance (and actually improving it most of the times). These results are presented in Table III. In this table, the first two columns are exactly the same as the ones in the previous tables. The third and fourth columns show the average time in seconds that was taken to classify a fold in the 10-fold cross-validation procedure. Notice that the reported times for the Jaccard-ICV method include the time spent in the selection of the cut-off value. The last column shows the ratio of the values in the third column to the values in the fourth column, which indicates how many times faster the Prob-Jaccard method is in comparison to Jaccard-ICV. These times were measured in a computer with 1.6 GHz Intel Core i5 processor and 4 GB 1600 MHz DDR3 memory.

Table III: Comparison of average CPU time in seconds per cross-validation fold.

		GO + PPI		
Group	Dataset	Jaccard-ICV	Prob-Jaccard	$\frac{\text{Jaccard-ICV}}{\text{Prob-Jaccard}}$
C. elegans	BP	18.048	0.835	21.614
	CC	9.577	0.399	24.003
	MF	10.802	0.438	24.662
	BP.CC	20.492	0.845	24.251
	BP.MF	20.316	0.861	23.596
	CC.MF	13.394	0.594	22.549
	BP.CC.MF	21.181	0.855	24.773
D. melanogaster	BP	0.639	0.023	28.684
	CC	0.492	0.018	24.200
	MF	0.469	0.016	25.059
	BP.CC	0.705	0.020	27.500
	BP.MF	0.676	0.021	29.905
	CC.MF	0.527	0.020	25.667
	BP.CC.MF	0.718	0.022	30.700
M. musculus	BP	0.639	0.023	27.783
	CC	0.492	0.018	27.333
	MF	0.469	0.016	29.313
	BP.CC	0.705	0.020	35.250
	BP.MF	0.676	0.021	32.190
	CC.MF	0.527	0.020	26.350
	BP.CC.MF	0.718	0.022	32.636
S. cerevisiae	BP	3.796	0.112	33.893
	CC	3.321	0.097	34.237
	MF	3.274	0.105	31.181
	BP.CC	4.008	0.116	34.552
	BP.MF	3.925	0.118	33.263
	CC.MF	3.488	0.108	32.296
	BP.CC.MF	4.190	0.126	33.254
Average		5.314	0.233	28.596

The last row of Table III shows that, on average, the Jaccard-ICV approach took 5.3 seconds to classify a single fold, while Prob-Jaccard took only 0.2 seconds. The last column in that row shows that the Prob-Jaccard approach was able to classify a single fold 28.6 times faster on average.

6. CONCLUSIONS

In this work, we presented a novel Jaccard distance measure for nearest-neighbor classification in sparse datasets with probabilistic binary features. We compared both the speed and the predictive performance of the 1NN classifier using both our novel distance measure and the traditional Jaccard distance (by applying an internal cross-validation to optimize the cut-off value).

The 1NN classifier using the proposed ProbJaccard distance measure is significantly faster than the Jaccard-ICV method. This is due to the fact that ProbJaccard handles the uncertainty from the data directly, so there is no need to perform an internal cross-validation to optimize a cut-off parameter. Additionally, the proposed ProbJaccard method has shown an overall improvement in the predictive performance of the 1NN classifier across 28 aging-related datasets, with a better average rank and higher number of wins when compared with the Jaccard-ICV method and a dataset with GO terms only, as shown in Table II; even though there was no statistically significant difference between the results of ProbJaccard and Jaccard-ICV.

Finally, this new distance measure can be extended to handle categorical features with more general types of uncertain values in sparse classification datasets. We leave this research for future work.

REFERENCES

Aggarwal, C. C. *Data classification: algorithms and applications.* CRC Press, 2014.

Altman, D. G. and Bland, J. M. Diagnostic tests 1: sensitivity and specificity. *British Medical Journal* 308 (6943): 1552, 1994.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., et al. Gene Ontology: tool for the unification of biology. *Nature genetics* 25 (1): 25–29, 2000.

de Magalhães, J. P., Budovsky, A., Lehmann, G., Costa, J., Li, Y., Fraifeld, V., and Church, G. M. The Human Ageing Genomic Resources: online databases and tools for biogerontologists. *Aging cell* 8 (1): 65–72, 2009.

Gao, Y., Xu, D., Zhao, L., and Sun, Y. The DNA damage response of C. elegans affected by gravity sensing and radiosensitivity during the Shenzhou-8 spaceflight. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis* 795 (1): 15–26, 2017.

Ge, J., Xia, Y., and Nadungodage, C. UNN: a neural network for uncertain data classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Hyderabad, India, pp. 449–460, 2010.

Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques.* Morgan Kaufmann, 2011.

Japkowicz, N. and Shah, M. *Evaluating learning algorithms: a classification perspective.* Cambridge University Press, 2011.

Lin, D., Zhang, J., Li, J., Xu, C., Deng, H.-W., and Wang, Y.-P. An integrative imputation method based on multi-omics datasets. *BMC Bioinformatics* 17 (1): 247, 2016.

Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R., and Cheung, D. Naive Bayes Classification of Uncertain Data. In *IEEE International Conference on Data Mining.* Miami, United States of America, pp. 944–949, 2009.

Shi, J., Zhang, Y., Qi, S., Liu, G., Dong, X., Huang, N., Li, W., Chen, H., and Zhu, B. Identification of potential crucial gene network related to seasonal allergic rhinitis using microarray data. *European Archives of Oto-Rhino-Laryngology* 274 (1): 231–237, 2017.

Stojanova, D., Ceci, M., Malerba, D., and Dzeroski, S. Using PPI network autocorrelation in hierarchical multi-label classification trees for gene function prediction. *BMC Bioinformatics* 14 (1): 285, 2013.

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., et al. STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Research* 43 (D1): D447–D452, 2014.

Tsang, S., Kao, B., Yip, K. Y., Ho, W.-S., and Lee, S. D. Decision trees for uncertain data. *IEEE transactions on knowledge and data engineering* 23 (1): 64–78, 2011.

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

Wan, C., Freitas, A. A., and de Magalhães, J. P. 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, 2015.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2016.

Xu, J., Li, N., Mao, X.-J., and Yang, Y.-B. Efficient probabilistic frequent itemset mining in big sparse uncertain data. In *Pacific Rim International Conference on Artificial Intelligence.* Gold Coast, Australia, pp. 235–247, 2014.

Yang, J.-L. and Li, H.-X. A probabilistic support vector machine for uncertain data. In *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications.* Hong Kong, China, pp. 163–168, 2009.

Yang, L., Chen, H., Cui, Q., Fu, X., and Zhang, Y. Probabilistic-KNN: A novel algorithm for passive indoor-localization scenario. In *IEEE Vehicular Technology Conference.* Glasgow, United Kingdom, pp. 1–5, 2015.

APPENDIX D – da Silva, P.N., Plastino, A., Fabris, F., Freitas, A.A. “A Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti- Longevity Genes” - Submitted to IEEE/ACM Transactions on Computational Biology and Bioinformatics

A Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti-Longevity Genes

Pablo Nascimento da Silva, Alexandre Plastino, Fabio Fabris, and Alex A. Freitas

Abstract—Understanding the ageing process is a very challenging problem for biologists. To help in this task, there has been a growing use of classification methods (from machine learning) to learn models that predict whether a gene influences the process of ageing or promotes longevity. One type of predictive feature often used for learning such classification models is Protein-Protein Interaction (PPI) features. One important property of PPI features is their uncertainty, i.e., a given feature (PPI annotation) is often associated with a confidence score, which is usually ignored by conventional classification methods. Hence, we propose the Lazy Feature Selection for Uncertain Features (LFSUF) method, which is tailored for coping with the uncertainty in PPI confidence scores. In addition, following the lazy learning paradigm, LFSUF selects features for each instance to be classified, making the feature selection process more flexible. We show that our LFSUF method achieves better predictive accuracy when compared to other feature selection methods that either do not explicitly take PPI confidence scores into account or deal with uncertainty globally rather than using a per-instance approach. Also, we interpret the results of the classification process using the features selected by LFSUF, showing that the number of selected features is significantly reduced, assisting the interpretability of the results. The datasets used in the experiments and the program code of the LFSUF method are freely available on the web at http://github.com/pablonsilva/FSforUncertainFeatureSpaces.

Index Terms—Ageing, Classification, Feature Selection, Uncertain Features, Gene Ontology, Protein-Protein Interaction

1 Introduction

Ageing is a complex process characterized by a continuous decline in the function of an organism that occurs with increasing age [9], ultimately leading to death. Even for related species, the speed at which such functional deterioration happens differs to some extent [8]. Although ageing research has advanced significantly in the last decades, it is still unclear which biological mechanisms contribute to the ageing process, even though genetic factors clearly make a major contribution to it [30].

Experiments in model organisms have identified several hundred genes that influence the ageing process (speeding it up or slowing it down) [19]. The discovery of such genes in model organisms may lead to the identification of homologous genes in humans which could lead to pharmacological interventions to treat ageing. Hence, it is particularly interesting to automatically classify genes (or proteins) in two different classes: pro-longevity and anti-longevity genes. Pro-longevity genes are those genes whose decreased expression reduces lifespan and/or those whose over-expression extends lifespan [27], [28]. Conversely, anti-longevity genes are those genes whose decreased expression extends lifespan and/or those whose over-expression decreases it.

Gene Ontology (GO) terms [23] have been widely used as predictive features for building models for the classification of pro-/anti-longevity genes [4], [21], [24], [25], [26], [27]. However, there are many other characteristics of genes (or proteins) that could be useful to the problem described in this work. So, in this work, we build predictive models using not only GO term features, but also Protein-Protein Interaction (PPI) features [20], a widely used characteristic of proteins that could potentially help finding those proteins linked with ageing [5], [17]. In a PPI dataset, each PPI indicates whether or not a protein (instance, or object to be classified) interacts with another protein. As PPI information is an important indicator of gene functions, the use of PPI features may improve the classifier's predictive accuracy. Also, as no protein works in isolation, the analysis of highly predictive PPI features could improve the interpretability of the classification model, leading to a better understanding of the ageing problem in general.

However, the use of PPI features for classification is not straight-forward. First, the values of PPI features are uncertain, i.e., such values are numeric scores representing the likelihood of interaction of two proteins (e.g., protein-A interacts with protein-B in 90% of the documented cases). Second, among the vast number of possible protein interactions, few interactions are realised, leading to a high feature sparsity and dimensionality. Note that the addition of PPI features brings a major challenge: the selection of the subset of protein interactions that are most suitable to perform an accurate prediction.

Given the previously described challenges of using PPI features and the fact that the quality of the feature set used to build a classification model has an enormous impact on its predictive accuracy [12], [13], *feature selection* methods can be used to cope with this problem. This type of methods

- *P.N. da Silva and A. Plastino are with the Institute of Computing, Universidade Federal Fluminense, Niterói, RJ, Brazil. E-mail: {psilva, plastino}@ic.uff.br*
- *F. Fabris and A.A. Freitas are with School of Computing, University of Kent, UK. E-mail: {fabiofabris@gmail.com, a.a.freitas@kent.ac.uk}*

aims at improving the predictive accuracy of the classifier by selecting a subset of relevant features. So, in this work, we propose a novel feature selection method tailored to deal with uncertain features, and we evaluate the proposal on uncertain features that represent interactions between proteins. As an additional contribution, we evaluated the effectiveness of combining GO and PPI features to predict a gene's effect on an organism's longevity. We also interpret the results of our method, showing that it can be a source of new biological insight.

This work is organised as follows. Section 2 reviews background and related work. Section 3 presents our novel feature selection method for uncertain features. Section 4 presents the results of experimental evaluations. Lastly, conclusions are presented in Section 5.

2 METHODS

2.1 Classification on Uncertain Feature Spaces

A classification problem can be formally defined as follows. Let $X = \{X_1, X_2, \ldots, X_d\}$ be the set of predictive features, where $d \geq 1$, and $C = \{C_1, C_2, \ldots, C_q\}$ be the finite set of possible classes, where $q \geq 2$. Given a training set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where each instance i is associated with a class value $y_i \in C$ and a feature vector $x_i = \{(x_{i1}, x_{i2}, \ldots, x_{id})\}$, where each x_{ij} represents the value of the feature X_j in the instance i, the goal in the classification task is to learn a classifier $h(X) \rightarrow y$ from D that, given an unlabelled instance $E = (x, ?)$, is capable of predicting its class y.

In this work, the uncertain feature space is defined as follows. Given an instance $x_i = \{(x_{i1}, x_{i2}, \ldots, x_{id})\}$, each value x_{ij}, where $0 \leq x_{ij} \leq 1$, represents a certainty score defining how likely the i-th instance has a positive feature value, which indicates that the protein represented by that instance interacts with the protein associated with the j-th feature. That is, if $x_{i1} > x_{i2}$, this means that the i-th instance is more likely to be positively associated with the first feature than to the second feature. Note that this certainty score is not necessarily a standard probability.

2.2 Feature Selection

Feature selection can be defined as finding a feature subset $F \subseteq X$, such that the predictive model $h(F)$ has a higher predictive accuracy than $h(X)$. It usually involves the removal of irrelevant or redundant features.

Feature selection methods, as a type of data preprocessing method, can be categorized into wrapper and filter methods [12], [13]. Wrapper methods measure the relevance of a feature subset by assessing the predictive accuracy of a classifier built using that subset. Hence, they select features tailored to the target classification algorithm, but they tend to be very time-consuming. By contrast, filter methods evaluate the predictive power of features generally, by using a relevance measure that is independent of the target classification algorithm. Filter methods tend to be much faster and more scalable than wrapper methods.

Feature selection and classification methods can also be categorized as eager or lazy. Eager methods select a single subset of features based on the training instances. Then, a model trained with the selected features is used to predict the class of any test instance. By contrast, lazy methods select a feature subset tailored for each test instance [1], [16], by observing the feature values in that test instance. Hence, lazy learning methods use one classification model for each testing instance, while eager methods build a single classifier for all testing instances.

The feature selection method proposed in this work (in Section 3) is a filter method that follows the lazy paradigm.

2.3 Related Work

Although uncertain features are present in many different applications (such as sensor data, biology data, among others), there are very few feature selection methods capable of exploring uncertain features in the literature. For instance, in [11], a feature selection method for graph classification is introduced. It deals with graphs where the linkage of nodes are fundamentally uncertain (i.e., each connection between two nodes holds a likelihood of being a real connection). The graph structure used in that work is broadly similar to those found in PPI networks. Note, however, that we are not interested in finding graph subsets, which is a significant difference between their approach and the one reported here.

Another feature selection method for uncertain data was proposed by [3]. They introduced a modified mutual information evaluation measure capable of dealing with uncertain features in two steps. First, each feature is evaluated by the modified mutual information measure. Second, a threshold is used to select the x% of features with better mutual information values to build the classifier. However, the uncertain data employed is quite different from the one described in this work, since each feature value is described by a Gaussian distribution. Another significant difference is the fact that the data used to build the classification model is not initially uncertain. The Gaussian distribution is built as follows. First, the real feature values are used as the mean of the distribution, and a user-defined parameter is used as the standard deviation of the distribution (this parameter is equal for every instance/feature in the dataset). Note that, this approach is very different from the method described in our work, where the uncertainty information is given as an input. Also, apart from having a hard effort to tune user-defined parameter, it has another significant drawback: it cannot handle high-dimensional data since it relies on a Kernel Density Estimation (KDE) to compute the mutual information, which is notably a computationally expensive method [3].

2.4 Data Preparation

2.4.1 *Gene Ontology (GO)*

The Gene Ontology (GO) database [23] annotates genes using terms from a expert-defined ontology. These annotations are from three different types: (i) Molecular Function (MF), which describes the molecular activities of individual genes; (ii) Cellular Component (CC), which contains information about where the gene products are active; and (iii) Biological Process (BP), containing the pathways and more general processes to which that gene product's activity contributes.

2.4.2 Protein-Protein Interections (PPI)

Protein-Protein Interactions (PPI) are defined as physical contacts (or functional interactions) between proteins that occur in a cell of a living organism [20]. There are many different databases describing interactions between proteins. In this work, we use the STRING database [22], which contains a collection of known and predicted protein-protein interactions. These interactions can be either direct (physical interaction) or indirect (functional interaction). The information available in this database comes from the following sources: computational prediction, lab experiments, knowledge transfer between organisms, automated text mining and from interactions observed in other databases. In the STRING database, each PPI is associated with a score calculated from the information in the database which indicates the confidence of certain interaction being actually present. I.e., a high confidence score means that there is more support regarding a given interaction in the database.

2.4.3 Building Datasets

We generated 28 datasets of ageing-related genes, using a similar methodology described in ([21], [25], [27]), concerning the effect of genes on an organism's longevity. These datasets were created by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (version: Build 17) ([14]), the Gene Ontology (GO) database (version: 2015-10-10) ([23]) and Protein-Protein Interactions from the STRING database ([22]). HAGR is a database of ageing- and longevity-associated genes in model organisms which provides ageing information for genes from four model organisms: *C. elegans* (worm), *D. melanogaster* (fly), *M. musculus* (mouse), and *S. cerevisiae* (yeast). As described earlier, the GO database provides three types of GO terms (features): biological process (BP), molecular function (MF) and cellular component (CC). So, for each of the 4 model organisms, we created 7 datasets, with 7 combinations of GO types, denoted by BP, CC, MF, BP.CC, BP.MF, CC.MF and BP.CC.MF. In each of these datasets, for each gene (instance), we incorporated the PPI features according to the data available in the STRING database.

Hence, each dataset contains instances (genes) from a single model organism. Each instance is formed by a set of binary features indicating whether or not the gene is annotated with each GO term, a set of uncertain features containing the score of each PPI and a binary class variable indicating if the instance is either positive ("pro-longevity" gene) or negative ("anti-longevity" gene) according to the HAGR database. To reduce overfitting, GO terms and PPI features with low support (annotating less than 3 genes) were removed from the dataset. Also, genes with no positive GO feature value were discarded. Thus, the number of instances of a dataset for a given model organism may vary across types of GO terms. Likewise, the number of GO terms vary across model organisms.

Information about the 28 datasets (7 datasets for each of 4 organisms) is shown in Table 1. The first column shows the organism associated with each dataset. The other columns show, respectively, the number of instances (#Inst), the number of features (#Feat), the number of GO terms (#GO), the number of PPI features (#PPI) and the proportion of instances from the positive ("pro-longevity") class (%P Class).

TABLE 1
Detailed information about the datasets used in the experiments.

	Dataset	#Inst	#Feat	#GO	#PPI	%P Class
C. elegans	BP	657	12438	990	11447	34.4
	CC	484	11163	177	10985	36.4
	MF	504	11151	262	10888	37.7
	BP.CC	664	12625	1167	11457	34.3
	BP.MF	663	12732	1252	11479	34.2
	CC.MF	566	11730	439	11290	36.2
	BP.CC.MF	667	12910	1429	11480	34.3
D. melanogaster	BP	132	7359	799	6559	72.0
	CC	122	6549	88	6460	70.5
	MF	126	6698	144	6553	70.6
	BP.CC	133	7502	887	6614	71.4
	BP.MF	133	7558	943	6614	71.4
	CC.MF	130	6816	232	6583	70.7
	BP.CC.MF	133	7646	1031	6614	71.4
M. musculus	BP	109	11513	1331	10181	68.8
	CC	107	10236	141	10094	68.2
	MF	106	10323	239	10083	67.9
	BP.CC	109	11654	1472	10181	68.8
	BP.MF	109	11752	1570	10181	68.8
	CC.MF	109	10562	380	10181	68.8
	BP.CC.MF	109	11893	1711	10181	68.8
S. cerevisiae	BP	331	6305	843	5461	13.3
	CC	331	5606	144	5461	13.3
	MF	331	5682	220	5461	13.3
	BP.CC	331	6449	987	5461	13.3
	BP.MF	331	6525	1063	5461	13.3
	CC.MF	331	5826	364	5461	13.3
	BP.CC.MF	331	6669	1207	5461	13.3

3 A Novel Lazy Feature Selection Method for Uncertain Features

We propose a new feature selection filter method called Lazy Feature Selection for Uncertain Features (LFSUF). The intuition behind this method is as follows. In the handled uncertain data, a feature value with high confidence (i.e., a feature value around one) means that the positive feature value has strong evidence of being actually present in an instance. Conversely, a feature value with low confidence (i.e., a feature value around zero) means that the feature is probably *not* present. Hence, LFSUF aims to select the subset of features whose positive value has the highest confidence (i.e., the highest likelihood of being present) in each test instance (adopting the lazy learning paradigm). Furthermore, the proposed method aims at selecting the subset of features which best correlates with the target class. In summary, the strategy aims at selecting, for each test instance, a subset of features with high confidence that also correlates well with the class.

LFSUF works as follows. In a preliminary step, the LFSUF evaluates the relevance r_i, using the F-Statistics (FStat) [29] of each feature $X_i \in X$. Then, we build a subset of features $F \in X$ containing all features with relevance greater than the mean of all relevance values ($\overline{r}$). In the testing phase, given a test instance t and a threshold th, LFSUF looks at every feature F in t, comparing the value of the feature with the threshold th. When the feature value is greater than the threshold, LFSUF sets this feature as selected. At the end of the process, LFSUF removes every feature not marked as selected, and the remaining features are used in the lazy classification of the current test instance t. The LFSUF algorithm is executed for each test instance. However, note that the relevance array is computed only once in the preliminary step, which is the most computationally expensive part of the algorithm, and then it is stored in memory to be used whenever a new instance needs to be classified.

Algorithm 1 describes LFSUF in detail. This algorithm outputs a subset of features named $SelectedFeatSubSet$. In the preliminary step (lines 1 to 6), the array $Relevance$ receives the relevance value of every feature X_i in X (lines 2 to 4). Then, in line 5, LFSUF calculates the mean of the relevance values. After that, every feature whose relevance value is greater than (or equal to) the mean relevance $\overline{r}$ is assigned to the subset F.

Algorithm 1 Lazy Feature Selection for Uncertain Features (LFSUF)

Input : t (test instance) and a threshold th
Output: a subset of features $SelectedFeatSubSet$

```
 1: # Begin of the preliminary step
 2: for each feature $X_i$ in $X$ do
 3:     $Relevance[X_i] \leftarrow FStat(X_i)$
 4: end for
 5: $\overline{r} \leftarrow mean(Relevance)$
 6: $F \leftarrow \{$all $X_i$ whose $Relevance[X_i] \geq \overline{r}\}$

 7: # Begin of the testing step
 8: $F_{max} \leftarrow F_1$
 9: for each feature $F_i$ in $t$ do
10:     $Status[F_i] \leftarrow$ "Removed"
11: end for
12: for each feature $F_i$ in $t$ do
13:     if $Value(F_i, t) > th$ then
14:         $Status[F_i] \leftarrow$ "Selected"
15:     end if
16:     if $Value(F_i, t) > Value(F_{max}, t)$ then
17:         $F_{max} \leftarrow F_i$
18:     end if
19: end for
20: if $Value(F_{max}, t) \leq th$ then
21:     $Status[F_{max}] \leftarrow$ "Selected"
22: end if
23: $SelectedFeatSubSet \leftarrow$ features with $Status$ set to
    "Selected"
24: return $SelectedFeatSubSet$
```

The main phase of LFSUF works as follows (lines 7 to 24). First, LFSUF assings the first feature in F to the variable F_{max} (line 8). Next, the $Status$ array is initialised with the "Removed" value for all features. In line 12, for each feature F_i in F, the function $Value(F_i, t)$ returns the value of F_i in the test instance t. The returned value is then compared to th (line 13). If the returned value is greater than th then this feature will be used in the classification task and is marked with the "Selected" tag (line 14). In line 16, we verify if the value of F_i in t is the maximum value found so far. If this is true then we update the pointer F_{max}. In lines 20 to 22, we verify if the highest feature value for a given instance is less than the threshold th. If this is true then no feature was selected, and in this special case we mark the feature with the highest value as "Selected". Finally, the feature subset $SelectedFeatSubSet$ receives all features whose $Status$ is "Selected" and this subset is returned by the algorithm (lines 23 and 24). Then, a lazy classifier is executed for the test instance t using only the selected features. Note that, if no feature has a value greater than the threshold th, the algorithm looks for the highest feature value in the instance and set that feature to "Selected", so there is always at least one feature being used in the classification task.

The LFSUF method presents some desirable characteristics: (i) it selects only feature values with high chance of being true (assuming that the threshold value is relatively high), which are clearly more informative than features with low confidence; (ii) since the number of features values with low confidence is large, it tends to select fewer features than the other methods used in our experiments (as shown later).

4 Results

In this Section, we present and analyse the experimental results in terms of predictive accuracy, testing: (i) what is the effect of combining GO and PPI features in the predictive accuracy of two classifiers for predicting longevity-related genes, and (ii) how effective is our feature selection method (LFSUF) in dealing with uncertain features.

4.1 Experimental Methodology

To select the best classification algorithm for this problem, we compared three traditional classifiers (1-NN with Euclidean distance, Naïve Bayes, and Random Forest) and two classifiers tailored for uncertain data: 1-NN using a distance measure capable of dealing with uncertain values called Probabilistic Jaccard [15] and a Decision Tree tailored for uncertain data (DTU) [18]. This comparison is provided as a supplementary material. Then, after this initial evaluation, for all experiments in this work, we employed the traditional Naïve Bayes (NB) and the 1-NN using the Jaccard distance (1-NN hereafter), since they achieved the best predictive results. It is worth noting that both NB and 1-NN with Euclidian distance were previously used to the prediction of longevity genes [24], [25], [26], [27].

The predictive accuracy was estimated by 10-fold cross-validation. Since most datasets have imbalanced class distributions (see Table 1), we evaluated the methods' predictive accuracy by using the Geometric Mean (GM) of sensitivity and specificity, which is defined as: $GM = \sqrt{Sensitivity * Specificity}$. A classifier that assigns the instances to each class with probability 0.5 would have a GM of about 50%.

To determine whether the differences in GM are statistically significant, we ran the non-parametric, rank-based

Friedman test and the Holm post-hoc test ([7]), as recommended by [2]. First, the Friedman test was run with the null hypothesis that the average ranks (based on GM values) of all methods are the same. The alternative hypothesis is that there is a difference between the average ranks of all methods as a whole, without identifying which pairs of methods have significantly different results. If the null hypothesis is rejected, we run the Holm post-hoc test (which corrects for multiple hypothesis testing) to compare the results of the best method overall against each of the other methods. Both the Friedman and Holm test were used at the 0.05 significance level.

4.2 The Effect of Combining GO and PPI Features in the Predictive Accuracy of Two Classifiers

The effect of combining GO and PPI features for predicting ageing-related classes is still unclear in the literature. To answer this question, we evaluated the 28 datasets containing GO and PPI features, described earlier. We created two versions of each dataset. The first version contains GO features only, and the second version contains both GO and PPI features. It is worth mentioning that the Probabilistic Jaccard distance used in the 1-NN classifier behaves like a traditional Jaccard distance when features are not uncertain (such as the GO feature set).

Table 2 presents the results of the computational experiment. The numerical columns show the GM results for Naïve Bayes and 1-NN, when applied to GO and GO.PPI feature sets. Each row presents the GM results for a given dataset. The last but one row presents the average rank (Avg. Rank) for each feature type (GO and GO.PPI), for each classifier (Naïve Bayes and 1-NN). This was calculated by first assigning the rank 1 (or 2) to the best (or worst) feature set type for each of the 28 datasets, and then averaging each feature set type's rank across the 28 datasets. The last row shows the number of wins (i.e., the number of times that a feature set type had the highest GM), for each classifier. In the row right below the table, the symbol $\succ$ denotes a statistically significant difference between methods, e.g., $\{a\} \succ \{b, c\}$ means that a is significantly better than b and c.

The results show that, for 1-NN, the feature set using GO and PPI has the best performance overall. It has the best average rank (1.21) and the highest number of wins (22 out of 28 datasets). This result is statisitically significantly better than the one using GO features only, according to the Holm test (p-value = 0.002). On the other hand, for Naïve Bayes, using only GO features leads to the best Avg. Rank, with the highest GM in 18 out of 28 datasets, but there is no significant difference between the results for using only GO features vs. GO and PPI features (p-value = 0.138).

The two best overall results in Table 2 are NB with GO features and 1-NN with GO and PPI features. When directly compared, the 1-NN combining both types of features outperformed the NB with only GO features in 24 out of 28 datasets.

In summary, combining GO and PPI features improve predictive accuracy by comparison with using only GO features in most cases when using the 1-NN classifier, but the opposite effect was observed with the Naïve Bayes (NB) classifier – i.e., it performs better when trained using only the subset of GO features. NB is known to have a poor predictive accuracy when applied to highly correlated features, which is more likely to happen on high dimensional feature spaces [12]. So, this weak result for NB can be explained by the very large number of PPI features, which is about 10 times the number of GO features.

TABLE 2
Geometric mean of sensitivity and specificity (%) obtained by Naïve Bayes and 1-NN on GO and GO.PPI feature sets with no feature selection.

		Naïve Bayes		1-NN	
	Datasets	GO	GO.PPI	GO	GO.PPI
C. elegans	BP	61.95	**69.30**	56.43	**64.56**
	CC	65.71	**66.84**	60.43	**63.50**
	MF	57.56	**69.43**	54.00	**66.96**
	BP.CC	61.87	**70.67**	61.42	**66.58**
	BP.MF	61.89	**71.58**	58.48	**65.44**
	CC.MF	64.22	**68.58**	60.44	**62.86**
	BP.CC.MF	62.38	**69.44**	58.60	**66.72**
D. melanogaster	BP	59.37	**59.41**	**66.12**	62.70
	CC	**66.69**	58.77	**72.39**	69.16
	MF	**57.98**	57.80	55.68	**59.42**
	BP.CC	**57.65**	55.98	63.56	**67.91**
	BP.MF	**57.25**	55.98	65.40	**66.48**
	CC.MF	**65.78**	59.11	59.71	**66.15**
	BP.CC.MF	**59.36**	55.98	64.47	**65.63**
M. musculus	BP	59.06	**59.53**	**68.35**	64.78
	CC	**64.07**	58.97	53.40	**65.66**
	MF	**63.45**	59.45	63.41	**70.39**
	BP.CC	**67.41**	57.05	**67.36**	62.12
	BP.MF	**64.88**	57.05	**69.54**	66.94
	CC.MF	**61.62**	57.46	61.15	**74.14**
	BP.CC.MF	**70.24**	57.46	**69.00**	63.21
S. cerevisiae	BP	**61.51**	52.38	57.89	**62.65**
	CC	**57.60**	50.32	53.94	**63.24**
	MF	34.23	**50.32**	45.58	**60.28**
	BP.CC	**63.08**	52.48	62.66	**67.05**
	BP.MF	**62.13**	52.38	55.04	**62.54**
	CC.MF	**59.87**	50.32	47.30	**61.57**
	BP.CC.MF	**62.82**	52.48	57.45	**64.15**
	Avg Rank	**1.36**	1.64	1.79	**1.21**
	#Wins	**18**	10	6	**22**

1-NN: $\{\text{GO.PPI}\} \succ \{\text{ GO}\}$

In the next section, we add a pre-processing step to our predictive workflow by using a feature selection method for uncertain features with the objective of improving the predictive accuracy of NB and 1-NN.

4.3 Comparison of Feature Selection Methods for Uncertain Features Using GO and PPI Features

To assess the effect of our proposed feature selection method on uncertain features, we run an experiment comparing our LFSUF method against two traditional feature selection methods from different paradigms and a baseline that does not perform any feature selection, all implemented within the Weka tool [6]. The first traditional feature selection method is a wrapper method with best first search (BF) available in the tool. BF was executed with default parameters and the GM measure as the optimisation function. The second method is a filter approach using the F-statistics. It computes the FStat for each feature and selects the $th_{fstat}\%$ features with the highest values. Since FStat is very sensitive to the th_{fstat} parameter, we select it by running an internal 3-fold cross-validation on the training set with th_{fstat} being selected from $1, 5, 10, 25$ and 50.

The LFSUF method uses a threshold (th) that regulates the level of confidence that features need to have in order to be used by the classification algorithm. Similarly to the FStat, we calibrate the parameter th of LFSUF by running an internal 3-fold cross-validation, with th being selected from $.150, .400, .700$ and $.900$. Those threshold values are the confidence levels suggested in the STRING database [22].

Tables 3 and 4 show the result of our experiment for Naïve Bayes and 1-NN respectively using the GO.PPI dataset. These tables show first the GM results when applying no feature selection (column 'No FS', with the same values as column GO.PPI in Table 2), and then the results for the BF, FStat and LFSUF methods.

These tables show that LFSUF achieved the best predictive accuracy-based average rank (across datasets) for both NB and 1-NN. For NB, LFSUF achieved the highest number of wins in 21 out of 28 datasets, and also the best average rank which was significantly better than the average rank of FStat, No FS and BF (Holm p-values of 0.022, 0.001 and 0.001, respectively). For 1-NN, LFSUF also obtained the best average rank and the highest number of wins, outperforming the other methods in 24 out of 28 datasets, with statistically significantly better average ranks than No FS, FStat and BF (Holm p-values of 0.034, 0.001 and 0.001, respectively). It is also worth saying that for both NB and 1-NN, LFSUF is always the best method for, respectively, *S. cerevisiae* and *C.elegans* datasets.

Note also that the best overall results in Tables 3 and 4 were obtained by LFSUF for NB and 1-NN, respectively. When these two results are compared, LFSUF with NB outperforms LFSUF with 1-NN in 18 out of 28 datasets. This result confirms that using PPI features along with GO features is helpful, since NB with LFSUF using GO and PPI features outperformed NB without feature selection using GO features only.

These results are particularly interesting since, for both classification algorithms, the predictive accuracy increases with the use of our feature selection method for uncertain features, showing that combining GO and PPI features and using our method clearly increases predictive accuracy.

4.4 Comparison of Feature Selection Methods for Uncertain Features using PPI Features

In the experiments reported in the previous section, all datasets contain features from GO (certain features) and PPI (uncertain features). However, it is also interesting to measure the predictive accuracy of the feature selection methods using only uncertain PPI features. For this task, we use four datasets (one for each model organism) with PPI featurs only, i.e., without any GO features. Like in the last section, we compare our feature selection method LFSUF against the feature selection methods BF and FStat, as well as against the baseline No FS.

TABLE 3
Geometric mean of sensitivity and specificity (%) obtained by NB with LFSUF and traditional feature selection methods using the GO.PPI datasets

	Datasets	No FS	BF	FStat	LFSUF
C.elegans	BP	69.30	59.15	**68.99**	69.20
	CC	66.84	64.51	65.52	**71.09**
	MF	69.43	62.37	67.45	**70.40**
	BP.CC	70.67	62.50	68.27	**70.21**
	BP.MF	**71.58**	62.51	67.81	70.04
	CC.MF	68.58	62.70	63.02	**72.17**
	BP.CC.MF	69.44	61.10	65.40	**70.68**
D.melanogaster	BP	59.41	52.07	**69.93**	59.81
	CC	58.77	57.41	68.44	**69.90**
	MF	57.80	53.96	60.49	**62.66**
	BP.CC	55.98	59.16	60.03	**64.77**
	BP.MF	55.98	50.99	61.52	**64.35**
	CC.MF	59.11	47.86	**70.28**	68.90
	BP.CC.MF	55.98	63.15	**64.91**	65.57
M.musculus	BP	59.53	53.35	69.48	**71.18**
	CC	58.97	60.18	63.78	**69.07**
	MF	59.45	57.63	66.01	**70.27**
	BP.CC	57.05	53.35	66.82	**72.57**
	BP.MF	57.05	52.53	**74.55**	73.80
	CC.MF	57.46	52.02	**73.03**	71.13
	BP.CC.MF	57.46	54.77	71.58	**72.00**
S.cerevisiae	BP	52.38	59.97	66.70	**74.57**
	CC	50.32	61.26	60.37	**73.52**
	MF	50.32	57.37	62.04	**71.31**
	BP.CC	52.48	60.11	68.59	**74.22**
	BP.MF	52.38	57.22	66.83	**73.53**
	CC.MF	50.32	56.83	56.09	**73.01**
	BP.CC.MF	52.48	54.35	67.20	**73.88**
	Avg Rank	3.00	3.57	2.18	**1.25**
	#Wins	3	0	4	**21**

$\{\text{LFSUF}\} \succ \{\text{FStat, No FS and BF}\}$

Table 5 shows the results for NB. LFSUF achieved a perfect average rank of 1 (winning in all 4 datasets), being statistically significantly better than No FS, BF and FStat methods (Holm p-values of 0.003, 0.019 and 0.007, respectively).

Table 6 shows the results for 1-NN. Again, LFSUF obtained the best average rank and the highest number of wins, being statistically significantly better than No FS, BF, and FStat methods (Holm p-values of 0.048, 0.003 and 0.019, respectively). The results show that LFSUF performed better than all other methods for all but one model organism. The exception was the *D. melanogaster* dataset, where 1-NN had higher predictive accuracy when no feature selection was used.

4.5 Evaluating the Feature Space Compression

Apart from the predictive accuracy of a classifier, another important result to be evaluated is the number of features selected for classifying each instance. LFSUF benefited from

TABLE 4
Geometric mean of sensitivity and specificity (%) obtained by 1-NN with LFSUF and traditional feature selection methods using the GO.PPI dataset.

	Datasets	No FS	BF	FStat	LFSUF
C.elegans	BP	64.56	65.45	62.03	**67.12**
	CC	63.50	62.27	63.80	**68.31**
	MF	66.96	62.53	58.86	**69.13**
	BP.CC	66.58	67.66	60.61	**67.93**
	BP.MF	65.44	64.76	61.36	**68.62**
	CC.MF	62.86	63.60	60.19	**68.94**
	BP.CC.MF	66.72	64.15	60.06	**68.59**
D.melanogaster	BP	62.70	56.75	60.50	**63.69**
	CC	69.16	67.47	65.31	**76.29**
	MF	59.42	53.75	55.27	**64.23**
	BP.CC	**67.91**	67.68	67.35	62.11
	BP.MF	66.48	50.89	65.50	**68.74**
	CC.MF	66.15	64.64	65.46	**66.42**
	BP.CC.MF	65.63	65.78	68.83	**70.95**
M.musculus	BP	64.78	56.87	71.84	**72.80**
	CC	65.66	58.48	65.87	**68.06**
	MF	70.39	58.19	68.90	**75.04**
	BP.CC	62.12	55.89	66.11	**70.54**
	BP.MF	66.94	56.87	**78.56**	74.24
	CC.MF	74.14	62.13	71.33	**77.10**
	BP.CC.MF	63.21	60.00	**73.11**	72.84
S.cerevisiae	BP	62.65	53.86	25.42	**73.67**
	CC	**63.24**	54.24	42.44	62.27
	MF	60.28	55.82	34.03	**67.72**
	BP.CC	67.05	57.81	33.54	**70.15**
	BP.MF	62.54	54.92	29.72	**69.94**
	CC.MF	61.57	55.84	40.14	**62.89**
	BP.CC.MF	64.15	58.51	39.99	**71.23**
	Avg Rank	2.32	3.29	3.18	**1.21**
	#Wins	2	0	2	**24**

{LFSUF} ≻ {FStat, No FS and BF}

TABLE 5
GM of sensitivity and specificity (%) obtained by NB with LFSUF and traditional feature selection methods, using only uncertain (PPI) features.

Dataset	No FS	BF	FStat	LFSUF
C.elegans	69.21	60.59	59.81	**70.32**
D.melanogaster	55.97	56.20	59.22	**63.61**
M.musculus	55.11	58.48	67.01	**72.07**
S.cerevisiae	50.22	62.48	30.09	**73.15**
Rank	3.25	2.75	3.00	**1.00**
#Wins	0	0	0	4

{LFSUF} ≻ {BF, FStat, No FS}

TABLE 6
GM of sensitivity and specificity (%) obtained by 1-NN with LFSUF and traditional feature selection methods, using only uncertain (PPI) features.

Dataset	No FS	BF	FStat	LFSUF
C.elegans	65.18	54.92	60.00	**68.74**
D.melanogaster	**64.04**	49.95	53.31	57.95
M.musculus	65.25	62.99	68.47	**72.03**
S.cerevisiae	60.65	66.75	42.59	**67.98**
Rank	2.25	3.50	3.00	**1.25**
#Wins	1	0	0	3

{LFSUF} ≻ {No FS, FStat, BF}

its flexibility as lazy feature selection method and selected a very small number of features customized for each test instance. By contrast, BF and FStat select substantially larger subsets of features (which are used for classifying all instances). On average across all datasets, LFSUF selected only 0.96% (for NB) and 2.68% (for 1-NN) of all PPI features. BF selected 5.01% (for NB) and 3.29% (for 1-NN) of all PPI features. The worst result was obtained by FStat, which selected 19.00% (for NB) and 18.00% (for 1-NN) of all PPI features. Recall that GO features do not undergo selection, i.e., all GO features are used for classifying every test instance.

Hence, LFSUF achieved overall the best predictive accuracy with the lowest number of features for both the 1-NN and the NB classifiers. This seems due to the removal of a large number of features with low predictive power. In the context of the LFSUF method, features with low predictive power are those whose feature value scores are low, representing low-confidence protein interactions.

4.6 Analysis of the Most Frequent Selected Features

We have ranked all PPI features for each model organism in decreasing order of selection frequency by the LFSUF method. For this ranking we used the datasets containing only PPI features (i.e., no GO features) and the results of NB classifier, since it achieved the best results overall. Due to space constraints, the full ranking is available online[1].The top-7 features for each of the 4 datasets (one per organism) of PPI features are shown in Table 7. In this table, the first column shows the model organism. This column is followed by the protein name, the number of instances (#Sel.) and the percentage of instances for which the feature was selected (%Sel).

Some of the most selected features have known relation with ageing, as can be verified in the HAGR database, which contains annotated pro-/anti-longevity genes and was used to build the datasets used in this work. In other words, these features are also instances in our datasets. Based on that, we verify that some of the top selected features represent interaction with known pro-longevity proteins – for example: protein F52C6.2 from *C. elegans*, which is ranked as the 6th most selected feature for that organism, and Pten, which is ranked 4th for the $M.musculus$ organism. Also, for *S. cerevisiae*, the protein PET127 is a known anti-longevity

[1] http://github.com/pablonsilva/FSforUncertainFeatureSpaces

TABLE 7
Top-7 PPI features selected by LFSUF for each model organism (dataset), sorted by selection frequency.

	Protein	#Sel	%Sel.
C.elegans	rab-14	219	32.82
	hsd-3	133	19.94
	Y71H2B.5	123	18.44
	pod-2	120	17.99
	ctl-2	114	17.09
	F52C6.2	112	16.79
	F11D5.7	111	16.64
M.musculus	Ripk4	25	22.94
	Lhx9	18	16.51
	Polr2k	16	14.68
	Pten	13	11.93
	Rad51c	13	11.93
	Rarg	12	11.01
	Tkt	11	10.09
D.melanogaster	eIF4E-3	33	24.81
	ari-1	28	21.05
	eIF4E-4	27	20.30
	PGRP-SB1	21	15.78
	not	21	15.78
	CG4452	20	15.04
	AnxB11	18	13.53
S.cerevisiae	RPS7B	129	38.97
	NOC4	37	11.18
	PUS2	36	10.88
	SPO20	35	10.57
	UTP6	35	10.57
	UTP25	35	10.57
	PET127	34	10.27

protein, and its ranked as the 7th most selected feature for that organism.

5 Conclusions

In this paper, we tackled the problem of feature selection in datasets containing Protein-Protein Interaction (PPI) features with uncertain values, i.e., feature values represented by a confidence score – where the higher the score, the higher the chance of the current instance (protein) actually interacting with the protein associated with the PPI feature. In this context, we proposed the Lazy Feature Selection for Uncertain Features (LFSUF) method, based on the hypothesis that, for a given instance, a feature with high confidence score has better class-discrimination power, since it has a strong evidence of being present in the current instance.

The proposed LFSUF method obtained overall the best predictive accuracy in the classification of pro-longevity vs. anti-longevity genes from four model organisms, when using two different classifiers (Naive Bayes and 1-NN) and two different types of feature sets – first, using both Gene Ontology (GO) and uncertain PPI features; and second, using only uncertain PPI features. Also, note that LFSUF achieved better predictive accuracy using smaller selected feature sets on average, when compared against other feature selection methods. This is desirable, since it improves the interpretability potential of the predictions made by the model. In summary, our results indicate that the application of lazy feature selection on datasets with uncertain features is an effective approach, leading to higher predictive accuracy and better interpretability potential.

Future work could include the proposal of novel feature selection strategies for other types of uncertain features. And also exploiting other feature selection paradigms, such as the wrapper approach.

Acknowledgments

This work was supported by grant 88882.183892/2018-01 from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Brazil) (to PNS) and grant 308369/2015-7 from Conselho Nacional de Desenvolvimento Científico e Tecnológico (Brazil) (to AP).

References

[1] D. W. Aha, "Lazy Learning", Kluwer Academic Publishers, 1997.

[2] J. Demsar, "Statistical comparisons of classifiers over multiple datasets", *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[3] G. Doquire and M. Verleysen, "Feature Selection with Mutual Information for Uncertain Data". In *Proc. of DaWaK*, LNCS 6862, pp. 330–341, 2011.

[4] F. Fabris, J. P. Magalhães, A. A. Freitas, "A review of supervised machine learning applied to ageing research", *Biogerontology*, vol. 18, no. 2, pp. 171–188, 2017.

[5] M. Fellenberg, K. Albermann, A. Zollner, H. W. Mewes, J. Hani, "Integrative analysis of protein interaction data". Proc. Int. Conf. Intell. Syst. Mol. Biol., pp. 152–161, 2000.

[6] M. Hall, "The weka data mining software: an update", *ACM-SIGKDD Exploration Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[7] S. Holm, "A simple sequential rejective method procedure", *Scandinavian Journal of Statistics*, vol. 6, pp. 65–70, 1979.

[8] R. Kaletsky and C. T. Murphy, "The role of insulin igf-like signaling in C. elegans longevity and aging". *Disease Models and Mechanisms*, vol. 3, no. 7, pp. 415-419, 2010.

[9] C. J. Kenyon, "The genetics of ageing", *Nature*, vol. 464, no. 7288, pp. 504-523, 2010.

[10] T. B. L. Kirkwood and S. N. Austad, "Why do we age?", *Nature*, vol. 408, no. 6809, pp. 233-238, 2000.

[11] X. Kong, P. S. Yu, X. Wang and A. B. Ragin, "Discriminative Feature Selection for Uncertain Graph Classification". In *Proc. of the 2013 SIAM International Conference on Data Mining*, pp. 82–93. SIAM, 2013.

[12] H. Liu and H. Motoda, "Computational Methods of Feature Selection", *CRC Press*, 2008.

[13] H. Liu and H. Motoda, "Feature Selection for Knowledge Discovery and Data Mining". *Springer*, 2012.

[14] J. Magalhães et al., "The human ageing genomic resources: Online databases and tools for biogerontologistis", *Ageing Cell*, vol. 8, no. 1, pp. 65–72, 2009.

[15] I. Martire, P. N. da Silva, A. Plastino, F. Fabris, A. A. Freitas, "A novel probabilistic Jaccard distance measure for classification of sparse and uncertain data". In *Proc. of Symposium on Knowledge Discovery, Mining and Learning, KDMILE 2017*, pp. 81–88, 2017.

[16] R. Pereira et al., "Lazy attribute selection: Choosing attributes at classification time". *Intelligent Data Analysis*, vol. 15, no. 5, pp. 715–732, 2011.

[17] D. E. L. Promislow, "Protein networks, pleiotropy and the evolution of senescence". Proceedings. Biological Sciences, vol. 271, no. 1545, pp. 1225-1234, 2004.

[18] B. Qin, Y. Xia, and F. Li, "DTU: A Decision Tree for Uncertain Data". In *Proc. Advances in Knowledge Discovery and Data Mining (PAKDD)*, pp. 4–15, 2009.

[19] J. M. V Raamsdonk, "Mechanisms underlying longevity: A genetic switch model of aging", *Experimental Gerontology*, vol. 107, pp. 136-139, 2018.

[20] J.D.L. Rivas and C. Fontanillo, "Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks". *PLoS Comput Biol.*, vol. 6, no. 6, pp. 1-8, 2010.

[21] P. N. da Silva, A. Plastino and A. A. Freitas, "A novel genetic algorithm for feature selectionin hierarchical feature spaces" In *Proc. of the 2018 SIAM International Conference on Data Mining*, pp 738–746, SIAM, 2018.

[22] D. Szklarczyk et al., "STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". *Nucleic Acids Res.*, vol. 47, pp. 607-613, 2019.

[23] The GO Consortium, "Gene ontology: Tool for the unification of biology", *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[24] C. Wan and A. A. Freitas, "Prediction of the pro-longevity or anti-longevity effect of caenorhabditis elegant genes based on bayesian classification methods", In *Proc. 2013 International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 373–380. IEEE Press, 2013.

[25] C. Wan and A. A. Freitas, "Two methods for constructing a geneontology-based feature network for a bayesian network classifier andapplications to datasets of ageing-related genes", In *Proc. ACM Conf. on Bioinfo. Comp. Biology and Health Informatics (BCB)*, pp. 27–36, 2015.

[26] C. Wan and A. A. Freitas, "An empirical evaluation of hierarchical feature selection methods for classification in bioinformatics datasets with gene ontology-based features", *Artif. Intel. Review*, vol. 50, no. 2, pp. 201–240, 2018.

[27] C. Wan. et al., "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 Bioinformatic*s, vol. 12, no. 2, pp. 262–275, 2015.

[28] C. Wan, "Hierarchical Feature Selection for Knowledge Discovery: application of data mining to the biology of ageing". 117 pages. Springer, 2019.

[29] L. Wang et al., "Feature selection methods for big data bioinformatics: A survey from the search perspective. Methods", vol. 111, no. 1, pp. 21–31, 2016.

[30] D. Wieser et al., "Computational biology for ageing", *Philosophical Trans. of the Royal Society B: Biological Sciences*, vol. 366, no. 1561, pp. 51-63, 2011.

Pablo Nascimento da Silva is a D.Sc. candidate at Fluminense Federal University in Brazil, where he also obtained his M.Sc. (2014) in Computer Science. Previously, he obtained a B.Sc. (2011) in Computer Engineering at the Catholic University of Petropolis, Brazil. His main research interests are machine learning and data science. He is a student member of the IEEE.

Prof. Alexandre Plastino is a Full Professor at Fluminense Federal University in Brazil. He obtained his B.Sc. (1988) and M.Sc. (1990) in Computer Science at Federal University of Rio de Janeiro (UFRJ) and his D.Sc. (2000) in Computer Science at Pontifical Catholic University of Rio de Janeiro (PUC-Rio). In 2008, he conducted a postdoctoral research at University of Kent in the United Kingdom. His current research interests concentrate on Data Science and Combinatorial Optimization.

Dr. Fabio Fabris worked as a postdoctoral research associate applying data mining to ageing research, with a focus on model interpretability. He completed his doctoral thesis on graphical models applied to ageing-related classification tasks under the supervision of Alex A. Freitas.

Prof. Alex A. Freitas is a Professor of Computational Intelligence at the University of Kent, UK. He has an interdisciplinary academic background, with a PhD in Computer Science (University of Essex, UK, 1997), in the area of machine learning (or data mining); and a research-oriented master's degree (an MPhil) in Biological Sciences (University of Liverpool, UK, 2011), in the area of the biology of ageing. His main research interests are machine learning, data mining and knowledge discovery, the biology of ageing, applications of machine learning to the life sciences, evolutionary algorithms and swarm intelligence.

Supplementary Material: A Novel Feature Selection Method for Uncertain Features: An Application to the Prediction of Pro-/Anti- Longevity Genes

Pablo Nascimento da Silva[1] , **Alexandre Plastino**[1] , **Fabio Fabris**[2] , **Alex A. Freitas**[2]

[1]Institute of Computing, Fluminense Federal University (UFF), Niterói-RJ, Brazil

[2]School of Computing, University of Kent, UK

This document contains additional material to the manuscript mentioned above submitted to the IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Table 1. Geometric mean of sensitivity and specificity (%) obtained by Naïve Bayes, 1-NN with Euclidian distance, Random Forest (RF), Decision Tree for Uncertain Data (DTU) and 1-NN with Probabilistic Jaccard (1-NN Jacc.) distance.

	Datasets	NB	1-NN Euc.	RF	UDT	1-NN Jacc.
C.elegans	BP	**69.30**	58.40	64.08	66.99	64.56
	CC	**66.84**	59.85	61.98	59.37	63.50
	MF	**69.43**	53.42	64.55	63.51	66.96
	BP.CC	**70.67**	59.94	61.96	65.36	66.58
	BP.MF	**71.58**	58.01	64.60	67.05	65.44
	CC.MF	**68.58**	58.59	63.21	60.98	62.86
	BP.CC.MF	**69.44**	59.96	61.98	65.44	66.72
D.melanogaster	BP	59.41	58.77	37.45	60.80	**62.70**
	CC	58.77	**71.85**	36.11	66.63	69.16
	MF	57.80	51.30	34.96	**60.88**	59.42
	BP.CC	55.98	56.09	35.96	51.08	**67.91**
	BP.MF	55.98	54.51	39.19	56.20	**66.48**
	CC.MF	59.11	60.95	39.57	**71.07**	66.15
	BP.CC.MF	55.98	56.79	34.82	62.73	**65.63**
M.musculus	BP	59.53	**65.09**	55.83	63.40	64.78
	CC	58.97	55.01	44.70	49.12	**65.66**
	MF	59.45	61.31	50.14	67.63	**70.39**
	BP.CC	57.05	**64.31**	50.94	45.33	62.12
	BP.MF	57.05	62.77	57.24	49.16	**66.94**
	CC.MF	57.46	48.88	50.97	60.19	**74.14**
	BP.CC.MF	57.46	**65.51**	44.94	49.45	63.21
S.cerevisiae	BP	52.38	54.68	27.34	50.49	**62.65**
	CC	50.32	52.49	31.57	43.78	**63.24**
	MF	50.32	43.51	22.32	49.09	**60.28**
	BP.CC	52.48	54.47	27.29	52.41	**67.05**
	BP.MF	52.38	49.83	22.32	51.93	**62.54**
	CC.MF	50.32	48.19	22.36	44.84	**61.57**
	BP.CC.MF	52.48	45.57	15.76	54.99	**64.15**
	Rank Avg.	2.57	3.29	4.46	3.11	**1.57**
	#Wins	7.00	4.00	0.00	2.00	**15.00**