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 [12], ultimately leading to death. Even for related species, the speed at which such functional deterioration happens differs to some extent [11]. 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 [33].

Experiments in model organisms have identified several hundred genes that influence the ageing process (speeding it up or slowing it down) [22]. 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 [30], [31]. Conversely, anti-longevity genes are those genes whose decreased expression extends lifespan and/or those whose over-expression decreases it.

Gene Ontology (GO) terms [26] have been widely used as predictive features for building models for the classification of pro-/anti-longevity genes [1], [3], [6], [8], [24], [28], [29], [30]. 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 [23], a widely used characteristic of proteins that could potentially help finding those proteins linked with ageing [1], [7], [20]. 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 [15], [16], *feature selection* methods can be used to cope with this problem. This type of method aims at improving the predictive accuracy of the classifier

- *P.N. da Silva and A. Plastino are with the Institute of Computing, Universidade Federal Fluminense, Niterói, RJ, Brazil. E-mail: {pablosilva@id.uff.br, 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}*

by selecting a subset of relevant features. It is a challenging problem since the number of candidate feature subsets grows exponentially with the number of features, which is usually a problem when dealing with bioinformatics datasets. More precisely, the number of candidate feature subsets is $2^d - 1$, where d is the number of features. There are many different techniques for dealing with this problem. The suitable technique or approach to be used is not easily identifiable and should be selected according to the problem at hand. For a comprehensive overview of feature selection techniques in bioinformatics domains the reader might refer to Wang et al. [32].

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 $X = \{X_1; X_2; \dots; X_d\}$ be the set of predictive features, where $d \geq 1$, and $C = \{C_1; C_2; \dots; C_q\}$ be the finite set of possible classes, where $q \geq 2$. Given a training set $D = \{(x_1; y_1); (x_2; y_2); \dots; (x_n; y_n)\}$, where each instance i is associated with a class value $y_i \in C$ and a feature vector $x_i = \{x_{i1}; x_{i2}; \dots; x_{id}\}$, 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 $x_i = \{x_{i1}; x_{i2}; \dots; x_{id}\}$, 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 [15], [16]. 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 [2], [19], 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 [14], a feature selection method for graph classification is introduced. It deals with graphs where the linkages 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 [5]. 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 [5].

2.4 Data Preparation

2.4.1 Gene Ontology (GO)

The Gene Ontology (GO) database [26] 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 [23]. There are many different databases describing interactions between proteins. In this work, we use the STRING database [25], 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 [24], [28], [30], 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) [17], the Gene Ontology (GO) database (version: 2015-10-10) [26] and Protein-Protein Interactions from the STRING database [25]. 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 predictive 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	12437	990	11447	34.4
	CC	484	11162	177	10985	36.4
	MF	504	11150	262	10888	37.7
	BP.CC	664	12624	1167	11457	34.3
	BP.MF	663	12731	1252	11479	34.2
	CC.MF	566	11729	439	11290	36.2
	BP.CC.MF	667	12909	1429	11480	34.3
D. melanogaster	BP	132	7358	799	6559	72.0
	CC	122	6548	88	6460	70.5
	MF	126	6697	144	6553	70.6
	BP.CC	133	7501	887	6614	71.4
	BP.MF	133	7557	943	6614	71.4
	CC.MF	130	6815	232	6583	70.7
	BP.CC.MF	133	7645	1031	6614	71.4
M. musculus	BP	109	11512	1331	10181	68.8
	CC	107	10235	141	10094	68.2
	MF	106	10322	239	10083	67.9
	BP.CC	109	11653	1472	10181	68.8
	BP.MF	109	11751	1570	10181	68.8
	CC.MF	109	10561	380	10181	68.8
	BP.CC.MF	109	11892	1711	10181	68.8
S. cerevisiae	BP	331	6304	843	5461	13.3
	CC	331	5605	144	5461	13.3
	MF	331	5681	220	5461	13.3
	BP.CC	331	6448	987	5461	13.3
	BP.MF	331	6524	1063	5461	13.3
	CC.MF	331	5825	364	5461	13.3
	BP.CC.MF	331	6668	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) [32] of each feature $X_i \in X$. Then, we build a subset of features $F \subseteq X$ containing all features with relevance greater than the mean of all relevance values ($\bar{r}$). In the testing phase, given a test instance t and a threshold th, LFSUF looks at every feature F in F, comparing the value of F_i in t 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 $\bar{r}$ is assigned to the subset F.

The main phase of LFSUF works as follows (lines 7 to 24). First, LFSUF assigns 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. Next, for each feature F_i in F, in line 13 the function $Value(F_i; t)$ returns the value of F_i in the test instance t, and the returned value is compared with th. 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 sets to "Selected" the feature with the highest value, 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).

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]  FStat(X_i)
4: end for
5: r̄  mean(Relevance)
6: F  {all X_i whose Relevance[X_i]  r̄}
7: # Begin of the testing step
8: F_max  F_1
9: for each feature F_i in t do
10:   Status[F_i]  "Removed"
11: end for
12: for each feature F_i in F do
13:   if Value(F_i; t) > th then
14:     Status[F_i]  "Selected"
15:   end if
16:   if Value(F_i; t) > Value(F_max; t) then
17:     F_max  F_i
18:   end if
19: end for
20: if Value(F_max; t)  th then
21:   Status[F_max]  "Selected"
22: end if
23: SelectedFeatSubSet  {features with Status set to "Selected"}
24: return SelectedFeatSubSet
```

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 [18] and a Decision Tree tailored for uncertain data (DTU) [21]. 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 Probabilistic Jaccard distance (1-NN hereafter), since they achieved the best predictive results. It is worth noting that both NB and 1-NN with Euclidean distance were previously used to the prediction of longevity genes [27], [28], [29], [30].

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: $GM = \sqrt{Sensitivity \cdot Specificity}$. 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 [10], as recommended by [4]. 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).

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: {GO.PPI} $\succ$ {GO}

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 [15]. 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.

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 [9]. 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 the best value of this threshold for each dataset by running an internal 3-fold cross-validation on the training set with th_{fstat} being selected from 1; 5; 10; 25 and 50.

We also also analysed the results of this th_{fstat}-tuning procedure in order to find out which th_{fstat} value was selected most often for each organism. Note that we run an external 10-fold cross-validation to validate the algorithms, and each organism is associated with 7 datasets, resulting in 70 (10 7) selections of the th_{fstat} value (each selection performed by an internal cross-validation on the training set). After the experimental evaluation, the most selected th_{fstat} for *C.elegans* and *M.musculus* datasets was 10 (selected in 86% and 83% of all 70 cases, respectively), while 25 was the most selected th_{fstat} for *D.melanogaster* and *S.cerevisae* (selected in 92% and 78% of the cases, respectively).

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 [25]. After the experimental evaluation, the most selected th for *C.elegans* datasets was 0:400 (selected in 78% of all 70 cases), whereas 0:900 was the most selected th for *D.melanogaster* *M.musculus* and *S.cerevisae* datasets (selected, respectively, in 90%, 92% and 78% of all 70 cases).

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, and Friedman p-value of 0.001). 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, and Friedman p-value of 0.001). 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.

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**

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

Note that LFSUF was the winner on the majority of the scenarios. However, in some scenarios of our experiments, the LFSUF's performance is not the best. We believe that there may be important features that are discarded by LFSUF since their F-statistics values are lower than the average F-statistics value. This may be the reason for the reduced performance of the classification in some scenarios and may be further explored in the future by changing the mechanism of eliminating features in the preliminary step.

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

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

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 features 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 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, and Friedman p-value of 0.003).

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} $\succ$ {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} $\succ$ {No FS, FStat, BF}

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, and Friedman p-value of 0.002). 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 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 per instance. 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.

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

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 associated with the PPI feature, 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' proteins are also represented as 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 protein, and its ranked as the 7th most selected feature for that organism.

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

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 with a probabilistic Jaccard distance measure) and two different types of feature sets – first, using both (certain) 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 per instance 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 (CAPES, Brazil) (to PNS) and grant 308369/2015-7 from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil) (to AP).

References

[1] C. Kerepesi, B. Daroczy, A. Sturm, T. Vellai, A. Benczur. "Prediction and characterization of human ageing-related proteins by using machine learning". *Scientific Reports*, vol. 8 no. 4094, 13 pages, 2018.
[2] D. W. Aha, "Lazy Learning", Kluwer Academic Publishers, 1997.
[3] M. Asif, H. F. Martiniano, A. M.Vicente and F. M. Couto. "Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology". *PloS one*, vol. 13 no. 12. 2018.
[4] J. Demsar, "Statistical comparisons of classifiers over multiple datasets", *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[5] G. Doquire and M. Verleysen, "Feature Selection with Mutual Information for Uncertain Data". In *Proc. of DaWaK*, LNCS 6862, pp. 330–341, 2011.