

A Global-Model Naive Bayes Approach to the Hierarchical Prediction of Protein Functions

Carlos N. Silla Jr. and Alex A. Freitas

Computing Laboratory, University of Kent, Canterbury, Kent, UK, CT2 7NF
{cns2,A.A.Freitas}@kent.ac.uk

Abstract—In this paper we propose a new global-model approach for hierarchical classification, where a single global classification model is built by considering all the classes in the hierarchy – rather than building a number of local classification models as it is more usual in hierarchical classification. The method is an extension of the flat classification algorithm naive Bayes. We present the extension made to the original algorithm as well as its evaluation on eight protein function hierarchical classification datasets. The achieved results are positive and show that the proposed global model is better than using a local model approach.

Keywords-hierarchical classification; bayesian classification; protein function prediction;

I. INTRODUCTION

Within the different types of machine learning problems, the most common focus is to solve flat classification problems. In the classification task the algorithm is given a set of labeled objects (training set), each of them described by a set of features, and the aim is to predict the label of unknown labeled objects (test set) based on their features. In a flat classification problem, each test example e (unseen during training) will be assigned a class $c \in C$ (where C is the set of classes of the given problem), where C has a “flat” structure (there is no relationship among the classes). This approach is often single label, i.e. the classifier will only output one possible class for each test example. Apart from flat classification, there are problems that are hierarchical by its nature, involving a hierarchy of classes to be predicted. E.g. in text categorization by topic, due to the large number of possible topics, the simple use of a flat classifier seems infeasible. As the number of topics becomes larger, flat categorizers face the problem of complexity that may incur in rapid increase of time and storage [1]. For this reason, using flat classification algorithms might not be fit to the problem. The machine learning method has to be tailored to deal with the hierarchical class structure.

One of the application domains that can truly benefit from hierarchical classification is the field of bioinformatics, more precisely the task of protein function prediction. This task is particularly interesting as, although the human sequencing genome project has ended, the contribution made for knowledge is less clear, because we still do not know the functions of many proteins encoded by genes [2]. Also, one of the

most used methods to infer new protein functions, BLAST, which is based on measuring similarity between protein sequences, has some limitations. In particular, proteins with similar sequences can have very different functions [3] and BLAST does not produce a classification model that can give the user insight about the relationship between protein features and proteins functions [4].

In this work, we extend the traditional flat (ignoring class-relationships) Naive Bayes to deal with a hierarchical classification problem. This extension allows the algorithm to create a global-model that allows the prediction of any class in the hierarchical class structure instead of only classes at the leaf nodes of the class hierarchy. The classification model is said to be global because it is built by considering all classes in the hierarchy – rather than building a number of local classification models as usual. We also augment the global-model Naive Bayes by using a notion of “usefulness” by taking into account the depth of the prediction. The motivation is that deeper predictions tend to be more useful (more specific and informative) to the user than more general predictions. We evaluate our approach on eight protein datasets and compare it against a suitably baseline approach tailored for hierarchical classification problems.

The remainder of this paper is organized as follows: Section II presents background on hierarchical classification. Section III discusses the new global-model naive Bayes method for hierarchical classification proposed in this paper. Section IV presents the experimental setup and reports the computational results on the task of hierarchical protein function prediction. Conclusions and some perspectives about future work are stated in Section VI.

II. HIERARCHICAL CLASSIFICATION

The existing hierarchical classification methods can be analyzed under different aspects [5], [4], as follows:

- The type of hierarchical structure of the classes, which can be either a tree structure or a DAG (Direct Acyclic Graph) structure. In this work, the datasets are organized into a tree-structured class hierarchy;
- How deep the classification in the hierarchy is performed. I.e., if the output of the classifier is always a leaf node (which [4] refers to as Mandatory Leaf-Node Prediction and [5] refers to as Virtual Category

Tree) or if the most specific (“deepest”) class predicted by the classifier for a given example could be a node at any level of the class hierarchy (which [4] refers to as Non-Mandatory Leaf Node Prediction and [5] refers to as Category Tree). In this work, we are dealing with a non-mandatory leaf node prediction problem.

- How the hierarchical class structure is explored by the algorithm. The existing hierarchical classification approaches can be classified into Local and Global approaches. In this work we propose a new global approach.

The local–model approach consists of creating a local classifier for every parent node [6] (i.e., any non-leaf node) in the class hierarchy (assuming a multi-class classifier is available) or a local binary classifier for each class node [7] (parent or leaf node, except for the root node). In the former case the classifier’s goal is to discriminate among the child classes of the classifier’s corresponding node. In the latter case, each binary classifier predicts whether or not an example belongs to its corresponding class. In both cases, these approaches are creating classifiers with a local view of the problem. Despite the differences on creating and training the classifiers, these approaches are often used with the same “top-down” class prediction strategy in the testing phase. The top-down class prediction approach works in the testing phase as follows. For each level of the hierarchy (except the top level), the decision about which class is predicted at the current level is based on the class predicted at the previous (parent) level. The main disadvantage of the local approach with the top-down class prediction approach is that a classification mistake at a high level of the hierarchy is propagated through all the descendant nodes of the wrongly assigned class.

In the global–model approach, a single (relatively complex) classification model is built from the training set, taking into account the class hierarchy as a whole during a single run of the classification algorithm. When used during the test phase, each test example is classified by the induced model, a process that can assign classes at potentially every level of the hierarchy to the test example [4]. In this work we propose a novel global classification approach to avoid the above–mentioned drawback of the local approach.

Most of the classification research in machine learning focus on the development and improvement of flat classification methods. In addition, in the field of hierarchical classification most approaches use a local-model approach with a top-down class prediction approach. The global–model approach is still under-explored in the literature and it deserves more investigation because it builds a singular coherent classification model. Even though a single model produced by the global-model approach will tend to be more complex (larger) than each of the many classification models produced by the local–model approach, intuitively the single global model will tend to be much simpler (smaller) than the

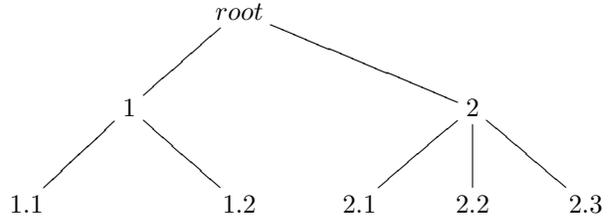


Figure 1. Hierarchical class structure example

entire hierarchy of local classification models. There is also empirical evidence for this intuitive reasoning [8], [9].

III. THE GLOBAL-MODEL NAIVE BAYES

As seen in the previous section, we are interested in developing a hierarchical classification algorithm that builds a global classification model. Moreover, we also want the algorithm to output its decision process in a human understandable format and to be able to cope with naturally missing attribute values. For these reasons, in this work, we consider the use of Bayesian algorithms instead of “black box” type of algorithms like SVM or neural networks. Bayesian methods range from the simple Naive Bayes (which assumes no dependency between the attributes given the class) to Bayesian networks, which efficiently encode the joint probability distribution for a large set of variables [10].

The training of a Bayesian classifier has two main steps: (1) Deciding the topology of the network representing attribute dependencies; (2) Computing the required probabilities. The second step, at least, needs to be adapted to hierarchical classification. Hence, in this paper we discuss how to adapt this second step to the task of hierarchical classification – considering the less investigated scenario of global classification models; whilst adapting the first step to hierarchical classification will be investigated in future research. Hence, in this paper we focus on the well-known naive Bayes algorithm, which has the advantages of simplicity and computational efficiency (an important point in hierarchical classification, given the large number of classes to be predicted). Considering Figure 1, where each node in the tree corresponds to a class, the Naive Bayes classifier has the following components:

- Topology: The topology is essentially the same for flat and hierarchical classification; the difference is that the “class node” has an internal hierarchical structure.
- Prior Probabilities - computed for each class: $P(1)$, $P(2)$, $P(1.1)$, $P(1.2)$, $P(2.1)$, $P(2.2)$, $P(2.3)$.
- Likelihoods: $P(A_i=V_{ij}|1)$, $P(A_i=V_{ij}|2)$, $P(A_i=V_{ij}|1.1)$, $P(A_i=V_{ij}|1.2)$, $P(A_i=V_{ij}|2.1)$, $P(A_i=V_{ij}|2.2)$. This is computed for each attribute A_i , and each value V_{ij} belonging to the domain of A_i , $i = 1, \dots, n$, $j = 1, \dots, i_m$, where n is the number of attributes and i_m is the number of values in the domain of the i_{th} attribute.

After training, during the testing phase, the question that arises is how to assign a class to a test example? The original (flat) Naive Bayes simply assigns the class with maximum value of the posterior probability given by $p(Class) = \prod_{i=1}^n (A_{i=V_{ij}}|Class) \times P(Class)$ [11]. However this needs to be adapted to hierarchical classification, where classes at different levels have different trade-offs of accuracy and usefulness to the user. In order to extend the original naive Bayes classifier to handle the class hierarchy, the following modifications were introduced in the algorithm:

- Modification of the prior calculations: During the training phase, when there is an example that belongs to a certain class (say class 2.1), this means that the prior probabilities of both that class and its ancestor classes (i.e. classes 2 and 2.1 in this case) are going to be updated. (This is because we are dealing with a “IS-A” class hierarchy, as usual.)
- Modification of the likelihood calculation: As in the prior calculations, when a training example is processed, its attribute-value pair counts are added to the counts of the given class and its ancestor classes. As in the previous example, if the training example belongs to class 2.1, the attribute-value counts are added to the counts of both classes 2 and 2.1.

These modifications will allow the algorithm to predict classes at any level of the hierarchy. However, although the predictions of deeper classes are often less accurate (since deeper classes have fewer examples to support the training of the classifier than shallower classes), deeper class predictions tend to be more useful to the user, since they provide more specific information than shallower class predictions. If we only consider the posterior class probability (the product of likelihood \times prior class probability) we would not take into account the usefulness to the user. It is interesting therefore to select a class label which has a high posterior probability and is also useful to the user. Therefore an optional step in the proposed method is to predict the class with maximum value of the product of posterior probability \times usefulness. The question that arises is how to evaluate the usefulness of a predicted class?

Given that predictions at deeper levels of the hierarchy are usually more informative than the classes at shallower levels, some sort of penalization for shallower class predictions is needed. In Clare’s work [12] the original formula for entropy was modified to take into account two aspects: multiple labels and prediction depth (usefulness) to the user. In this work we have modified the part of the entropy-based formula described in [12]. The main reason to modify this formula is that while Clare was using a decision tree classifier based on entropy, in this work we are using a Bayesian algorithm that makes use of probabilities. Therefore, we need to adapt the “usefulness” measure from [12] to the context our algorithm. Also, all that we need is a measure to

assign different weights to different classes at different class levels. Therefore, we adapt Clare’s measure of usefulness by using a normalized usefulness value based on the position of each class level in the hierarchy. Moreover, we only use the normalized value of the Clare’s equation to measure the usefulness:

$$usefulness(c_i) = 1 - \left(\frac{a(c_i) \log_2 treesize(c_i)}{max} \right) \quad (1)$$

where:

- $treesize(c_i) = 1 +$ number of descendant classes of c_i (1 is added to represent c_i itself)
- $a(c_i) = 0$, if $p(c_i) = 0$; $a(c_i) =$ a user defined constant (default=1) otherwise.
- max is the highest value obtained by computing $a(c_i) \log_2 treesize(c_i)$ and it is used to normalize all the other values into the range $[0, 1]$.

To make the final classification decision, the global-model naive Bayes has two options. The first option is to assign the final class label with the maximum value of posterior probability (Equation 2). The second option is to assign the class label which maximizes the product of the posterior probability and usefulness (Equation 3).

$$classify(\mathbf{A}) = \arg \max_{class} \prod_{i=1}^n (A_{i=V_{ij}}|Class) \times P(Class) \quad (2)$$

$$classify(\mathbf{A}) = \arg \max_{class} \prod_{i=1}^n ((A_{i=V_{ij}}|Class) \times P(Class)) \times Usefulness(Class) \quad (3)$$

IV. EXPERIMENTAL DETAILS

A. Establishing a Baseline Method

An important issue when dealing with hierarchical classification is how to establish a meaningful baseline method. Since we are dealing with a problem where the classifier’s most specific class prediction for an example can be at any level of the hierarchy (non-mandatory leaf node prediction – see Section 2), it is fair to have a comparison against a method whose most specific class prediction can also be at any level in the class hierarchy.

Therefore, in this work, as a baseline method, we use the same broad type of classifier (Naive Bayes), but with a conventional local-model approach with a top-down class prediction testing approach. More precisely, during the training phase, for every non-leaf class node, a naive Bayes multi-class classifier was trained to distinguish between the node’s child classes. To implement the test phase, we used the top-down class prediction strategy (see Section 2) in the context of a non-mandatory leaf-node class prediction problem. The

Table I
BIOINFORMATICS DATASETS DETAILS.

Protein Type	Signature Type	# of Attributes	# of Examples	# Classes/Level
Enzyme	Interpro	1,216	14,027	6/41/96/187
	Pfam	708	13,987	6/41/96/190
	Prints	382	14,025	6/45/92/208
	Prosites	585	14,041	6/42/89/187
GPCR	Interpro	450	7,444	12/54/82/50
	Pfam	75	7,053	12/52/79/49
	Prints	283	5,404	8/46/76/49
	Prosites	129	6,246	9/50/79/49

criterion for deciding at which level to stop the classification during the top-down classification process is based on the usefulness measure (see Section 3).

Since we already have the measure for usefulness of a predicted class, we decided to use the following stopping criterion: If $p(c_i) \times usefulness(c_i) > p(c_j) \times usefulness(c_j)$ for all classes c_j that are a child of the current class c_i , then stop classification. In other words, if the posterior probability times the usefulness (given by Equation 1) computed by the classifier at the current class node is higher than the posterior probability times the usefulness computed for each of its child class nodes, then stop the classification at the current class node – i.e., make that class the most specific (deeper) class predicted for the current test example.

B. Bioinformatics Datasets Used in the Experiments

In this work we have used datasets about two different proteins families: Enzymes and GPCRs (G-Protein-Coupled Receptors). Enzymes are catalysts that accelerate chemical reactions while GPCRs are proteins involved in signalling and are particularly important in medical applications as it is believed that from 40% to 50% of current medical drugs target GPCR activity [13].

Each dataset [14] has four different versions based on different kinds of predictor attributes, and in each dataset the classes to be predicted are hierarchical protein functions. Each type of binary predictor attribute indicates whether or not a “protein signature” (or motif) occurs in a protein. The motifs used in this work were: Interpro Entries, FingerPrints from the Prints database, Prosites Patterns and Pfam. Apart from the presence/absence of several motifs according to the signature method, each protein has two additional attributes: the molecular weight and the sequence length.

Before performing the experiments, the following pre-processing steps were applied to the datasets: (1) Every class with fewer than 10 examples was merged with its parent class. If after this merge the class still had fewer than 10 examples, this process would be repeated recursively until the examples would be labeled to the Root class. (2) All examples whose most specific class was the Root class were removed. (3) A class blind discretization algorithm based on equal-frequency binning (using 20 bins) was applied to the molecular weight and sequence length attributes, which were

the only two continuous attributes in each dataset. Table I presents the datasets’ main characteristics after these pre-processing steps. The last column of Table I presents the number of classes at each level of the hierarchy (1st/2nd/3rd/4th levels).

V. COMPUTATIONAL RESULTS

In this section, we are interested in answering the following questions by using controlled experiments: (a) How does the choice of a local (with top-down class prediction approach) or global (with the proposed method) approach affects the performance of the algorithms? (b) How does the inclusion of the usefulness criterion (Equation 1) affects the global model Naive Bayes algorithm? All the experiments reported in this section were obtained by using the datasets presented in Section IV-B, using stratified ten-fold cross-validation.

In order to evaluate the algorithms we have used the metrics of hierarchical precision (hP), hierarchical recall (hR) and hierarchical f-measure (hF) proposed in [15]. These measures are extended versions of the well known metrics of precision, recall and f-measure but tailored to the hierarchical classification scenario. They are defined as follows: $hP = \frac{\sum_i |\hat{C}_i \cap \tilde{C}_i|}{|\hat{C}_i|}$, $hR = \frac{\sum_i |\hat{C}_i \cap \tilde{C}_i|}{|\tilde{C}_i|}$, $hF = \frac{2 * hP * hR}{hP + hR}$, where \hat{C}_i is the set consisting of the most specific predicted class and all its ancestor classes and \tilde{C}_i is the set consisting of the true class and all its ancestor classes. The main advantage of using this particular metric is that it can be applied to any hierarchical classification scenario (i.e. single label, multi-label, tree-structured, dag-structured, mandatory-leaf node or non-mandatory leaf node problems). In addition, this measure penalizes shallow predictions because such predictions would have relatively low recall values, therefore introducing some pressure for predictions to be as deep as possible (to increase recall) as long as precision is not too compromised. This approach to cope with the trade-off between precision and recall is suitable to our non-mandatory leaf-node prediction problem.

To measure if there is any statistically significant difference between the hierarchical classification methods being compared, we have employed the Friedman test with the post-hoc Shaffer’s static procedure for comparison of multiple classifiers over many datasets as strongly recommended

Table II
HIERARCHICAL PRECISION (hP), RECALL (hR) AND F1-MEASURE (hF) ON THE HIERARCHICAL PROTEIN FUNCTION DATASETS.

Databases	LMNBwU			GMNB			GMNBwU		
	hP	hR	hF	hP	hR	hF	hP	hR	hF
GPCR-Interpro	70.49	67.29	67.90	87.60	71.33	77.01	84.39	74.76	78.27
GPCR-Pfam	66.49	59.17	61.32	77.23	57.52	64.40	70.35	60.13	63.53
GPCR-Prints	70.13	66.32	66.99	87.06	69.42	75.38	83.04	73.00	76.51
GPCR-Prosite	63.45	55.95	58.11	75.64	53.73	61.14	66.38	56.61	59.89
EC-Interpro	74.85	80.23	76.64	94.96	89.58	90.53	94.07	92.84	92.65
EC-Pfam	74.94	79.73	76.47	95.15	86.94	88.72	93.69	92.25	92.13
EC-Prints	78.35	82.73	79.79	92.21	87.26	87.98	90.96	90.62	89.92
EC-Prosite	81.73	86.52	83.20	95.14	89.53	90.70	93.38	92.45	92.01

by [16]. Table III presents the results of this test using the values of hierarchical F-measure. The first column of Table III presents which classifiers are being compared. The second column presents the p value of the statistical test, which needs to be lower than the corrected critical value shown on the third column, in order to have a statistical significance difference between the performance of two classifiers at a confidence level of 95%.

A. Evaluating the Local Vs. Global Model Approaches

We first evaluate the impact of the usefulness component in the different types of hierarchical classification algorithms. Table II presents the results comparing the baseline local-model naive Bayes (LMNBwU) described in Section IV-A with the proposed global-model naive Bayes (GMNBwU), both with usefulness.

For all eight datasets the proposed global-model with usefulness obtained significantly better results than the local-model with usefulness. The statistical significance of the detailed results shown in Table II is confirmed by the first row of Table III, where the p value is much smaller than the corrected critical value. The same result is achieved by the global-model naive bayes without usefulness as shown in Table II and confirmed by the second row of Table III.

These results corroborate the ones reported in [9] where a global-model decision-tree approach was also better than a local-model one. Most previous studies comparing the local-model and global-model approaches have focused on mandatory leaf node prediction problems [8] [17], which is a simpler scenario – since there is no need to decide at which level the classification should be stopped for each example and there is no need to consider the trade-off between predictive accuracy and usefulness.

B. Evaluating the impact of the usefulness measure in the global-model Naive Bayes

Let us now evaluate the impact of the optional usefulness criterion in the proposed global-model naive Bayes, which considers the trade-off between accuracy and usefulness when deciding what should be the most specific class-predicted for a given test example. Table II shows the hierarchical measures of precision, recall and f-measure of

the global-model Naive Bayes without (GMNB) and with the usefulness criterion (GMNBwU).

The analysis of the results collaborate with our previous statements. That is, the GMNB has an overall higher hierarchical precision than the GMNBwU, while the GMNBwU have a higher overall hierarchical recall than the GMNB. This means that that by adding the usefulness to the global-model naive bayes, the classifier is really making deeper predictions at the cost of its precision. It should be noted however, that there is no statistically significant difference between the two classifiers, as show in the third row of Table III.

The decision of which version of the classifier to use will depend on the type of protein being studied and the costs associated with the biological (laboratory) experiments in order to verify if the predictions are correct.

Table III
STATISTICAL TESTS TABLE FOR $\alpha = 0.05$

algorithm	p	Shaffer
LMNBwU vs. GMNBwU	4.6525E-4	0.01666
LMNBwU vs. GMNB	0.0124	0.05
GMNB vs. GMNBwU	0.3173	0.05

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a novel algorithm that is an extension of the Naive Bayes algorithm to handle hierarchical classification problems by producing a single global classification model – rather than a number of local classification models as in the conventional local classifier approach with the top-down class prediction approach. Moreover, contrary to the usual scenario of hierarchical classification problems where the algorithm has to predict one of the leaf classes for each test example, in this work we dealt with the less-conventional scenario where the algorithm can predict, as the most specific class for a test example, a class at any level of the hierarchy (also known as a non-mandatory leaf-node prediction problem). In this scenario, we have chosen to combine the posterior probability of each class with the notion of prediction usefulness based on class depth, since deeper classes tend to be more useful (more informative) to the user than shallower classes.

In order to perform the experiments, we employed suitable hierarchical classification measures to this non-mandatory leaf-node prediction scenario and also established a meaningful baseline hierarchical classification method by modifying a local classifier approach with the top-down class prediction approach to take into account the same usefulness measure used by the proposed global-model algorithm.

The proposed global-model and the baseline local-model algorithms were evaluated on eight proteins datasets. The two versions of the proposed global-model algorithm achieved significantly better performance than the local-model approach. We also presented results showing that the notion of usefulness allows the algorithm to obtain a hierarchical f-measure similar to the one obtained without the use of usefulness but making more specific predictions, which tend to be more useful to the user. As future research, we intend to evaluate this method on a larger number of datasets and compare it against other global hierarchical classification approaches, like the ones proposed in [9], [18].

ACKNOWLEDGMENT

We want to thank Dr. Nick Holden for kindly providing us with the datasets used in this experiments. The first author is financially supported by CAPES – a Brazilian research-support agency (process number 4871-06-5).

REFERENCES

- [1] D. Tikk, G. Biró, and J. D. Yang, “A hierarchical text categorization approach and its application to frt expansion,” *Australian Journal of Intelligent Information Processing Systems*, vol. 8, no. 3, pp. 123–131, 2004.
- [2] D. W. Corne and G. B. Fogel, *Evolutionary Computation in Bioinformatics*. Morgan Kaufmann, 2002, ch. An Introduction to Bioinformatics for Computer Scientists, pp. 3–18.
- [3] J. A. Gerlt and P. C. Babbitt, “Can sequence determine function?” *Genome Biology*, vol. 1, no. 5, 2000.
- [4] A. A. Freitas and A. C. P. L. F. de Carvalho, *Research and Trends in Data Mining Technologies and Applications*. Idea Group, 2007, ch. A Tutorial on Hierarchical Classification with Applications in Bioinformatics, pp. 175–208.
- [5] A. Sun and E.-P. Lim, “Hierarchical text classification and evaluation,” in *Proc. of the IEEE Int. Conf. on Data Mining*, 2001, pp. 521–528.
- [6] D. Koller and M. Sahami, “Hierarchically classifying documents using very few words,” in *Proc. of the 14th Int. Conf. on Machine Learning*, 1997, pp. 170–178.
- [7] S. D’ Alessio, K. Murray, R. Schiaffino, and A. Kershenbaum, “The effect of using hierarchical classifiers in text categorization,” in *Proc. of the 6th Int. Conf. Recherche d’ Information Assistee par Ordinateur*, 2000, pp. 302–313.
- [8] E. Costa, A. Lorena, A. Carvalho, A. A. Freitas, and N. Holden., “Comparing several approaches for hierarchical classification of proteins with decision trees,” in *Advances in Bioinformatics and Computational Biology*, ser. Lecture Notes in Bioinformatics, vol. 4643. Springer, 2007, pp. 126–137.
- [9] C. Vens, J. Struyf, L. Schietgat, S. so Dzeroski, and H. Blockeel, “Decision trees for hierarchical multi-label classification,” *Machine Learning*, vol. 73, no. 2, pp. 185–214, 2008.
- [10] D. Heckerman, “A tutorial on learning with bayesian networks,” Microsoft, Technical Report MSR-TR-95-06, 1995.
- [11] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [12] A. Clare, “Machine learning and data mining for yeast functional genomics,” Ph.D. dissertation, University of Wales Aberystwyth, 2004.
- [13] D. Filmore, “It’s a GPCR world,” *Modern drug discovery*, vol. 7, no. 11, pp. 24–28, 2004.
- [14] N. Holden and A. A. Freitas, “Hierarchical classification of protein function with ensembles of rules and particle swarm optimisation,” *Soft Computing Journal*, vol. 13, pp. 259–272, 2009.
- [15] S. Kiritchenko, S. Matwin, and A. F. Famili, “Functional annotation of genes using hierarchical text categorization,” in *Proc. of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, 2005.
- [16] S. García and F. Herrera, “An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons,” *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.
- [17] M. Ceci and D. Malerba, “Classifying web documents in a hierarchy of categories: A comprehensive study,” *Journal of Intelligent Information Systems*, vol. 28, no. 1, pp. 1–41, 2007.
- [18] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, “Kernel-based learning of hierarchical multilabel classification models,” *Journal of Machine Learning Research*, vol. 7, pp. 1601–1626, 2006.