

A Review of Performance Evaluation Measures for Hierarchical Classifiers

Eduardo P. Costa *

Depto. Ciências de Computação
ICMC/USP - São Carlos
Caixa Postal 668, 13560-970, São Carlos-SP, Brazil

André C. P. L. F. Carvalho ‡

Depto. Ciências de Computação
ICMC/USP - São Carlos
Caixa Postal 668, 13560-970, São Carlos-SP, Brazil

Ana C. Lorena †

Universidade Federal do ABC
Rua Santa Adélia, 166 - Bairro Bangu
09.210-170 Santo André-SP, Brazil

Alex A. Freitas §

Computing Laboratory,
University of Kent, Canterbury,
CT2 7NF, UK

Abstract

Criteria for evaluating the performance of a classifier are an important part in its design. They allow to estimate the behavior of the generated classifier on unseen data and can be also used to compare its performance against the performance of classifiers generated by other classification algorithms. There are currently several performance measures for binary and flat classification problems. For hierarchical classification problems, where there are multiple classes which are hierarchically related, the evaluation step is more complex. This paper reviews the main evaluation metrics proposed in the literature to evaluate hierarchical classification models.

Introduction

Several criteria may be used to evaluate the performance of classification algorithms in supervised Machine Learning (ML). In general, different measures evaluate different characteristics of the classifier induced by the algorithm. Therefore, the evaluation of a classifier is a matter of on-going research (Sokolova, Japkowicz, & Szpakowicz 2006), even for binary classification problems, which involve only two classes and are the most studied by the ML community.

For classification problems with more than two classes, named multiclass problems, the evaluation is often more complex. Many of the existing evaluation metrics were originally developed for binary classification models. Besides the multiclass characteristic, in many problems, the classes are hierarchically related. The evaluation of hierarchical classifiers using measures commonly adopted for conventional flat classifiers does not contemplate the full characteristics of the hierarchical problem. Therefore, they are not adequate.

This paper presents a review of the main evaluation measures proposed to evaluate hierarchical classifiers. Another study of hierarchical evaluation measures is presented in (Kiritchenko 2005), but with a distinct emphasis. Different

from (Kiritchenko 2005), this work concentrates on general evaluation techniques for single label hierarchical classification.

This paper is organized as follows. Initially, the main types of hierarchical classification problems are briefly described. Next, the main evaluation measures for flat classification models are discussed. Later, measures proposed in the hierarchical context are addressed. In the end, the main conclusions are presented.

Hierarchical Classification

Given a dataset composed of n pairs (x_i, y_i) , where each x_i is an data item (example) and y_i represents its class, a classification algorithm must find a function which maps the data item to their correct classes.

The majority of classification problems in the literature involves flat classification, where each example is assigned to a class out of a finite (and usually small) set of flat classes. Nevertheless, there are more complex classification problems, where the classes to be predicted are hierarchically related (Freitas & Carvalho 2007; Sun, Lim, & Ng 2003a; 2003b). In these classification problems, one or more classes can be divided into subclasses or grouped into superclasses. These problems are known in the ML literature as hierarchical classification problems.

There are two main types of hierarchical structures: a tree and a Directed Acyclic Graph (DAG). The main difference between the tree structure (Figure 1.a) and the DAG structure (Figure 1.b) is that, in the tree structure, each node has just one parent node, while, in the DAG structure, each node may have more than one parent. For both structures, the nodes represent the problem classes and the root node corresponds to “any class”, denoting a total absence of knowledge about the class of an object.

Hierarchical classification problems often have as objective the classification of a new data item into one of the leaf nodes. The deeper the class in the hierarchy, the more specific and useful is its associated knowledge. It may be the case, however, that the classifier does not have the desired reliability to classify a data item into deeper classes, because deeper classes tend to have fewer examples than shallower classes. In this case, it would be safer to perform a classification into higher levels of the hierarchy.

In general, the closer the predicted class is to the root of

*ecosta@icmc.usp.br

†ana.lorena@ufabc.edu.br

‡andre@icmc.usp.br

§A.A.Freitas@kent.ac.uk

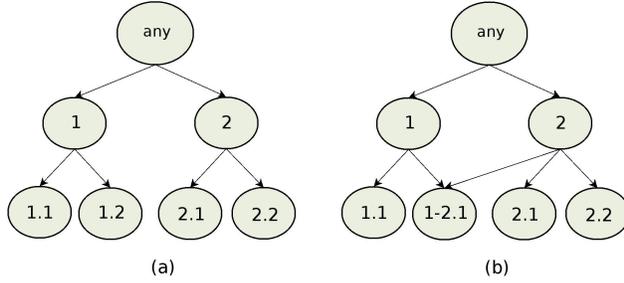


Figure 1: Examples of hierarchies of classes: (a) structured as a tree and (b) structured as a DAG.

the tree, the lower the classification error tends to be. On the other hand, such classification becomes less specific and, as a consequence, less useful. Therefore, a hierarchical classifier must deal with the trade-off class specificity versus classification error rate.

In some problems, all examples must be associated to classes in leaf nodes. These problems are named “mandatory leaf-node prediction problems”. When this obligation does not hold, the classification problem is an “optional leaf-node prediction problem”.

Following the nomenclature in (Freitas & Carvalho 2007), four approaches to deal with hierarchical problems with ML techniques may be cited: transformation of the hierarchical problem into a flat classification problem, hierarchical prediction with flat classification algorithms, top-down classification and big-bang classification.

The first approach reduces the original hierarchical problem to a single flat classification problem, which often considers only the leaf node classes from the hierarchy. This idea is supported by the fact that a flat classification problem may be viewed as a particular case of hierarchical classification, in which there are no subclasses and superclasses. Traditional approaches for flat classification may be applied in this context.

The second approach divides a hierarchical problem into a set of flat classification problems, usually one for each level of the hierarchy. Each class level is treated as an independent classification problem. Flat classification algorithms may then be used for each level.

In the top-down approach, one or more classifiers are trained for each level of the hierarchy. This produces a tree of classifiers. The root classifier is trained with all training examples. At the next class level, each classifier is trained with just a subset of the examples. E.g, in the class tree of Fig. 1(a), a classifier associated with the class node 1 would be trained only with data belonging to class 1.1 or 1.2, ignoring instances from classes 2.1 or 2.2. This process proceeds until classifiers predicting the leaf class nodes are produced. In the test phase, beginning at the root node, an example is classified in a top-down manner, according to the predictions produced by a classifier in each level. Although this approach produces a tree of classifiers, each classifier

is built by a flat classification algorithm. Its disadvantage is that errors made in higher levels of the hierarchy are propagated to the most specific levels.

In the big-bang approach, a classification model is created in a single run of the algorithm, considering the hierarchy of classes as a whole. Therefore, it increases the algorithmic complexity, but it can potentially avoid the previously mentioned disadvantage of the top-down approach. After the classifier training, the prediction of the class of a new data item is carried out in just one step.

Flat Performance Measures

Generally, the evaluation measures in classification problems are defined from a matrix with the numbers of examples correctly and incorrectly classified for each class, named confusion matrix. The confusion matrix for a binary classification problem (which has only two classes - positive and negative), is shown in Table 1.

Table 1: Confusion Matrix

True Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

The FP, FN, TP and TN concepts may be described as:

- False positives (FP): examples predicted as positive, which are from the negative class.
- False negatives (FN): examples predicted as negative, whose true class is positive.
- True positives (TP): examples correctly predicted as pertaining to the positive class.
- True negatives (TN): examples correctly predicted as belonging to the negative class.

The evaluation measure most used in practice is the accuracy rate (Acc). It evaluates the effectiveness of the classifier by its percentage of correct predictions. Equation 1 shows how Acc is computed, where $|A|$ denotes the cardinality of set A .

$$Acc = \frac{|TN| + |TP|}{|FN| + |FP| + |TN| + |TP|} \quad (1)$$

The complement of Acc is the error rate (Err) (Equation 2), which evaluates a classifier by its percentage of incorrect predictions. Acc and Err are general measures and can be directly adapted to multiclass classification problems.

$$Err = \frac{|FN| + |FP|}{|FN| + |FP| + |TN| + |TP|} = 1 - Acc \quad (2)$$

The recall (R) and specificity (Spe) measures evaluate the effectiveness of a classifier for each class in the binary problem. The recall, also known as sensitivity or true positive rate, is the proportion of examples belonging to the positive class which were correctly predicted as positive.

The specificity is the percentage of negative examples correctly predicted as negative. R and Spe are given by equations 3 and 4, respectively.

$$R = \frac{|TP|}{|TP| + |FN|} \quad (3)$$

$$Spe = \frac{|TN|}{|FP| + |TN|} \quad (4)$$

Precision (P) is a measure which estimates the probability that a positive prediction is correct. It is given by Equation 5 and may be combined with the recall originating the F-measure. A constant β controls the trade-off between the precision and the recall, as can be seen in Equation 6. Generally, it is set to 1.

$$P = \frac{|TP|}{|TP| + |FP|} \quad (5)$$

$$F - measure = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R} \quad (6)$$

Other common evaluation measure used in binary classification problems is the ROC (Receiver Operating Characteristics), which relates sensitivity and specificity. Although ROC curves were originally developed for two-class problems, they have also been generalized for multi-class problems (Everson & Fieldsend 2006). The reader should refer to (Bradley 1997) for a detailed description of ROC analysis.

All previous measures are inadequate for hierarchical classification models. By not taking into account the hierarchical structure of the problem, they ignore the fact that the classification difficulty tends to increase for deeper levels of the class hierarchy.

Consider, for example, the hierarchy presented in Fig. 1(a). Suppose that two examples, both from class 2.1, were incorrectly predicted. One was associated with class 2.2 and the other as belonging to class 1.1. In a uniform misclassification cost measure, as those previously presented, both misclassifications would be equally penalized. Therefore, the measure would not consider that, based on the hierarchy, classes 2.1 and 2.2 are more similar to each other than classes 2.1 and 1.1.

In order to avoid these deficiencies, new evaluation measures have been proposed, specific for hierarchical classification models. They are discussed in the next section.

Hierarchical Performance Measures

There are several alternatives to measure the predictive performance of a hierarchical classification algorithm. They can be generally grouped into four types: distance-based, depth-dependent, semantics-based and hierarchy-based. It should be noticed that some measures may use concepts from more than one approach. They were classified according to their most prominent characteristic.

In order to facilitate the understanding of how the evaluation of a hierarchical classifier may be performed, this section has the equations for the main evaluation measures described. The following notation was adopted: C_t represents

the true class of an example, C_p the predicted class and M is the number of classes in the problem.

Distance-based Measures

Classes that are close to each other in a hierarchy tend to be more similar to each other than other classes. Therefore, this approach considers the distance between the true class and the predicted class in the class hierarchy in order to compute the hierarchical classification performance. It was proposed in (Wang, Zhou, & Liew 1999) and used in (Sun & Lim 2001) in the context of hierarchical text classification. More precisely, Sun & Lim extended the conventional (flat classification-based) measures of precision, recall, accuracy and error rate for the context of distance-based hierarchical classification error. However, a drawback of this approach is that it does not consider the fact that classification at deeper class levels tends to be significantly more difficult than classification at shallower levels.

Sun & Lim first calculate the false positive contribution for each class C_i ($FpCon_i$) (Equation 7, using terms defined in equations 8 and 9). To calculate this contribution, an acceptable distance, denoted as Dis_{θ} , is defined. It should be specified by the user and should be larger than zero. These equations were adapted from (Sun & Lim 2001), where they were originally designed to evaluate multilabel hierarchical problems. A classification problem is named multilabel when a data may be associated to more than one class simultaneously. In the case of multilabel hierarchical classification problems, more than one path may be followed in the hierarchy for the classification of a new example.

$$FpCon_i = \sum_{\mathbf{x} \in FP_i} RCon(\mathbf{x}, C_p) \quad (7)$$

$$Con(\mathbf{x}, C_p) = 1.0 - \frac{Dis(C_p, C_t)}{Dis_{\theta}} \quad (8)$$

$$RCon(\mathbf{x}, C_p) = \min(1, \max(-1, Con(\mathbf{x}, C_p))) \quad (9)$$

For each class, first the contribution of each false positive (Con) is calculated (Equation 8). In this equation, \mathbf{x} denotes a data item and $Dis(C_p, C_t)$ denotes the distance between C_p and C_t . Next, a Refined-Contribution ($RCon$) is calculated (Equation 9), which normalizes the contribution of each example in the $[-1, 1]$ interval. Once the $RCon$ value has been obtained for each false positive, a summation is performed and the $FpCon_i$ value for class C_i is obtained, using Equation 7.

Besides $FpCon_i$, a false negative contribution ($FnCon_i$) is calculated for each class C_i . Its calculation is similar to that of $FpCon_i$, as can be seen in Equation 10. In the calculation of $RCon$ (Equation 9) and Con (Equation 8), C_p and C_t are replaced by C_t and C_p , respectively.

$$FnCon_i = \sum_{\mathbf{x} \in FN_i} RCon(\mathbf{x}, C_t) \quad (10)$$

Based on the $FpCon_i$ and $FnCon_i$ values, some extended measures may be defined: precision, recall, accuracy

rate and error rate, which are presented in equations 11, 12, 13 and 14, respectively.

$$P = \frac{\max(0, |TP_i| + FpCon_i + FnCon_i)}{|TP_i| + |FP_i| + FnCon_i} \quad (11)$$

$$R = \frac{\max(0, |TP_i| + FpCon_i + FnCon_i)}{|TP_i| + |FN_i| + FpCon_i} \quad (12)$$

$$Acc = \frac{|TN| + |TP| + FpCon_i + FnCon_i}{|FN| + |FP| + |TN| + |TP|} \quad (13)$$

$$Err = \frac{|FP| + |FN| - FpCon_i - FnCon_i}{|FN| + |FP| + |TN| + |TP|} \quad (14)$$

Depth-dependent Measures

The approach proposed by (Blockeel *et al.* 2002) tries to avoid the distanced-based measures' drawback by making classification error costs at shallower class levels higher than classification error costs at deeper class levels. In this extended measure, the distance between two classes is defined as a function of two factors, namely: (a) the number of edges between the predicted class and the true class in the graph representing the class hierarchy (where each node of the graph represents a class); and (b) the depth of the true and predicted classes in the class hierarchy. For the sake of simplicity, consider the case of a tree-based class hierarchy. One way of defining this function involves assigning a weight (cost) to each edge in the tree representing the class hierarchy. Hence, the classification error associating the difference between a predicted class and the true class is given by the summation of the weights of all the edges in the path between these two classes. In order to implement the principle that classification error costs at shallower class levels should be higher than classification error costs at deeper levels, the weights of edges at deeper levels tend to be smaller than the weights of edges at shallower levels.

Observe that this approach introduces the problem of how to set the weight for each edge. The solution used in (Holden & Freitas 2006) and (Blockeel *et al.* 2002) decreases the value of weights exponentially with the depth of the edge. Both works used these weights in the calculation of the accuracy rate. However, this solution presents its own problems. One of them occurs when the tree is very unbalanced in the sense that the tree depth (i.e., the distance between the root node and a leaf node) varies significantly for different leaf nodes. When this is the case, a misclassification involving a pair of predicted and true leaf classes at a level closer to the root will be less penalized than a misclassification involving a pair of predicted and true leaf classes at a level further from the root, simply because the latter will be associated with more edges in the path between the predicted and the true classes. It can be argued that this smaller penalization is not fair, because the classification involving the shallower level cannot reach a deeper level due to the limited depth of the former.

Another problem pointed by (Lord *et al.* 2003) also considers the case where the hierarchy depth varies significantly for different leaf nodes. According to Lord *et al.*, when two classes C_1 and C_2 are located in different subtrees of the root class - i.e., when the deepest common ancestor of both C_1 and C_2 is the root class - the fact that one of these classes is deeper than the other does not necessarily mean that the former is more informative to the user than the latter. For instance, a class at the third level of the tree can be associated with information as specific as a class at the eighth level of the tree, if the two classes are in different subtrees of the root class. Therefore, the assignment of weights considering only the depth of an edge - and not the information associated with the classes at the two end points of the edge - can be a problem.

There is another drawback in the distance-based and depth-dependent distance-based measures. Although the distance between the predicted and the true class is easily defined in the case of a tree-based class hierarchy, the definition of this distance in the case of a DAG-based class hierarchy is more difficult and involves more computational time. In particular, in a DAG, the concept of the "depth" of a class node is not trivial, since there can be multiple paths from the root node to a given class node.

Semantics-based Measures

This approach, also proposed by (Sun & Lim 2001), uses the concept of class similarity to calculate the prediction performance of the classification model. The inspiration to use similarity between classes to estimate the classification error rate is the intuitive notion that it is less severe to misclassify a new example into a class close to the true class than into a class with no relation to the true class.

This similarity may be calculated in different ways. In (Sun & Lim 2001), each class is described by a feature vector. For example, C_i is described by the feature vector $\{w_1, w_2, \dots, w_H\}$, where H is the number of features. The similarity between classes is calculated using these vectors. This similarity is later used to define the precision, recall, accuracy and error rates.

For each pair of classes C_i and C_j , the Category Similarity (CS) is calculated (Equation 15). The similarities between the categories can be used to define the Average Category Similarity (ACS), given by Equation 16. For each category, the CS and ACS values are used to calculate the contribution (Con) of the false positives and false negatives. For the false positive case, the calculation of Con is given by Equation 17. For the false negative case, the same equation is used replacing C_p and C_t by C_t and C_p , respectively. Once Con is calculated, the procedure to obtain the performance evaluation measures is the same followed for the distance-based measures, also proposed by (Sun & Lim 2001) and described in the subsection "Distanced-based Measures". Again, some adaptations were performed in the equations, which were originally proposed to multilabel classification problems.

$$CS(C_i, C_j) = \frac{\sum_{k=1}^H w_k \times v_k}{\sqrt{\sum_{k=1}^H w_k^2 \times \sum_{k=1}^H v_k^2}} \quad (15)$$

$$ACS = \frac{2 \times \sum_{i=1}^M \sum_{j=i+1}^M CS(C_i, C_j)}{M \times (M - 1)} \quad (16)$$

$$Con(\mathbf{x}, C_p) = \frac{CS(C_p, C_t) - ACS}{1 - ACS} \quad (17)$$

A problem with the semantics-based measures is that several hierarchies, particular those related to biology, already take some form of semantic relationships into consideration when they are built (Freitas & Carvalho 2007). In these situations, the classes closer in the hierarchy are also semantically more similar. Thus, the distance should also be considered in the prediction evaluation. The distance-independent semantic measures between pairs of classes could also be used to construct a new hierarchy and the error rates could then be evaluated using the distance between classes in this new hierarchy.

Hierarchy-based Measures

This approach uses concepts of ancestral and descendant classes to formulate new evaluation measures. One example of such approach is presented in (Ipeirotis, Gravano, & Sahami 2001). Ipeirotis et al. use the concept of descendant classes in their performance evaluation by considering the subtrees rooted in the predicted class and in the true class. Each subtree is formed by the class itself and its descendants. The intersection of these subtrees is then used to calculate extended precision and recall measures. To calculate the precision, the number of classes belonging to the intersection is divided by the number of classes belonging to the subtree rooted at the predicted class, as can be seen in Equation 18. In this equation, $Descendant(C)$ represents the set of classes contained in the subtree whose root is C . It corresponds to the descendants of C , including C itself.

$$hP = \frac{|Descendant(C_p) \cap Descendant(C_t)|}{|Descendant(C_p)|} \quad (18)$$

To calculate the recall, the number of classes belonging to the intersection is divided by the number of classes in the subtree rooted at the true class, as illustrated in Equation 19. Once the hierarchical prediction and recall measures have been calculated, they are used to calculate a hierarchical extension of the F-measure. The problem with this measure is its assumption that the predicted class is either a subclass or a superclass of the true class. When these classes are in the same level, for example, their intersection is an empty set.

$$hR = \frac{|Descendant(C_p) \cap Descendant(C_t)|}{|Descendant(C_t)|} \quad (19)$$

In a similar measure, Kiritchenko, Matwin & Famili (Kiritchenko, Matwin, & Famili 2004) uses the notion of ancestors in order to calculate the classification errors. For this, the authors consider the number of common ancestors in the true class and the predicted class. To calculate the precision, this value is divided by the number of ancestors of the

predicted class (Equation 20). It is important to notice that the set $Ancestor(C)$ includes the class C itself. Besides, the root node is not considered an ancestor from class C , because, by default, all examples belong to the root node.

$$hP = \frac{|Ancestor(C_p) \cap Ancestor(C_t)|}{|Ancestor(C_p)|} \quad (20)$$

To calculate the recall, the number of common ancestors is divided by the number of ancestors of the true class, as can be observed in Equation 21. As in the previous approach, these measures were used to calculate an extension of the F-measure, named hierarchical F-measure. Independently, a similar measure was proposed by (Eisner *et al.* 2005). It is an extension of the approach from (Poulin 2004) to calculate the precision and recall values for multilabel problems.

$$hR = \frac{|Ancestor(C_p) \cap Ancestor(C_t)|}{|Ancestor(C_t)|} \quad (21)$$

Other Evaluation Measures

Other measures were also investigated, as those defined in (Cesa-Bianchi, Gentile, & Zaniboni 2006), (Cai & Hofmann 2004), (Wu *et al.* 2005), (Wang, Zhou, & He 2001), (Lord *et al.* 2003) and (Verspoor *et al.* 2006). These other measures are not directly related with the previous categories.

The measures from (Cesa-Bianchi, Gentile, & Zaniboni 2006) and (Cai & Hofmann 2004) can be regarded as application-specific. Wu et al. and Wang, Zhou & He propose measures for the multilabel case, which lose their evaluation power when applied to non-multilabel problems, which are the focus of this paper. Besides, the measure proposed by (Wang, Zhou, & He 2001) is only used in the construction of the hierarchical classifier, and not in its evaluation. In (Verspoor *et al.* 2006), an extension of the measure defined in (Kiritchenko, Matwin, & Famili 2004) is proposed, adapting it to the multilabel context. Lord et al. proposed a similarity measure that combines the semantic information associated with a class with the structure of the class hierarchy. Nevertheless, this similarity was not used in the evaluation of a classification model. A possible extension of this measure allows the use of this similarity index in the evaluation of a hierarchical classification model, as performed in (Sun & Lim 2001).

Another approach that can be used to estimate the hierarchical classification error rates involves the use of a misclassification cost matrix. This approach, proposed in (Freitas & Carvalho 2007), is a generalization of the misclassification cost matrix for standard flat classification (Witten & Frank 2000). This matrix stores the pre-defined cost of each possible misclassification. One drawback of this approach is the definition of these costs, which may be a subjective task. Besides, for a classification problem with a large number of classes, a frequent scenario in hierarchical classification problems, the dimension of the cost matrix becomes too high, which makes it even more difficult the calculation of the costs.

Conclusions

This survey reviewed some of the main evaluation measures for hierarchical classification models.

It can be observed from the study conducted that there is not yet a consensus concerning which evaluation measure should be used in the evaluation of a hierarchical classifier. Several measures have been proposed, but none of them was adopted frequently by the ML community. As future work, it would be interesting to empirically compare the hierarchical classification evaluation measures. Until now, reported works usually compare one given hierarchical measure to a flat counterpart. However, a direct comparison of the relative effectiveness of different measures of hierarchical classification is far from trivial, because there is no clear definition of a criterion for choosing “the best measure”, out of different measures. However, there is a strong need for empirical comparisons of different hierarchical classification measures, to better identify their similarities and differences - e.g., to what extent they are correlated with each other.

Additionally, as several hierarchical classification problems are also multilabel, it is important to investigate the use of hierarchical evaluation measures in classification problems which are hierarchical and multilabel.

Acknowledgement

The authors would like to thank the financial support from the Brazilian research agencies FAPESP and CNPq.

References

- Blockeel, H.; Bruynooghe, M.; Dzeroski, S.; Ramon, J.; and Struyf, J. 2002. Hierarchical multi-classification. In *Proceedings of the ACM SIGKDD 2002 Workshop on Multi-Relational Data Mining (MRDM 2002)*, 21–35.
- Bradley, A. P. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30:1145–1159.
- Cai, L., and Hofmann, T. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the Thirteenth ACM conference on Information and knowledge management*, 78–87.
- Cesa-Bianchi, N.; Gentile, C.; and Zaniboni, L. 2006. Hierarchical classification: combining Bayes with SVM. In *Proceedings of the 23rd international conference on Machine learning*, 177–184. ACM Press New York, NY, USA.
- Eisner, R.; Poulin, B.; Szafron, D.; Lu, P.; and Greiner, R. 2005. Improving Protein Function Prediction using the Hierarchical Structure of the Gene Ontology. In *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology - CIBCB'05*, 1–10.
- Everson, R. M., and Fieldsend, J. E. 2006. Multi-class roc analysis from a multi-objective optimisation perspective. *Pattern Recogn. Lett.* 27(8):918–927.
- Freitas, A. A., and Carvalho, A. C. P. F. 2007. *A Tutorial on Hierarchical Classification with Applications in Bioinformatics*. In: D. Taniar (Ed.) *Research and Trends in Data Mining Technologies and Applications*. Idea Group. 176–209.
- Holden, N., and Freitas, A. A. 2006. Hierarchical Classification of G-Protein-Coupled Receptors with PSO/ACO Algorithm. In *Proceedings of the 2006 IEEE Swarm Intelligence Symposium (SIS-2006)*, 77–84.
- Ipeirotis, P.; Gravano, L.; and Sahami, M. 2001. Probe, count, and classify: categorizing hidden web databases. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, 67–78. ACM Press New York, NY, USA.
- Kiritchenko, S.; Matwin, S.; and Famili, A. 2004. Hierarchical Text Categorization as a Tool of Associating Genes with Gene Ontology Codes. In *Proceedings of the 2nd European Workshop on Data Mining and Text Mining for Bioinformatics*, 26–30.
- Kiritchenko, S. 2005. *Hierarchical Text Categorization and Its Application to Bioinformatics*. Ph.D. Dissertation, School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Ottawa, Canada.
- Lord, P.; Stevens, R.; Brass, A.; and Goble, C. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics* 19(10):1275–1283.
- Poulin, B. 2004. Sequenced-based protein function prediction. Master’s thesis, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.
- Sokolova, M.; Japkowicz, N.; and Szpakowicz, S. 2006. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Proceedings of the AAAI’06 workshop on Evaluation Methods for Machine Learning*, 24–29.
- Sun, A., and Lim, E. P. 2001. Hierarchical text classification and evaluation. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, 521–528. IEEE Computer Society Washington, DC, USA.
- Sun, A.; Lim, E. P.; and Ng, W. K. 2003a. Hierarchical text classification methods and their specification. *Cooperative Internet Computing* 256:18 p.
- Sun, A.; Lim, E. P.; and Ng, W. K. 2003b. Performance measurement framework for hierarchical text classification. *Journal of the American Society for Information Science and Technology* 54(11):1014–1028.
- Verspoor, K.; Cohn, J.; Mniszewski, S.; and Joslyn, C. 2006. A categorization approach to automated ontological function annotation. *Protein Science* 15(6):1544–1549.
- Wang, K.; Zhou, S.; and He, Y. 2001. Hierarchical classification of real life documents. In *Proceedings of the 1st (SIAM) International Conference on Data Mining*, 1–16.
- Wang, K.; Zhou, S.; and Liew, S. 1999. Building hierarchical classifiers using class proximity. In *Proceedings of the 25th International Conference on Very Large Data Bases*, 363–374.
- Witten, I. H., and Frank, E. 2000. *Data Mining - practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers.
- Wu, H.; Su, Z.; Mao, F.; Olman, V.; and Xu, Y. 2005. Prediction of functional modules based on comparative genome analysis and Gene Ontology application. *Nucleic Acids Research* 33(9):2822–2837.