Confusion and distance metrics as performance criteria for hierarchical classification spaces

Wilbert van Norden
Defence Materiel Organisation
CAMS – Force Vision
Den Helder, the Netherlands
w.l.van.norden@forcevision.nl

Catholijn Jonker
Man-Machine Interaction Group
Delft University of Technology
Delft, the Netherlands
c.m.jonker@tudelft.nl

***Abstract*—When intelligent systems reason about complex problems with a large hierarchical classification space it is hard to evaluate system performance. For classification problems, different evaluation criteria exist but these either focus on a belief expressed on all possible, mutually exclusive labels (soft classification) or they are based on the set of labels that are returned by a classifier (hard classification) for hierarchical labels. Measures to evaluate a classifier that assigns belief on all labels when these are hierarchical related however are lacking. This paper puts forward two new criteria for evaluation of soft output for hierarchical labels using a generic and flexible model of the solution space. The first criterion gives information on the accuracy of the system and the second on the robustness. Results with these new criteria are compared to existing criteria for a hierarchical classification task with different classifiers.**

***Index Terms*—Ontology modelling, Classification, Performance evaluation**

I. Introduction

In situated applications of intelligent systems, the used world model is important. This holds especially for applications where a (complex) classification solution space needs to be examined to find the desired answer. Hierarchical labels in classification increase the complexity of the solution space in contrast to solution spaces with mutually exclusive labels. Furthermore, different classifying agents may have a different world model and/or expertise. Getting one classification out of such a set of classifying agents requires the agents to cooperate and a good strategy for handling the possibly conflicting individual classifications.

The different classifying agents (CAs) assigned to execute a classification task, perform their task based on their own world-view, expertise. The agents might even use different sensors. An integrating agent (IA) has three important tasks in such a MAS. The first is finding a integrated world model to correctly combine beliefs held by the individual CAs. Secondly, it should combine the provided user information with the beliefs held by the CAs. Lastly, it will have to estimate individual classification performance of the various CAs to justify ignoring certain agents' belief in some scenarios. This paper uses such a MAS architecture for classification, see figure 1, and presents new criteria to evaluate system performance.

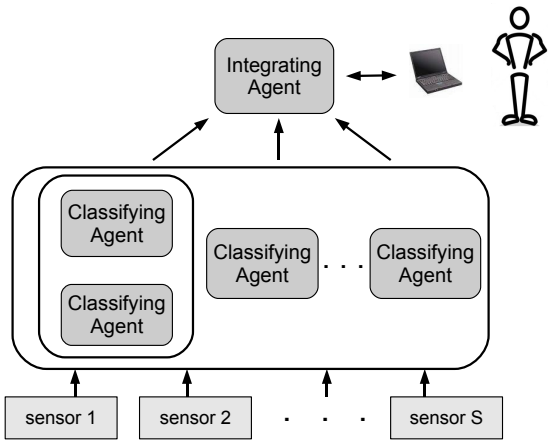
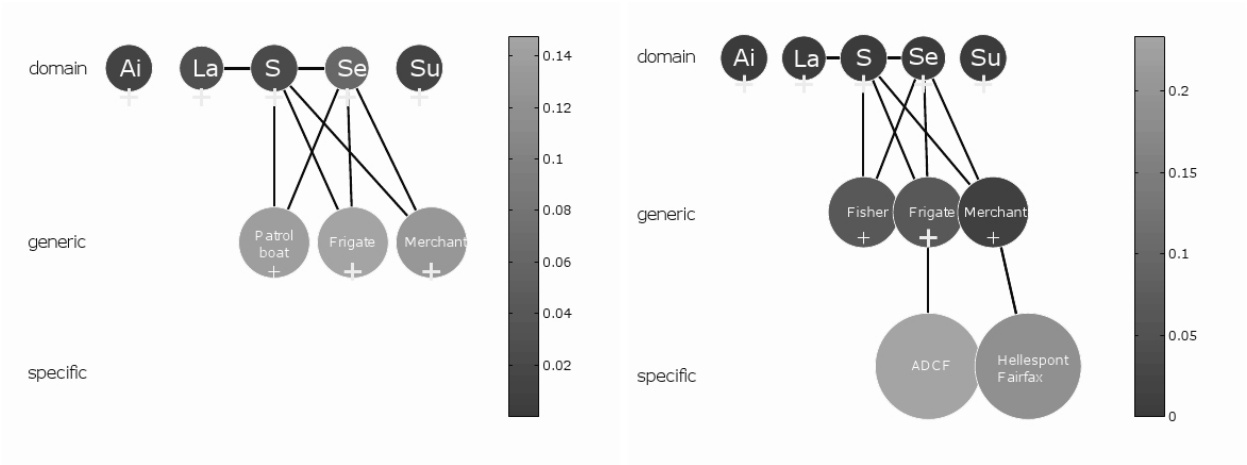


Fig. 1. System architecture

Yang proposes several measures for hierarchical classification, [1]. Though these criteria provide a good starting point for hierarchical solution spaces, they need to be expended since they assume that each label only has exactly one parent. Here, a more generic model is used with multiple parents. Furthermore, these criteria assume classifiers that produce labels (hard classifiers) instead of classifiers that assign a degree of belief on all labels in the classification space (soft classification).

We propose performance criteria that operate on different hierarchy levels and that can cope with soft classifiers. Using this approach means that performance no longer is represented by a single value but by performance measures on each specificity level. The advantage is that more insight is given in the strengths and weaknesses of the individual agents and the classification system as a whole. With these new measures different set-ups for the MAS are evaluated for a multi-class problem.

Section II explains why new test criteria are needed. Since the new criteria are based on a hierarchical classification space, this model is discussed in Section III and Section IV introduces our new criteria. The test scenario we use is discussed in Section V. The CAs and the IA used in testing are briefly



(a) Solution with agent *1*

(b) Solution with agent *2*

Fig. 2. Different classification solutions for an object with true classification ADCF

discussed in Section VI. Results based on traditional test criteria and results based on our new criteria are given in Section VII. Finally, Section VIII closes with some concluding remarks.

II. Traditional evaluation

In Fig. 2 the classification solution of two different CAs in a maritime domain is shown. These CAs assigned a normalised belief based on position on a sea chart and measured speed. Although CA *2*, Fig. 2(b), assigned much of its belief, namely *0.233*, to the correct solution (an Air Defence and Command Frigate), it also assigned much belief (*0.193*) to a wrong label. In contrast, CA *1* (Fig. 2(a)) has spread its belief over more generic labels, but all of them ships. In that sense, CA *1* admits to not having an exact solution whereas CA *2* suggests a more definitive answer, since it assigns belief to specific classes. Given the information that was available to both CAs, the latter seems more realistic since a specific distinction cannot be made based on only that information.

Comparing the output of CA *1* with that of CA *2* with an error-estimation criterion leads to the conclusion that CA *2* has better performance since it finds the right solution. For practical purposes however, the output of CA *1* is more desirable since it admits having uncertainty on various types of ships. This keeps a worst-case scenario (namely frigates which are considered be more of a threat) open that would be (wrongfully) neglected using the output of CA *2*. A test criteria for classifiers operating on non-exclusive classes should therefore take into account how CAs (or other classifiers) spread their belief over classes. This is accomplished by examining the entries in the confusion matrix (M) in more detail. Whereas the more traditional approach focusses mainly on the diagonal of M.

In multi-label learning (see e.g., [2]) the F_1 measure as proposed by Yang, [1], is usually used. This criterion is based on recall and precision. Recall is defined as the number correctly found labels divided by the number of correct labels. The precision is defined as the number of correctly found labels divided by the number of found labels. Re-writing these quantities for soft normalised classifier output means that for both recall and precision the denominator equals *1*. Thus the F_1 metric equals the precision (or the recall). All three metrics then have equal value which reduces their added value. New metrics are therefore needed for soft classifiers that are still based on a hierarchical solution space.

III. Solution space

The model for the solution space we use, is based on labels with different levels of specificity that may fully overlap. These different specificity levels lead to a hierarchical solution space. For K specificity levels and N_k elements on specificity level $k \in \{1, 2, \ldots, K\}$, each label in the solution space is denoted $\theta_{k,n}$ with $n \in \{1, 2, \ldots, N_k\}$ and all elements at the same level of specificity are mutually exclusive.

The non-exclusive elements in the model are called child and parent labels. Since they occur at different specificity levels, the a-th order ancestor element is given by (1) with $v \in \{1, 2, \ldots, N_{k-a}\}$, for $0 \leq (k-a) \leq K$. In (1), $\bowtie$ denotes the join-operator on label names. For notational ease we will further refer to labels as θ_i where a mapping Ω is used: $i = \Omega(k, n) = n + \sum_{w=1}^{k-1} N_w$.

$$\theta_{k,n}^{\uparrow a} = \bowtie \{\theta_{k-a,v} \mid \theta_{k,n} \cap \theta_{k-a,v} \neq \varnothing\} \tag{1}$$

When a new CA is introduced that uses a different world model, this should be inserted into the model at the appropriate specificity levels. The parent and child elements should also be indicated. Fitting a new world model into the existing model could either be done by an operator or e.g., by an IA using the methodology described by Taylor et al. in [3].

IV. New metrics

Section II showed that the diagonal of a confusion matrix does not give enough information. The criterion F_1 that is usually applied in multi-label situations does not necessarily give the required information because the classifiers can be soft. Two new metric types are therefore introduced based on the entire confusion matrix: confusion metrics and distance metrics. These metrics are inspired by the notion of the loss function from e.g., [4] and [5].

A. Confusion metrics

For evaluation of multi-label learning systems, the loss function counts the number of labels in the right branch of the classification tree that are found. This loss function may be used in various ways, see e.g., [4], [5], and [6].

These methods use the knowledge of parent and child relations. It seems logical to do something similar for soft classification. We look at the relevant values in the confusion matrix and sum those values. These *confusion metrics* examine the confusion between non-exclusive classes at different specificity levels. When classifying a car with ground truth "*2000 Volvo V70 T5*" e.g., the confusion values of all "*Volvo*" types except the "*2000 Volvo V70 T5*" itself are summed. Since many different subsets of cars are possible, this is repeated for each specificity level in the world model obtaining multiple values, e.g., for "*station-wagon*", or "*5-door car*".

In contrast to the loss function approaches, these confusion values are not summed over all the different ancestral orders. Instead, the a-th order ancestor confusion, denoted $B_a(\theta_i)$, on label θ_i is calculated by (2) where $j \neq i$ holds.

$$B_a(\theta_i) = \sum_{\substack{\theta_j \in \left(\theta_i^{\uparrow a}\right) \\ \theta_j \notin \left(\theta_i^{\uparrow a-1}\right)}} M_{i,j} \tag{2}$$

In (2) the confusion matrix is denoted M and a single values $M_{i,j}$ represent the mean value of belief a classifier assigns to label θ_j when the correct label is θ_i. For overall classifier evaluation, the mean value of the a-th order ancestor confusion over all labels is examined, $\overline{B_a}$. For hard classifiers with exclusive classes the mean value of the diagonal of the confusion matrix is often used, [7]. This value is produced by $\overline{B_0}$.

B. Distance metrics

Equation (2) indicates the total amount of confusion with the a-th order ancestors. Fig. 2 however shows, that this metric does not give all information. When a CA is wrong, it would be preferable that its confusion is evenly (or uniformly) spread over the child elements in the right branch. Our second criterion type, the distance metric, therefore determines the root-mean-square (RMS) distance of confusion values in the branch to the mean value for each specificity level.

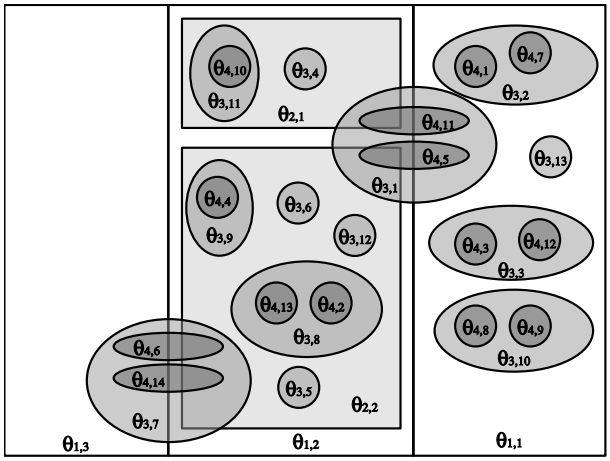


Fig. 3. Venn diagram of the example database

$$\delta_a(\theta_i)^2 = \frac{1}{P_a(\theta_i)} \cdot \sum_{\theta_j \cap \theta_i^{\uparrow a} = \theta_j} \left(M_{i,j} - \frac{B_a(\theta_i)}{P_a(\theta_i)} \right)^2 \tag{3}$$

The RMS distances for the a-th order ancestor confusion is denoted δ_a and is computed using (3) for each label θ_i with $j \neq i$. In order to calculate the mean value, the number of children elements of the a-th order ancestor are required. For notational ease, the number of labels in that set is denoted $P_a(\theta_i)$. To determine overall classifier performance the mean value over the class labels is determined, $\overline{\delta_a}$.

V. Test Scenario

To show the effectiveness of our evaluation criterion, we evaluate different classifiers with traditional evaluation criteria and with our new criteria. We use simulated data of a multi-class problem with non-exclusive classes. The classes are chosen to be very similar, since this also is the case in real-world applications such as ship recognition in maritime domains, crisis response, security systems and medical diagnostic systems.

A. Classification space

We created an integrated world model with *32* labels divided over *4* specificity levels, see the Venn Diagram in Fig. 3. For each of these labels some behaviour has been described where these descriptions are more specific for labels on more specific levels. For the least specific level, $k = 1$, uniform distributions are used whereas at the most specific level Gaussian or Laplace distributions are used. In general, when specificity increases, the (excess) kurtosis (see e.g., [8]) of the membership fields increase as well.

B. Train and test data

Train and test data is generated based on the membership fields that describe possible and normal behaviour. For each specific and generic class in the database *60* objects are

randomly generated, leading to a total of *1620* objects of which *33%* is used for training and the rest for testing.

VI. The Agents

To compare the different evaluation criteria, we evaluate three different types of CAs, namely

1) CAs that are based on standard trained classifiers ([7]), called trained CAs or TCAs,
2) CAs that are based on standard trained classifiers where different CAs are trained for different specificity levels, called model-based trained-CAs or MBTCAs,
3) CAs based on the model-based (MB) classification approach from [9]; referred to as model-based CAs, or MBCAs.

Two different integrating agents (IAs) are also considered, the first uses a model-driven combination rule. The second using a voting mechanism for the combination of belief.

A. (MB-)Trained classifying agents

For training the TCAs, different classifier techniques are used:

- *3*-Nearest Neighbour (*3*-NN) classifiers;
- Linear Distance Classifiers (LDCs); and
- Dissimilarity Classifiers (DisCs).

The MBTCAs are all based on LDCs based on initial test results. Each of these MBTCAs operates on a single specificity level in the integrated model of the classification space.

Classifier performance depends on the number of features that are used and feature evaluation is used to determine the most informative attributes, [7]. Based on initial results, the *3*-NN and the DisCs, are trained for two features, the LDC for three. For training and testing these classifiers the *Pattern Recognition Toolbox*[1] is used.

B. Model-Based Classifying Agents

A MBCA can be run for each membership field. Each MBCA works on the same principle: determine a Confidence Interval (denoted CI) based on known information and see how well this fits the membership field, [9]. These MBCA use the known dependencies between attributes to calculate a fitness whereas approaches like [10] use new models to deal with dependant attributes after which training is required.

$$CI = \prod_{j=1}^{J_\ell} \operatorname{erf}\left(\frac{\alpha}{\sqrt{2}}\right) \tag{4}$$

A membership field (Γ_X) of class X is described by J attributes. This field is a function of the values y_j that are determined for those attributes. For each value of the CI, a boundary value α is determined, see [9] which finds (4) for Gaussian distributed information about y_j where erf denotes the error function, [11]. Based on this boundary value, a contour line for a given CI is determined consisting of $\vec{y_j}$ values that constitute each contour line. Integrating over the contour line, sums the membership on the CI, this summed membership is denoted Φ_X and is a function of $\vec{y_j}$ that describes the contour line given α, (5). The total fitness on class X, is the integral over α of the summed membership values weighed by normalisation weight factor ($W(\alpha)$). For the boundaries of this integral we know that $\alpha \in [0, \infty\rangle$ since $CI \in [0, \ldots, 1]$. This fitness is denoted $m(X)$ and is given by (6).

$$\Phi_X(\alpha) = \underbrace{\int \cdots \int}_{j=1,\ldots,J} \Gamma_X(\vec{y_j})\, \mathrm{d}\vec{y_j} \tag{5}$$

$$m(X) = \int_0^\infty W(\alpha)\Phi_X(\alpha)\, \mathrm{d}\alpha \tag{6}$$

C. Integrating Agent

In this work we use two different types of IAs. The first is rather simple, it takes the average values of all CAs on the various labels in the integrated world model. This could mean that the added value of this world model is degraded. The required time for the IA however is still feasible for real-time application when combining the opinions of multiple CAs.

The second type of IA utilises the world model to calculate combined belief on the classification solution. The Proportional Redistribution Rule number 6 (PCR6) from [12] is used to achieve this, [13]. This rule uses the Dezert-Smarandache framework (see e.g., [14]) which can handle hierarchical solution spaces as well as conflicting and uncertain sources.

VII. Results

Six system set-ups are evaluated on the various evaluation criteria. Each of the three types of classifiers from Section VI are combined with both combination mechanism.

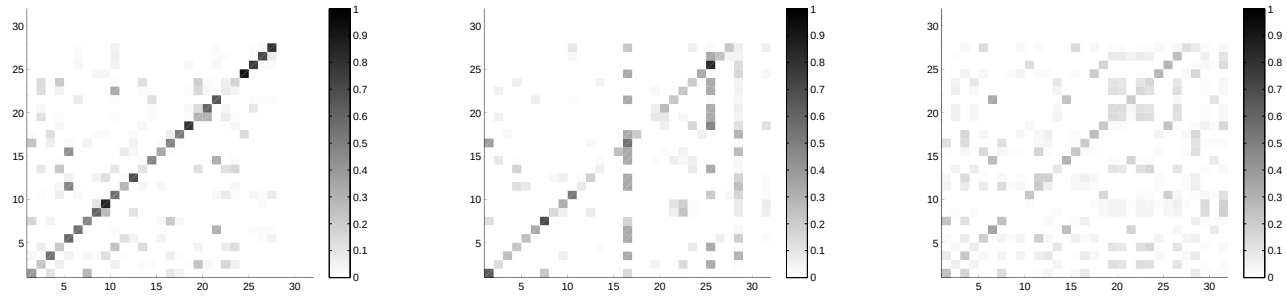
A. Error estimation

The error estimate is determined by counting how many test objects are not classified correctly based on the hard classification output. These error estimates are shown in Fig. 4 for the different system set-ups. All set-ups show a high error estimation rate. This is not unexpected since the class labels are very specific and the error estimation criterion does not take less generic answers into account.

B. Confusion

Traditionally, M is examined to compare classifiers in more detail. Fig. 6 shows M for the three types of CAs combined with PCR6. In figures 5(a) and 5(b) the downside of only considering the mean value of the confusion matrix diagonal

[1]From the Pattern Recognition Group of Delft University of Technology, www.prtools.org



(a) TCAs (b) MBTCAs (c) MBCAs

Fig. 5. Confusion matrices for various MASs when the IA uses PCR6

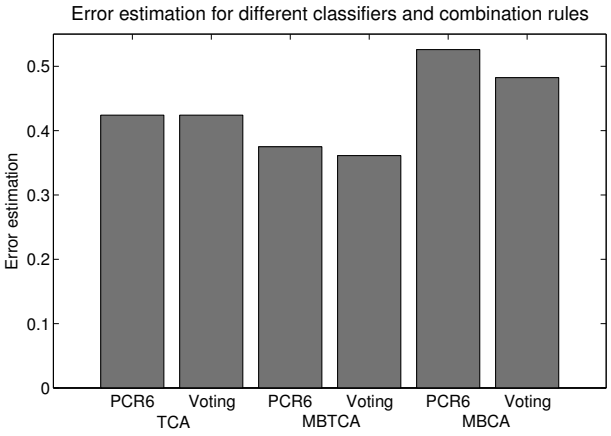


Fig. 4. Error estimations for various system set-ups

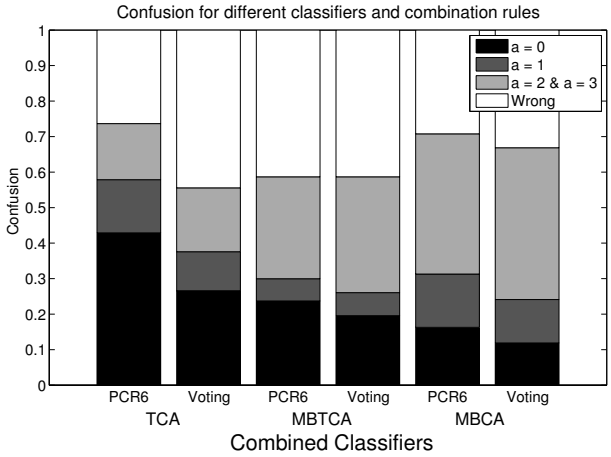


Fig. 6. Confusion distribution for various system set-ups

becomes apparent. This metric might be high because some classes are classified with a high precision whereas others are never classified correctly. In Fig. 5(c) the overall mean on the diagonal is low, but roughly the same for all classes, which might be desirable when robustness is desired.

For the system using MBTCAs an additional downside is visible in the confusion matrix. Distinct vertical lines show up in the visualisation of the confusion matrix. This means that despite the information, the agents have a certain bias for a small amount of classes.

C. New confusion metrics

Fig. 6 shows the mean branch confusion, $\overline{B_a}$. In this figure we see some interesting results when looking at the mean amount of wrongly labelled data from this approach. In Fig. 4 the highest error-rate was for the MBCAs combined with PCR6, the highest mean value in the confusion matrix on wrong labels however is assigned by TCAs combined with the voting IA. Second worst on this criterion is the system with the MBTCAs. Remarkable, since this scored best based on the error-estimation criterion.

In Fig. 6 the advantage of PCR6 over a simple voting strategy is also visible. That this effect occurs most with the TCAs is expected. These do not take the interrelations between labels into account whereas the MBCAs and the MBTCAs do. The knowledge of the solution space that PCR6 uses therefore has most effect on the TCAs. In general, the more knowledge of the model is used by the CAs, the less difference between the voting algorithm and PCR6 occurs.

D. Distance metrics

In Fig. 7 the results are shown for the RMS deviations. These results support the conclusions that the MBTCAs do not give the best results and that the MBC yield the best results looking at how belief is spread over less specific classes. They also show a smaller RMS deviation on the diagonal of the confusion matrix. This means that this system set-up has a stable performance on all classes. In contrast, the TCAs are only good at classifying a limited number of classes.

The overall conclusion is that the MBC combined with PCR6 has best performance. This fits well with the expectations since trained classifiers focus on dissimilarities — based on the assumption of exclusiveness — that do not occur in the solution space. The MB approaches however do not search for dissimilarities but utilises the non-exclusiveness of classes.

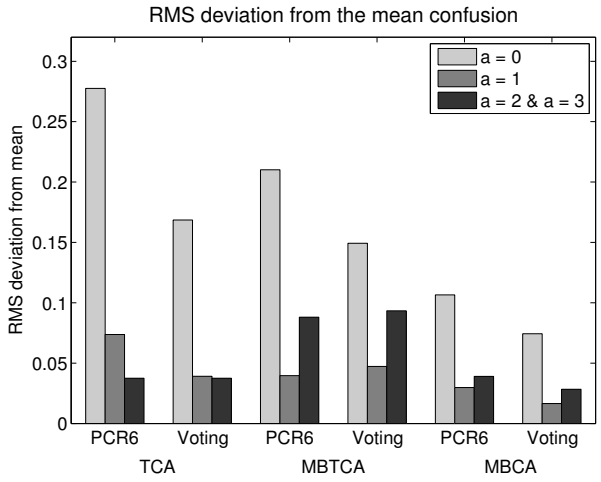


Fig. 7. RMS deviations for various system set-ups

VIII. Conclusions

Existing performance criteria are either based on soft classifier output for exclusive solution spaces, or hard classifier output in hierarchical ones. However, criteria that can cope with the complexity of hierarchical solution spaces and that can deal with soft classifier output did not exist yet.

Based on a generic hierarchical solution space we showed that such criteria can be found. For each specificity level a criteria is introduced for both accuracy and robustness. A numerical example in which various types of agents with different types of integrating agents were compared based on the new criteria as well as existing ones. The results showed that using the right criteria results in different conclusions on system performance.

References

[1] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.

[2] 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.

[3] M. E. Taylor, C. Matuszek, B. Klimt, and M. Witbrock, "Autonomous classification of knowledge into an ontology," in *Proceedings of the 20th International FLAIRS Conference*, May 2007.

[4] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 209–216.

[5] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *Journal of Machine Learning Research*, vol. 7, pp. 31–54, 2006.

[6] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proceedings of the 13th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2004, pp. 78–87.

[7] F. van der Heijden, R. P. Duin, D. de Ridder, and D. M. Tax, *Classification, parameter estimation and state estimation - an engineering approach using Matlab*. Chichester, England: John Wiley & Sons, 2004.

[8] D. Joanes and C. Gill, "Comparing measures of sample skewness and kurtosis," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 47, no. 1, pp. 183–189, 1998.

[9] W. L. van Norden, F. Bolderheij, and C. M. Jonker, "Classification support using confidence intervals," in *Proceedings of the 11th International Conference on Information Fusion*, 30 June – 3 July 2008, pp. 295–301.

[10] H. Langseth and T. D. Nielsen, "Classification using hierarchical naïve bayes models," *Journal of Machine Learning*, vol. 63, no. 2, pp. 135–159, 2006.

[11] E. W. Weisstein, "Erf." From MathWorld – A Wolfram Web Resource. http://mathworld.wolfram.com/Erf.html, internet, cited: May 2008.

[12] A. Martin and C. Oswald, *Advances and Applications of DSmT for Information Fusion (collected works)*. Rehoboth (MA): American Research Press, 2006, vol. 2, ch. 2. A new generalization of the proportional conflict redistribution rule stable in terms of decision, pp. 69–88.

[13] K. A. Scholte and W. L. van Norden, "Applying the PCR6 rule of combination in real time classification systems," in *Proceedings of the 12th International Conference on Information Fusion*, Seattle (WA), USA, 6–9 July 2009.

[14] F. Smarandache and J. Dezert, "An introduction to the DSm theory for the combination of paradoxical, uncertain and imprecise sources of information," in *Proceedings of the 13th International Congress of Cybernetics and Systems*, 6–10 July 2005.