Modeling hesitation and conflict : a belief-based approach for multi-class problems

Thomas Burger, Oya Aran, Alice Caplier

Modeling hesitation and conflict: A belief-based approach for multi-class problems

Thomas Burger
France Telecom R&D
28, Ch. Vieux Chêne,
Meylan, France
thomas.burger@orange-ftgroup.com

Oya Aran
Bogazici University
Dep. of Comp. Eng., Bebek 34342,
Istanbul, Turkey
aranoya@boun.edu.tr

Alice Caplier
INPG - LIS
46 av. Félix Viallet,
Grenoble,France
caplier@lis.inpg.fr

Abstract

Support Vector Machine (SVM) is a powerful tool for binary classification. Numerous methods are known to fuse several binary SVMs into multi-class (MC) classifiers. These methods are efficient, but an accurate study of the misclassified items leads to notice two sources of mistakes: (1) the response of each classifier does not use the entire information from the SVM, and (2) the decision method does not use the entire information from the classifier responses. In this paper, we present a method which partially prevents these two losses of information by applying Belief Theories (BTs) to SVM fusion, while keeping the efficient aspect of the classical methods.

1. Introduction

There are two kinds of MC classifiers: Those which directly handle all the classes, and those which fuse the decision of several binary classifiers in order to produce the final decision. Even if the first kind of classifiers seems straightforward, it is often more efficient to use classifiers of the second kind: Binary classifiers are simpler to implement and to train, and it is possible to combine them in order to fit the problem structure or prior knowledge.

SVMs are a good example of such tools which are extremely successful at binary classification tasks [1][2]. However, the combinatorial process they rely on, limits extensions to MC problems. Several methods exist to solve MC problem through combinations of SVMs or by directly defining MC objective functions [3]. However, none of these methods outperform the others and finding the optimal multi-class SVM classifier is an open research area. These points are presented in Section 2.

In section 3, we focus on BTs, from which our combination scheme is derived. In section 4, we present the application of BTs to the fusion of binary SVMs. Section 5 illustrates the method on various public datasets.

2. Support vector machines

2.1. Back to basic

Let x be an item for which a set of numerical attributes $(x_1, x_2, \ldots, x_n)$ is known. This item is supposed to belong to one of the two classes, C^1 or C^2. If for each class, the classifier "knows" that the attributes are correlated in a specific manner, it is possible to automatically find the class to which x belongs. In order to "teach" these correlations to the classifier, one relies on a statistically representative dataset of the type of items to classify. Unfortunately, as any representative statistical knowledge may be biased, it may not fit the real probabilistic distribution and may lead to misclassification.

As it is impossible to have an absolute prior knowledge on the bias of a training dataset, the main idea behind SVM strategy is to make sure that the separating hyperplane is the furthest possible of all the items of the training dataset. Hence, as long as its bias is smaller than the distance between the classes and the hyperplane (the margin), no misclassification can occur.

In practice, problems suffer from other difficulties, such as the intermingling of the classes (which is solved via the *slack variables trick* [2]), or the absence of linear separator (which is solved via the *kernel trick* [1]). This makes their implementation and use absolutely not trivial, but whatever the complexity, the main idea is that SVMs are able to provide the distance to the hyperplane with respect to the distance between the margins.

2.2. MC problems with SVM

Let C be a set of classes. To solve a MC problem on C with binary classifiers, most of the methods propose to project the training dataset on several binary sub-datasets. In such a case, sub-datasets are not necessarily of smaller size, but their two classes C^i and C^j are such that:

$$\forall i, j \in \lfloor 1, |C| \rfloor \quad C^i \cap C^j = 0 \quad \text{and} \quad C^i, C^j \in 2^C \quad \text{with } 2^C = \left\{ C^k / C^k \subseteq C \right\}$$

2^C is called the *powerset* of C. For each sub-dataset, a classifier gives a partial result. All the partial results are then fused to provide the final classification.

In case of SVM, the two most popular methods follow this principle:

- 1vs1 scheme: Considering C contains N classes, $N.(N-1)/2$ classifiers are taken into account, any of each trained on a sub-dataset only containing two classes C^i and C^j. The fusion process is a voting procedure.

***- 1vsAll scheme*:** Considering C contains N classes, N classifiers are taken into account, any of each trained on a sub-dataset containing the entire original training dataset, relabeled in C^i and $C\backslash C^i$. The fusion process uses the value of the decision function of each classifier and selects the one with the maximum.

These methods do not have a clear superiority to one another in terms of accuracy [4]. However *1vs1* scheme is faster since the sub-problems are easier to solve and thus more suitable for practical use [4].

For the *1vs1* scheme, several improvements exist. They are supposed to give more accurate results in the case of ties, (such as extracting posterior probabilities of each class and apply a weighted voting), but it appears that such methods are more or less equivalent in practice [4].

2.3. Two drawbacks of the voting procedures

Here are presented two of the main drawbacks of this voting procedure as a fusion scheme.

First, in this voting procedure, only the sign of the decision value is taken into account: if SVM positions an example nearby its hyperplane (likely to be wrong), and another SVM positions it far away from its margin (likely to be right) they should not have the same influence in the global decision process.

Secondly, even if dedicated rules are programmed to deal with ties in the voting, such a procedure does not properly handle all the possible contradictions between binary classifiers (when two or more classifiers have responses which are incoherent between each other, such as when C^1 is preferred than C^2, C^2 is preferred than C^3, and C^3 is preferred than C^1, or when two different classes are preferred than all the others). Actually, such contradictions may lead to *undetermined cases* (a situation in which, with respect to the training, it is impossible to expect a good answer from the classifier).

To deal with the *undetermined cases* depending on the risk of the misclassifications, the operator may choose a peculiar strategy*,* such as (1) the cautious one which rejects any undetermined item or (2) the betting one, which classifies it in the most likely class, or any other strategy in between.

There are three kinds of different *undetermined cases*, which should be differentiated:

- Incoherence: the relations between the attributes of the item to classify do not fit any of the classes. The most classical strategy is to reject such an item (Figure 1a).

- Uncertainty: the relations between the attributes of the item to classify do not really fit any of the classes, but it may be due to the bias of the statistical representation of the classes used during the training. Such an item should either be (1) rejected or (2) classified in the most likely class, or (3) remain unclassified (Figure 1b).

- Doubt: the item to classify equally fit several classes. This item is not supposed to be rejected nor randomly classified (Figure 1c).

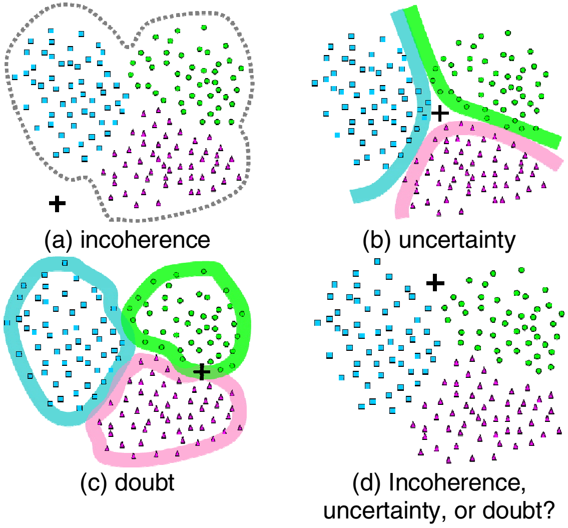


Figure 1: The various cases for an undetermined item are in practical difficult to discriminate

It is really difficult to reject items because of *incoherence* with SVM. One needs to have either (1) a training dataset made of such rejected items or (2) a background model which is formed only by positive examples. This is beyond our scope and we do not deal with *incoherence* in this paper. Nonetheless, *uncertainty*, *incoherence* and *doubt* are difficult to discriminate as the bias of the training is unknown (Figure 1d).

In this paper, we propose a fusion method which:

- deals with the interest of each partial result in a more refined manner that a simple voting process,
- allows the operator to choose a peculiar strategy for undetermined items and to classify them with respect to their nature (*doubt* or *uncertainty*) and the chosen strategy, in a manner directly rooted in the fusion process.

Such a method heavily relies on the belief-function theory, which is presented in the next section.

3. Introduction to Belief Theories

Belief theories refer to numerous model based on belief functions. Originally introduced by Dempster in the study of the bounds of a family of probabilities, it has been theorized by Shafer [5]. It has also been adapted or compared since then to various purposes in information theory, such as data fusion [7], fuzzy measure, possibility theory [9], and Bayesian theory [6]. Our goal is not to discuss these interpretations, so we globally refer to them as Belief Theories (BTs). We present here the main concept we use.

3.1. Belief functions

Let Ω be the set of N exclusive hypothesizes $h_1 \ldots h_N$. We call Ω the frame of discernment. Let $m(.)$ be a belief function on 2^Ω (the powerset of Ω). $m(A)$ represents our mass of belief in the proposition A (A corresponds to an element of 2^Ω, or equivalently, to a subset of Ω) :

$$\begin{aligned} m : 2^\Omega &\rightarrow [0,1] \\ A &\mapsto m(A) \qquad \text{with} \quad \sum_{A \subseteq \Omega} m(A) = 1 \end{aligned}$$

Note that,

- contrarily to probabilistic models, the belief can be assigned to non-singleton propositions, which allows modeling the *hesitation* between elements (which can be due to both doubt and uncertainty).

- Ø belongs to 2^Ω. A belief in Ø corresponds to *conflict* in the model, throughout an assumption in an *undefined hypothesis* of the frame of discernment (incoherence), or throughout a *contradiction* between the information on which the decision is made (uncertainty).

Providing such modeling through probability would be more difficult: the power of BTs is to allow *hesitation* and *conflict* to be modeled in a more refined manner than equi-probabilities (on which strong assumptions are made on missing information). The direct consequence is that no information is lost by such a modeling.

3.2. Fusion process

Let us explain how to combine several belief functions into a new belief function (under associativity and symmetry assumptions). In that purpose, we use the conjunctive combination. For N belief functions $m_1, \ldots, m_N$ from N sources, it is defined as:

$$\begin{aligned} m_\cap &= m_1 \bullet m_2 \bullet \ldots \bullet m_N \\ m_\cap &: \left(2^\Omega\right) \rightarrow [0,1] \\ m_\cap(A) &= \sum_{A_1 \cap \ldots \cap A_N = A} \left(\prod_{n=1}^{N} m_n(A_n) \right) \qquad \forall A \subseteq 2^\Omega \end{aligned}$$

Let us have a simple example with only two beliefs to fuse: each of them has partial information on the color of the item to classify (Red, Green or Blue). The combined belief function, (with $N = 2$), is formulated as:

$$\begin{aligned} m_\cap(A) &= m_1(A) \bullet m_2(A) \\ &= \sum_{A_1 \cap A_2 = A} \left(\prod_{n=1}^{2} m_n(A_n) \right) \\ &= \sum_{A_1 \cap A_2 = A} \left(m_1(A_1) \cdot m_2(A_2) \right) \end{aligned}$$

The conjunctive combination is a sum (with a peculiar pattern) over a product of all the possible elements of the powerset of each original belief. One can represent this product, $m_1(A_1) \cdot m_2(A_2)$, on a 2-dimensional table, in which each entry corresponds to one of the two original belief to fuse. Each cell is filled by the value of the product of the entries (cf. Table 1).

Table 1: Conjunctive combination of two sources

	m_1 (Ø)	m_1(BLUE)	m_1(GREEN)	m_1(RED)	m_1(B.∪ G.)	m_1(B.∪ R.)	m_1(R.∪ G.)	m_1 (Ω)
m_2 (Ø)								
m_2 (BLUE)		🟦			🟦	🟦		🟦
m_2 (GREEN)			▨		▨		▨	▨
m_2 (RED)				🟥		🟥 x	🟥	🟥
m_2 (B.∪ G.)		🟦	▨		🟨	🟦	🟩	🟨
m_2 (B.∪ R.)		🟦		🟥	🟦	🟪	🟥	🟪
m_2 (R.∪ G.)			▨	🟥	🟩	🟥	▦	▦
m_2 (Ω)		🟦	▨	🟥	🟨	🟪	▦	⬛

Then the sum, $\sum_{A_1 \cap A_2 = A} (\ )$, simply corresponds to a special pattern on which the content of the cells of the table are summed in order to produce the new belief function. Hence, in Table 1, all the cells with the same background texture or color are summed and the value is attributed to the element corresponding of the powerset.

In the general case of N belief functions, the principle is exactly the same (on a N-dimensional table).

4. Belief Theories combined with SVM

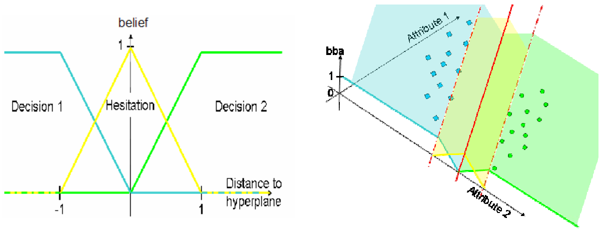
Our purpose is to associate a belief function on 2^C to each SVM decision in order to fuse them in a single final belief function. This permit that, (1) the amount of belief is related to the distance to the hyperplane so that this distance is taken into account and (2) the partial decision are fused in a manner that prevents any loss of information. Then we balance the two drawbacks spot in §2.3.

To perform such things, one needs, (1) to transform the SVM output, and (2) to implement the conjunctive combination. The first point is the subject of this section, whereas we consider the implementation aspects to be beyond the scope of this paper.

4.1. SVM output transformation

Intuitively, the SVM margins are self-designated borders to separate regions of certitude from regions of hesitation in the attribute space. An obvious way to create a belief function that models such a belief is to use fuzzy sets [10] (Figure 2).

Then, it is possible to associate a belief to each SVM output, so that they are fused together thanks to the conjunctive combination and a decision is made on the entire set of classes.



(a) (b)

Figure 2: (a) We define fuzzy sets on the distance to the hyperplane, (b) which model belief functions on the attribute space

4.2. Discussion on the hesitation pattern

Hesitation can be tuned by modifying the corresponding fuzzy set distribution. The one we propose first (Figure 2 and Figure 3) is the most natural one, and no prerogative is made on its definition.

- ***The purpose of the distribution*** of the hesitation is not an interesting point, as the only interest is to roughly model a lack of knowledge. That is the reason why we are not interested in patterns such as Figure 3b. It corresponds to a model with more parameters that need a prior knowledge to tune (a *second order* model, whereas we have assumptions only on *first order* model).

- ***The purpose of the width*** of the hesitation model has two aspects: first of all, from SVM point of view, the margin size is related to the SVM tuning, and the doubt model can be tuned by simply being supported by the margin. Secondly, from BTs point of view, hesitation and conflict are dual concepts and the tuning of the hesitation distribution is to be related to the balance desired between this two contradictory notions. Let us imagine an item (to be classified between two classes) which is hesitation-prone for a source of belief. The conjunctive combination of such a source with another source gives either (1) hesitation if the other source hesitates, (2) or on the contrary certitude, if the other source is certain. If we reduce the hesitation distribution to the minimum (such as in Figure 3c), the first source gives a result which is certain but might be wrong. The conjunctive combination gives either (1) a conflict if the beliefs are not the same, (2) or, on the contrary, certitude if they concur. By modifying the hesitation pattern, the result of the combination evolves from {hesitation, certitude} to {conflict, certitude}.

Then, the suppression of the hesitation simply corresponds to a situation where the conflict is emphasized. It also corresponds to the situation of a binary decision procedure, where voting is replaced by a fusion method which points out undetermined items.

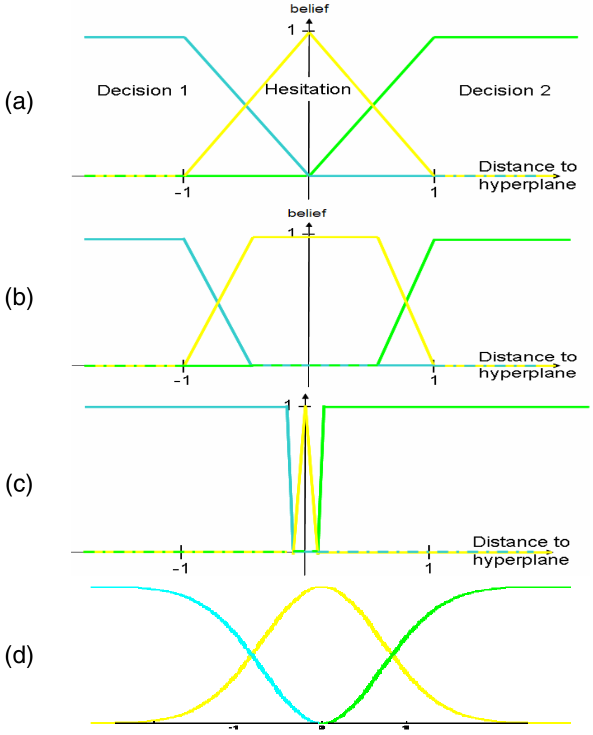


Figure 3: Four different tunings of the hesitation

On the contrary, if we enlarge the hesitation support, the system is less likely to consider an item as conflictive. When no conflict occurs anymore, we reach the limit of the hesitation modeling efficiency, and it is useless to consider a larger hesitation support. The absence of conflict corresponds to a hesitation distribution which is equal or more tolerant than the limit of the hesitation efficiency for the corresponding problem.

As a default doubt distribution, one uses in the sequel the one of Figure 3a, as it fits SVM philosophy.

4.3. Extension to other classifiers

The method can also fit any binary classifier in which the distance to the separating hyperplane is known. It is nevertheless harder to determine a border between the hesitation and the two classes without any margin. One can use statistical analysis to set a Gaussian definition (Figure 3d) of the hesitation. It is not straightforward to settle the parameters of such a Gaussian model, as the aim is to have an empty margin rather than filled with items. Moreover, as the distance between the classes is not supposed to follow the trivial case of a Mahalanobis distance, the Gaussian distribution is not a priori well-suited for the hesitation modeling.

5. Applications

In this section, we illustrate the two advantages of our method on examples, in order to prove its efficiency. To have meaningful results, we lead various experiments on the same protocol:

- We used LIBSVM [11], which is a complete and efficient C/C++ library for SVM.
- C-SVM with RBF kernel is used in all runs.
- Accuracy has to be defined in a manner that is coherent with decisions from our method and the voting procedure (in order to allow comparisons). In that purpose, we use the *Pignistic Transform* [7], which transforms a belief function onto a probability function, on which an *argmax* decision is made. Then both methods provide decision on the set of classes *C* instead of its powerset, on which accuracy is to be computed. The *Pignistic Transform* is defined as:

$$\text{BetP}(h) = \frac{1}{1 - m(\emptyset)} \sum_{h \in A,\, A \subset \Omega} \frac{m(A)}{|A|} \quad \forall h \in \Omega$$

This transform corresponds to sharing the hesitation between the implied hypotheses, and normalizing the whole by the conflictive mass. As BetP does not lead to any interpretation of the conflict (which has been suppressed), it can be compared to the limit of the efficiency of the hesitation modeling with respect to the problem (in which no conflict occurs).

- In order to show our method allows the operator to deal with undetermined items, we illustrate the following strategy: Let us consider that *doubtful items* are dealt by choosing the most reliable class, but *uncertain items* are rejected in order to be reused in a next-coming retraining. Such a pattern corresponds to associate uncertainty (in terms of classification) to conflict (in terms of data fusion) and doubt to hesitation (in order to process it via BetP): Let m_{final} be the belief function on which the classification is made. If $m_{final}(\emptyset) = max_{2^\Omega}(m_{final}(.))$, then, the item is uncertain and discarded, else, the item is classified via the result of $argmax_{2^\Omega}(\text{BetP}(m_{final}))$. Then, we define:

$$Acc_{Sup} = \frac{\text{Number Of Well Classified}}{\text{Total Number} - \text{Number Of Rejected}}$$

$$Acc_{Inf} = \frac{\text{Number Of Well Classified}}{\text{Total Number}}$$

which means that Acc_{Sup} does not consider the rejected item, whereas Acc_{Inf} considers them as systematically false. As a consequence, one defines the rejection rate:

$$R = 1 - \frac{Acc_{Inf}}{Acc_{Sup}}$$

- To evaluate the improvement of our method, one considers the rate of avoided mistakes. *AvMis*, the percentage of avoided mistakes is defined as:

$$AvMis = \frac{\text{Number Of Mistake Avoided}}{\text{Original Number Of Mistake}}$$

$$= \frac{\text{Original Rate Of Mistake-New Rate Of Mistake}}{\text{Original Rate Of Mistake}}$$

$$= \frac{\text{1-Original Accuracy-1 + New Accuracy}}{\text{1-Original Accuracy}}$$

$$= \frac{\text{New Accuracy-Original Accuracy}}{\text{1-Original Accuracy}}$$

- When comparisons with state-of-the-art are needed, we use the classical *1vs1* combination scheme. It is implemented in LIBSVM [11] by a trivial voting procedure and no peculiar strategy for ties. In a belief-based implementation, each classifier $K^{i,\ j}$, which separates class C^i from class C^j provides an answer on three hypotheses: The doubt between the two classes, and a preference for any of the two classes. This answer has to be converted onto a belief function on 2^C. In that purpose, $K^{i,\ j}$ is considered as a C^i-C^j discarder: the belief in the doubt is assigned to *C*, the belief in C^i is assigned to $C\backslash C^j$ and the belief in C^j is assigned to $C\backslash C^i$. This trick simply allows a coherent combination of $K^{i,\,j}$ and $K^{g,\ h}$.
- When comparisons with state-of-the-art are needed, we do not try to optimize the SVM tuning, as our purpose is not really to have powerful discrimination, but to focus attention on the improvement of the fusion scheme, which is easier to notice on average classification rates than on accurate classifications.

In the next paragraphs comparable results are shown on a dedicated dataset and various other datasets.

5.1. Vowels dataset

The experiment is performed on the vowels dataset from [12], with a training dataset of 528 items, a testing dataset of 462 items, 10 attributes and 11 classes.

Using the classical *1vs1* voting procedure of binary SVMs and no posterior optimization, a classification rate of 55.8% is achieved on the test set. The distances to the hyperplane on these SVMs (which are normalized with respect to the margin size) are saved and reused in our fusion process based on BTs. Via BetP, the classification rate reaches 57.4%. It means that, 3.6% of the errors are simply avoided by using the same classifier but a smarter decision scheme.

If we consider a reject class which gathers all the conflictive uncertain items and a default doubt model (Figure 3a), Acc_{Sup} =58.1% and Acc_{Inf} =56.5%. If the doubt is restricted to the minimum (only on the hyperplane and nowhere else, as in Figure 3c), we have Acc_{Sup} =60.4% and Acc_{Inf} =52.8%. Between the limit of the doubt handling (BetP) and the limit of the rejection we defined, it is possible to tune the decision process to have any rejection rate, *R,* from 0% to 12.6%.

If we consider a *1vsAll* scheme, the performances are equivalent. Actually, they are slightly worse but the difference is too small to be significant; it can theoretically be interpreted as the direct consequence of a smaller number of sources to combine. (*N* vs. *N.(N-1)/2*).

5.2. Other datasets

In Table 2, various other datasets are presented, on which the same protocol is applied. 5-letter is a part of the dataset Letters [12], which is a dataset of 20.000 items corresponding to the 26 letters of the English alphabet. We made a reduced dataset on 5 letters, STUVX. Texture is another dataset from [12]. HCS is a dataset we made on Cued Speech hand shapes. It is based on the Hu invariants of binary hand image [13],[14].

Table 2: Description of the datasets

	Number of classes	Number of attributes	Training dataset size	Testing dataset size
Vowels	11	10	528	462
5-letter	5	16	1950	1952
Texture	7	19	210	2100
HCS	8	7	732	196

Results are shown in Table 3: ***voting*** gives the accuracy of the voting strategy, ***BetP*** gives the accuracy with the Pignistic Transform (i.e., with doubt shared), ***AvMis*** gives the rate of avoided mistakes thanks to BetP. ***Default Doubt*** and ***No Doubt*** correspond to the modeling of the doubt by using the models in Figure 3a and 3c respectively. R_{max} is the rejection rate corresponding to the absence of doubt. For 5-letter, we do not deal with undetermined cases as the original accuracy is really high: The improvement is too small to be represented with 3 meaningful digits, and thus, the comparison is worthless.

Table 3: Results in % for various datasets

	voting	BetP	**AvMis**	Default doubt		No doubt		R_{max}
				Acc_{Sup}	Acc_{Inf}	Acc_{Sup}	Acc_{Inf}	
vowels	55.8	57.4	**3.6**	58.1	56.5	60.4	52.8	12.6
5-letter	99.2	99.8	**73.3**					
texture	91.6	95.9	**51.2**	96.0	95.4	96.4	94.6	1.9
HCS	78.6	86.2	**35.7**	86.2	82.7	86.8	80.6	7.1

The results show an important rate of avoided mistakes thanks to our fusion method. This improvement is strongly dependent on the coherence classes. They also illustrate the ability to have a different processing for the various kinds of conflictive items (such as dealing with doubt by sharing it and keeping uncertain items for re-training).

6. Conclusion

We provide a simple method to combine the fusion methods of BTs with SVMs. The advantages are (1) optimizing the fusion of the sub-classifications, (2) dealing with undetermined cases due to uncertainty and doubt. Future works will focus on reject class for contradiction due to incoherence and on providing a complete decision scheme for undetermined items by extending the Pignistic Transform.

7. Acknowledgment

This work is the result of a cooperation supported by SIMILAR, European Network of Excellence (www.similar.cc).

The LIBSVM [11] add-on corresponding to these works has been implemented by Alexandra Urankar and is available on demand.

8. References

[1] B. Boser, I. Guyon, and V. Vapnik. “A training algorithm for optimal margin classifiers”, Fifth Annual Workshop on Computational Learning Theory, 1995.

[2] C. Cortes and V. Vapnik. “Support-vector network” *Machine Learning* 20, 273–297, 1995.

[3] R. Rifkin and A. Klautau. “In defense of one-vs-all classification”, *Journal of Machine Learning Research*, Vol.5, pp. 101-141, 2004.

[4] C.-W. Hsu and C.-J. Lin. “A comparison of methods for multi-class support vector machines”, *IEEE Transactions on Neural Networks*, Vol. 13, pp. 415-425, 2002.

[5] G. Shafer. *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[6] G. Shafer and P. P. Shenoy. “Probability propagation,” *Ann. Math. Art. Intel.*, vol. 2, pp. 327–352, 1990.

[7] P. Smets and R. Kennes. "The transferable belief model", *Artificial Intelligence*, 66(2): 191–234, 1994.

[8] L.A. Zadeh. *On the Validity of Dempster’s Rule of Combination of Evidence.* Berkeley, ERL, 1979.

[9] D. Dubois and H. Prade. "On the unicity of Dempster rule of combination". *Int. J. Intelligent System*, pp. 133–142, 1996.

[10] J.M. Nigro and M. Rombaut. “Idres: a rule-based system for driving situation recognition with uncertainty management”, *Information Fusion*, Vol. 4, dec. 2003.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: *a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[12]UCI machine learning database repository, http://www.ics.uci.edu/~mlearn/.

[13] A. Caplier, L. Bonnaud, S. Malassiotis and M. Strintzis. "Comparison of 2D and 3D analysis for automated Cued Speech gesture recognition", SPECOM, St Petersburg, Russia, 2004.

[14] T. Burger, A. Caplier and S. Mancini. "Cued Speech Hand Gesture Recognition Tool", EUSIPCO, Antalya, Turkey, 2005.