Dealing with precise and imprecise decisions with a Dempster-Shafer Theory based algorithm in the context of handwriting recognition

Thomas Burger, Yousri Kessentini, Thierry Paquet

Dealing with precise and imprecise decisions with a Dempster-Shafer theory based algorithm in the context of handwritten word recognition

Thomas Burger
Université Européenne de Bretagne,
Université de Bretagne-Sud, CNRS,
Lab-STICC, Vannes, France
thomas.burger@univ-ubs.fr

Yousri Kessentini, Thierry Paquet
Université de Rouen, Laboratoire LITIS EA 4108,
site du Madrillet, St Etienne du Rouvray, France
yousri.kessentini@univ-rouen.fr
thierry.paquet@univ-rouen.fr

Abstract

The classification process in handwriting recognition is designed to provide lists of results rather than single results, so that context models can be used as post-processing. Most of the time, the length of the list is determined once and for all the items to classify. Here, we present a method based on Dempster-Shafer theory that allows a different length list for each item, depending on the precision of the information involved in the decision process. As it is difficult to compare the results of such an algorithm to classical accuracy rates, we also propose a generic evaluation methodology. Finally, this algorithm is evaluated on Latin and Arabic handwritten isolated word datasets.

1. Introduction

In handwriting recognition, most of the classifiers are able to provide an ordered list of the potential classes, and not only a single preferred class. Hence, when evaluating an algorithm, not only a single accuracy rate is given, but several of them: The TOP 1 accuracy rate $Acc(1)$ considers items for which the classifier ranks the true class at the first position of the output ordered list, the TOP 2 accuracy rates $Acc(2)$ considers items for which the classifier ranks the true class among the first two positions of the list, and so on until the TOP N accuracy rates, noted $Acc(N)$.

The interests of this methodology are manifold: Firstly, it provides a richer description of the performances of an algorithm, which helps to discriminate among several algorithms indicating high performances. Secondly, it helps to point out the weaknesses of a classifier or the difficulties of a dataset: For example, if the accuracy rates peaked systematically at $N = 4$, it probably means that there exist some groups (or clusters) of less than four similar classes among which discrimination is difficult. Consequently, a hierarchical classification [1] would be interesting. But the main interest of this evaluation methodology is to quantify what a context model would bring: For instance, if $Acc(2)$ is far better than $Acc(1)$ on a word recognition application, a grammatical model that discriminates among the two best propositions would improve the scores. Then, it is best to set the algorithm so that lists of two elements are given. In the sequel, the size of the list provided by the classifier, generically noted N, is called the **cardinality of a decision**.

The main drawback of this methodology roots in the choice of N. Once this value is tuned, all the items will be assessed a decision with the same cardinality N, regardless to the difficulty of the classification task. Then, it does not take into account that some items are difficult to recognize (then, an imprecise decision with a great cardinality is suited), whereas, somes are trivial (then, a precise decision of cardinality 1 is suited). In this article, we adress the possibility to adapt the cardinality of the decision to each item to classify.

To the best of our knowledge, no such strategies exist in handwriting recognition field. Nevertheless, many works have been proposed to improve the reliability of the recognition systems and to find out how trustworthy is the output of the classifier. Such methods are known as rejection strategies. For such an aim, rejection mechanisms are usually used to reject word hypotheses according to an established threshold [2, 15, 13]. In [2] four different strategies to reject isolated handwritten street and city names are described. They are based on normalized likelihoods and the estimation of posterior probabilities. In [15] authors present several confidence measures and a neural network to either accept or reject word hypothesis lists for the recognition of cour-

tesy check amounts. In [13] varieties of novel rejection thresholds including global, class-dependent and hypothesis-dependent thresholds are proposed to improve the reliability in recognizing unconstrained handwritten words.

The first goal of this paper is to adapt to handwriting recognition a new approach for decision-making based on Dempster-Shafer theory (DST): When an item is processed, this approach allows determining without additional external information, whether a precise decision is trustful (with a cardinality of 1), or on the contrary, whether an imprecise decision is more adapted.

Contrarily to ours, classical algorithms provide the same cardinality for all the decisions. Hence, how is it possible to fairly compare the results of classical algorithms of the state of the art and ours ? The second goal of this paper is to propose an adapted evaluation methodology.

Section 2 is a brief presentation of the basis of DST. Section 3 presents our DST-based method to make precise/imprecise decisions. In section 4, we present a method to quantify the performances of any algorithm providing imprecise decisions, so that they can be compared to the performances of the classical algorithms. Finally, in section 5, we illustrate this methodology by evaluating the performance of our DST-based method.

2. Basis of Dempster-Shafer Theory

Dempster-Shafer Theory[17, 20] may be explained in many ways. Although not the most rigorous, the most intuitive way is to consider it is an application of the theory of random sets[1] [14, 19] in order to model knowledge inference problems. Its main interest is to provide different representation for the imprecision (the information is not focused enough) and the uncertainty (due to the randomness). This formalism is especially useful when the collected data is noisy or semi-reliable.

Formally, let $\Omega = \{\omega_1, ..., \omega_K\}$ be a finite set called the **frame** or the **state-space** which is made of exclusive and exhaustive hypotheses. A **mass function** m is defined on the powerset of Ω, noted $\mathcal{P}(\Omega)$ and it maps onto $[0, 1]$ so that $\sum_{A \subseteq \Omega} m(A) = 1$ and $m(\emptyset) = 0$. Then, a mass function is roughly a probability function defined on $\mathcal{P}(\Omega)$ rather than on Ω. Of course, it provides a richer description, as the support of the function is greater: If $|\Omega|$ is the cardinality of Ω, then $\mathcal{P}(\Omega)$ contains $2^{|\Omega|}$ elements. The uncertainty is modeled by the fact that two outcomes ω_i and ω_j are given some mass, i.e. $m(\omega_i) > 0$ and $m(\omega_j) > 0$, in a manner similar to with probability. On the contrary, the imprecision is modeled by the fact, that it is possible to associate some mass to $\{\omega_i, \omega_j\}$, without being precise enough to promote either ω_i or ω_j. Then, we have $m(\{\omega_i, \omega_j\}) > 0$.

Here are some additional vocabulary: A subset $F \subseteq \Omega$ such that $m(F) > 0$ is called a **focal element** of m. If $\max_i(|F_i|) = k$, then, the mass function is said to be k**-additive** [8]. In other words, a k-additive mass function has at minimum one focal elements of cardinality k, and no focal elements of cardinality $> k$. If the focal elements of a mass function m are nested, then, m is said to be **consonant**. We do not detail here all the operations defined in the DST and interested readers should refer to [17, 20].

Finally, probabilistic modeling and Dempster-Shafer modeling are two different ways to consider a problem. The second looks richer, but, (1) it has a computation cost, and (2) from a decision point of view, the insufficient reason principle [12] states that imprecision should be converted into uncertainty. Thus, there is no "best" theory, and the choice of one rather than the other mainly depends on the problem. That is why, the literature is full of methods to convert a mass function into a probability [3] and conversely [6, 21], so that it is possible to move from a model to the other one. In [23] a method is also described to derive a mass function from a set of classifiers that do not provide any probabilistic output. Thus, from now, it is safe to consider that it is possible to derive a mass function from most of the classical classifiers used in handwriting recognition. Moreover, the description given by the mass function is richer than a probabilistic output, and a probabilistic output is richer than a simple ordered list [23]. Hence, a mass function is always able to encode all the information contained in any list-type output of a classifier, including the coonfidence values associated to any item of the list (see [10]).

3. The decision-making procedure

3.1. Principle of the algorithm

In this section, we present how to make decisions such as described in the introduction from the mass function derived from a classifier. We expect this procedure (1) to be able to focus on precise decision (cardinality of 1) when possible, while remaining imprecise otherwise, and (2) the maximum cardinality of the decision is controlled, otherwise too imprecise and consequently worthless decisions would occur.

According to random set theory, a mass function with Ω as frame can be seen as a probability distribution with $\mathcal{P}(\Omega)$ as support. Making a *maximum a posteriori* decision on this probability distribution is pos-

[1] It is a generalization of probability theory where outcomes can be non-empty sets of random size.

sible. Practically, it means that $2^{|\Omega|}$ different options can be considered during the decision process. Among these options $|\Omega|$ of them correspond to precise decisions, and the others are imprecise, as they correspond to non-singleton focal elements. Nonetheless, such a decision-making procedure may lead to unpleasant situations: If the cardinality of the decision is close to $|\Omega|$, it is practically so imprecise that it is irrelevant. This is why, a precise decision, based on a *maximum a posteriori* strategy on a classical probability distribution is most of the time preferred. An alternative would be to consider k-additive mass functions, with controlled k.

The pignistic transform [18] is a very popular method to convert a mass function onto a probability distribution. It is designed so that, after the application of the transform, a *maximum a posteriori* decision corresponds to a decision which is made according to a statistically winning strategy.

In [4], a generalization of the pignistic transform is presented, and some of its mathematical properties are studied in [5]. This transform is used in [1] to derive a two-level classification procedure, where the second level is optional, in order to recognize American Sign Language in videos. Here, we aim at using it to make imprecise decisions is the context of handwriting recognition. The interest of this generalization is to convert any mass function into a k-additive mass function, the value of k being controlled. The original pignistic transform is a particular case of this transform, as a probability distribution corresponds to a 1-additive mass function. Moreover, this generalization is designed according to the original pignistic transform, so that it respects the same requirements. Consequently, it can be used in a statistically winning strategy too.

Finally, by considering a correctly constructed k-additive mass function as a probability, it is possible to implement precise and imprecise decisions according to our expectation of a controlled imprecision.

3.2. Description of the transform

Let $m^{[Cl]}$ be the mass function obtained from the output of the classifier. Let k be the maximum autorized value for the cardinality of the decision. The purpose is to convert $m^{[Cl]}$ onto a k-additive mass function $m^{[k]}$, and to select the focal element F_* such that:

$$F_* = \arg\max_{F_i} \left(m^{[k]}(F_i) \right) \quad (1)$$

If $|F_*| = 1$, the decision is precise, and if $2 \leq |F_*| \leq k$ the decision is imprecise, but with a limited imprecision. Practically, $m^{[k]}$ is defined $\forall B \subseteq \Omega$ such that $|B| \leq k$ as:

$$m^{[k]}(B) = m^{[Cl]}(B) + \sum_{\substack{A \supset B, \\ A \subseteq \Omega, |A| > k}} \frac{m^{[Cl]}(A) \cdot |B|}{\mathcal{N}(|A|, k)} \quad (2)$$

$m^{[k]}(.) = 0$ otherwise, and where

$$\mathcal{N}(|A|, k) = \sum_{i=1}^{k} \binom{|A|}{i} \cdot i = \sum_{i=1}^{k} \frac{|A|!}{(i-1)!(|A|-i)!}$$

represents the number of subsets of A of cardinality at most k, each of them being "weighted" by its cardinality. Intuitively, the mass $m(A)$ associated with any focal element A of cardinality $|A| > k$ is divided into $\mathcal{N}(|A|, k)$ equal parts, and these parts are redistributed to the focal elements of cardinality $\leq k$ in a manner proportional to their cardinality.

3.3. Warning

In pattern classification [7] it is possible to find an item which does not correspond to any of the classes. Then, an interesting issue is to discard it automatically by defining a reject class. In lexicon-driven handwriting approach, assuming that all words are present in the lexicon, the notion of rejection does not refer to a reject class. Instead, it refers to the rejection of the classifier decision. This rejection may occur as there is not enough evidence to come to a unique decision since 1) more than one word hypothesis appears adequate; 2) no word hypothesis appears adequate.

In fact, the precise/imprecise decision process presented in this paper is not a reject class implementation, nor it is a procedure to reject the decision-making. It is obviously possible to degenerate our procedure to implement a rejection of the decision process, by rejecting the cases where a too imprecise decision is given, but this strategy implies a loss of information. Moreover, the generalization of the pignistic transform is not designed to allow the tuning of the proportion of imprecise decisions, as for classical decision rejection, but to tune the maximum amount of imprecision. Then, the state of mind is completely different and comparisons would be abusive. This is why, in the sequel, we do not position this algorithm with respect to reject classification or with rejection of decision.

4. Evaluation methodology

The proposed strategy is of course more flexible. Nonetheless, before using it, we need to make sure that it will not lead to lower performances. Hence, we need to quantify its performances so that they can be compared to the classical ones of the state of the art, especially the $Acc(1), Acc(2), \ldots Acc(N)$ values. The aim

of this section is the definition of such an evaluation methodology.

The central idea is the following: In classical algorithms, N is both the maximum cardinality for each decision and the mean cardinality of all the decisions on a datasets (as they have all the same cardinality). In our case, k represents the maximum cardinality for each decision, but not the mean cardinality of all the decisions. The idea is to compute the mean cardinality, and to consider all the performances with respect to this value.

Prior to any rigorous definition, let us consider a toy example where 100 items are classified. For 60 of them, a decision of cardinality 1 is made, and a decision of cardinality 2 is made for the remaining 40 of them. Thus, the **mean cardinality** of the decision process is $\frac{60+2\times 40}{100} = 1.4$. Obviously, this value belongs to $\mathbb{Q}$, contrarily to the classical cases, as N belongs to $\mathbb{N}$. Thus, we note it Q.

More formally, if T is the size of the dataset and if $\alpha_j,\ j > 0$ represents the number of items for which a decision of cardinality j is made, then, the mean cardinality of the decision is:

$$Q = \frac{\sum_j j \cdot \alpha_j}{T}$$

Then, the next step is to define the **Rational Rank Accuracy** rate $Acc(Q)$, i.e. an accuracy rate for a fictive rank Q (this rank being a rational number). This accuracy must be coherent with the classical $Acc(N)$, which represents the accuracy where decisions have a cardinality of N. By definition, if $\delta_i^N = 1$ when the i-th element of the dataset is ranked in the first N elements of the output list of classifier (and $\delta_i^N = 0$ otherwise), then,

$$Acc(N) = \frac{\sum_{i=1}^{T} \delta_i^N}{T} = \frac{\sum_{i=1}^{T} \delta_i^N \times N}{T \times N} = \frac{\sum_{i=1}^{T} \delta_i^N \times N}{\sum_{i=1}^{T} \delta_i^T \times N}$$

as $T = \sum_{i=1}^{T} \delta_i^T$. Moreover N corresponds to the cardinality of the decision for item i. Let L_i, be the list of the classifier, we have $N = |L_i|$. Hence, we provide a more general definition of the accuracy:

$$Acc\left(\overline{|L_i|}\right) = \frac{\sum_{i=1}^{T} \delta_i^{|L_i|} \times |L_i|}{\sum_{i=1}^{T} \delta_i^T \times |L_i|}$$

where $\overline{|L_i|}$ is the mean of the $|L_i|$'s, or, in other words, Q. Let us expand this expression so that elements with lists of same size are grouped:

$$Acc(Q) = \frac{\sum_j \left[j \times \sum_{i/|L_i|=j} \delta_i^j \right]}{\sum_j \left[j \times \sum_{i/|L_i|=j} \delta_i^T \right]} = \frac{\sum_j j \cdot \beta_j}{\sum_j j \cdot \alpha_j}$$

where β_j the number of items for which a decision of cardinality j is **correctly** made.

In our toy example, if among the 60 precise decisions, 50 are correct, and among the 40 imprecise decisions, 30 are correct, then the accuracy is:

$$Acc(1.4) = \frac{50 \times 1 + 30 \times 2}{60 \times 1 + 40 \times 2} = \frac{110}{140} = 78.6\%$$

It is very simple to interpret $Acc(Q)$. Let us note $\lfloor Q \rfloor$ the integer part of Q and $\lceil Q \rceil = \lfloor Q \rfloor + 1$. $Acc(\lfloor Q \rfloor)$ and $Acc(\lceil Q \rceil)$ are classical accuracy rates that can be computed for any classical algorithm. From them, it is possible to compute $iAcc(Q)$, a linear interpolation of $Acc(\lfloor Q \rfloor)$ and $Acc(\lceil Q \rceil)$ for the value Q. This latter serves as a reference value for the $Acc(Q)$ of the algorithm to evaluate. For instance, if the reference algorithm has $Acc(1) = 80\%$ and $Acc(2) = 90\%$, then, $Acc(1.7)$ must be compared to $iAcc(1.7) = 87\%$.

In the next section, we aim at evaluating our algorithm. To do so, we use the same setting (same dataset, same classifier, same learning, and so on) with a classical decision making process and ours. Then, we compare the $Acc(Q)$ of our algorithm with the $iAcc(Q)$ for the reference algorithm. If the interpolated value is greater, our algorithm is not efficient, whereas if the interpolated value is smaller, it is efficient.

To achieve a detailed comparison, we also need to consider **Partial Accuracy** rate of rank $j, \forall j \leq k$, noted $pAcc(j)$. It simply corresponds to the rate β_j/α_j, i.e. the accuracy rates computed on the restricted set of items on which a decision of cardinality j is made. In case of a random choice for the cardinality of the decision, $pAcc(j)$ is roughly equivalent to the classical accurcay $Acc(j)$ computed on all the dataset. In our case, the method is designed on purpose, so that the precise decisions are made only when they are reliable. Hence, we expect the $pAcc(j)$'s to be higher for the decisions small cardinality. This point is interesting, as it means precision and robustness are linked.

5. Application to multiscript handwriting recognition

To evaluate this decision process, experiments are conducted on three publicly available databases: IFN/ENIT benchmark database of Arabic words and RIMES and IRONOFF databases for Latin words. The IFN/ENIT [16] contains 32,492 handwritten words (Arabic symbols) of 946 Tunisian town/villages names written by 411 different writers. Four different sets (a, b, c, d) are used for training and 3000 word images from set (e) for testing. The RIMES database [9] is composed of isolated handwritten word snippets extracted

from handwritten letters (latin symbols). In our experiments, 36000 snippets of words are used to train the different HMM classifiers and 3000 words are used in the test. IRONOFF [22] is both an on-line and off-line dataset. The subdataset IRONOFF-Chèque only contains a small lexicon of roughly 30 words used on French checks (numbers, currencies, etc.). 7956 words are used for the training and 3987 are used for tests.

Datasets	$Acc(1)$	$Acc(2)$	$Acc(3)$	$Acc(4)$
RIMES	54.10	66.40	72.13	75.87
IFN/ENIT	73.60	79.77	82.83	84.60
IRONOFF	85.65	91.51	93.84	95.55

Table 1. TOP N accuracy rates for the RIMES, IFN/ENIT and IRONOFF datasets.

The absolute accuracy of the classifier being not an issue here, a simple protocol is applied: A HMM classifier based on the upper contour description of the image of the word is used to derive a posterior probabilities for the word to recognize to belong to each class [11]. As it clearly appears in Table 1 where the TOP 1 to TOP 4 performances are given, we have chosen three datasets of heterogeneous difficulty with respect to the classifier used : RIMES is rather difficult, IFN/ENIT is of intermediate difficulty and IRONOFF is the simplest.

After the classification step, the posterior probability distribution of the HMM classifier is converted into a consonant mass function with the transform from [6]. Roughly, the probabilities are decreasingly sorted, and masses proportional to the pairwise differences are associated to the corresponding focal elements. Then, the generalization of pignistic transform is applied, with different values of $k \in \{2, 3, 4\}$. In fact, for $k = 1$ the results are those of the $Acc(1)$ in Table 1. The performances are computed according to the previous section, and are presented in Table 2. They are based on the values T, α_j, β_j summarized in Table 3.

In Table 2, we consider $\Delta = Acc(Q) - iAcc(Q)$. Positive values ($\geq 0.4$) shows an improvement, whereas values close to zero ($|\Delta| < 0.1$) shows the two decision-making algorithms are equivalent. There are no negative values indicating our method is less efficient than the classical one. Hence, according to the Rational Rank Accuracy rates, the improvement of the result is significant for the RIMES and IRONOFF datasets: even 1 point of improvement is satisfying as (1) it must be compared to the remaining proportion of error mistake (it is easier to improve from 53 to 54% than from 97 to 98%), and as (2) the classification procedure as well as the learning phase are the same (the improvements only rely on the decision process). On the other hand, no improvement/lessening appears on the IFN/ENIT dataset.

Datasets	RIMES	IFN/ENIT	IRONOFF
k = **2**			
Q	1.778	1.705	1.675
$Acc(Q)$ (%)	64.96	78.00	90.29
$iAcc(Q)$ (%)	63.67	77.95	89.60
Δ	**+1.29**	**+0.06**	**+0.69**
$pAcc(1)$ (%)	70.12	85.67	96.05
$pAcc(1)$ (%)	64.22	76.40	88.91
k = **3**			
Q	2.235	2.075	2.068
$Acc(Q)$ (%)	69.22	79.95	92.11
$iAcc(Q)$ (%)	67.75	80.00	91.67
Δ	**+1.47**	**-0.04**	**+0.45**
$pAcc(1)$ (%)	70.95	85.91	96.00
$pAcc(2)$ (%)	69.48	80.62	91.07
$pAcc(3)$ (%)	68.81	77.97	91.53
k = **4**			
Q	2.72	2.467	2.536
$Acc(Q)$ (%)	71.76	81.19	94.06
$iAcc(Q)$ (%)	70.53	81.20	92.76
Δ	**+1.23**	**-0.01**	**+1.31**
$pAcc(1)$ (%)	74.21	87.10	96.76
$pAcc(2)$ (%)	72.45	83.21	93.19
$pAcc(3)$ (%)	71.76	80.09	93.61
$pAcc(4)$ (%)	71.14	79.35	94.02

Table 2. Comparison between our algorithm and the classical approach.

According to the Partial Accuracy rates, the more the results are precise, the more robust ($pAcc(N) \geq pAcc(N+1)$). This is noticeable for RIMES and IFN/ENIT datasets, but less obvious for IRONOFF. When compared to the classical TOP N of Table 1, it is interesting to see that precise decisions are trustful. This is an additional major interest of our method, that can be illustrated on an example: Let us consider a sentence of 20 words, and a classical decision making algorithm tuned to provide decision of cardinality 4. The context model must discriminate among $4^{20} = 1.1 \times 10^{12}$ possible sentences. On the contrary, using our algorithm, the number possible sentences is far lower (this reduction is quantified by $k - Q$), but mostly, the few words with precise decision are rather trustful (the $pAcc(1)$'s are really high), which implies they act like fix points which drastically reduce the number of combinations, and thus, the number of possible sentences.

Finally, the improvement is the most noticeable on RIMES dataset, i.e. the most difficult one. This fact cor-

Datasets	T	$k=2, \alpha_1, \beta_1, \alpha_2, \beta_2$	$k=3, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3$	$k=4, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \alpha_4, \beta_4$
RIMES	3000	666, 467, 2334, 1499	661, 469, 973, 676, 1366, 940	539, 400, 686, 497, 850, 610, 925, 658
IFN/ENIT	3000	886, 759, 2114, 1615	887, 762, 1001, 807, 1112, 867	775, 675, 786, 654, 703, 563, 736, 584
IRONOFF	3979	1292, 1241, 2687, 2389	1299, 1247, 1109, 1010, 1571, 1438	1080, 1045, 808, 753, 971, 909, 1120, 1053

Table 3. Nessary values for the computations of the results of Table 2.

responds to the general remark of [23], indicating that DST-based method are particularly efficient on difficult handwriting recognition problems.

6. Conclusion

We have presented a new decision making algorithm in the context of handwriting recognition. We stressed its interest and developed a methodology to compare it to the state of the art. We experimentally proved its slight superiority on three different datasets of various difficulties and scripts. Beyond the accuracy rates, we have justified its use in a complete handwriting recognition setup. Future works will focus on dealing with the imprecise decisions, by the use of context model.

References

[1] O. Aran, T. Burger, A. Caplier, and L. Akarun. A belief-based sequential fusion approach for fusing manual and non-manual signs. *Pattern Recognition*, 42(5):812–822, May 2009.

[2] A. Brakensiek, J. Rottland, and G. Rigoll. Confidence measures for an address reading system. *International Conference on Document Analysis and Recognition*, 1:294–298, 2003.

[3] T. Burger. Defining new approximations of belief function by means of dempster's combinations. In *Workshop on the Theory of Belief Functions*, 2010.

[4] T. Burger and A. Caplier. A generalization of the pignistic transform for partial bet. In *Proceedings of ECSQARU'2009, Verona, Italy*, pages 252–263, July.

[5] T. Burger and F. Cuzzolin. The barycenters of the k-additive dominating belief functions & the pignistic k-additive belief functions. In *Workshop on the Theory of Belief Functions*, 2010.

[6] D. Dubois, H. Prade, and P. Smets. New semantics for quantitative possibility theory. In S. Benferhat and P. Besnard, editors, *Proc. of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 2001)*, pages 410–421, Toulouse, France, 2001. Springer-Verlag.

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[8] M. Grabisch. K-order additive discrete fuzzy measures and their representation. *Fuzzy sets and systems*, 92:167–189, 1997.

[9] E. Grosicki, M. Carre, J. Brodin, and E. Geoffrois. Results of the rimes evaluation campaign for handwritten mail processing. *International Conference on Document Analysis and Recognition*, 0:941–945, 2009.

[10] Y. Kessentini, T. Burger, and T. Paquet. Evidential combination of multiple HMM classifiers for multi-script handwritting recognition. In *Proc. IPMU'10*, 2010.

[11] Y. Kessentini, T. Paquet, and A. B. Hamadou. Off-line handwritten word recognition using multi-stream hidden markov models. *Pattern Recognition Letters*, 30(1):60–70, 2010.

[12] J. M. Keynes. Fundamental ideas. *A Treatise on Probability, Ch. 4*, 1921.

[13] A. L. Koerich, R. Sabourin, and C. Y. Suen. Recognition and verification of unconstrained handwritten words. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1509–1522, 2005.

[14] H. T. Nguyen. On random sets and belief functions. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 105–116. 2008.

[15] G. Nikolai. Optimizing error-reject trade off in recognition systems. In *ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 1092–1096, Washington, DC, USA, 1997. IEEE Computer Society.

[16] M. Pechwitz, S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri. Ifn/enit - database of handwritten arabic words. *Colloque International Francophone sur l'Ecrit et le Document*, pages 129–136, 2002.

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

[18] P. Smets. Constructing the pignistic probability function in a context of uncertainty. In M. Henrion, R. Shachter, L. Kanal, and J. Lemmer, editors, *Uncertainty in Artificial Intelligence, 5*, pages 29–39. Elsevier Science Publishers, 1990.

[19] P. Smets. The transferable belief model and random sets. *International Journal of Intelligent Systems*, pages 37–46, 1992.

[20] P. Smets. The transferable belief model. *Artif. Intell.*, 66(2):191–234, 1994.

[21] J. J. Sudano. Inverse pignistic probability transforms. In *Proc. 5th Inter. Conf. on Information Fusion*, pages 763–768, 2002.

[22] C. Viard-Gaudin, P. M. Lallican, P. Binter, and S. Knerr. The ireste on/off (ironoff) dual handwriting database. *International Conference on Document Analysis and Recognition*, 0:455–458, 1999.

[23] L. Xu, A. Krzyzak, and C. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Tran. Syst. Man Cybern.*, (3), 1992.