An Overview of Classifier Fusion Methods

Dymitr Ruta and Bogdan Gabrys

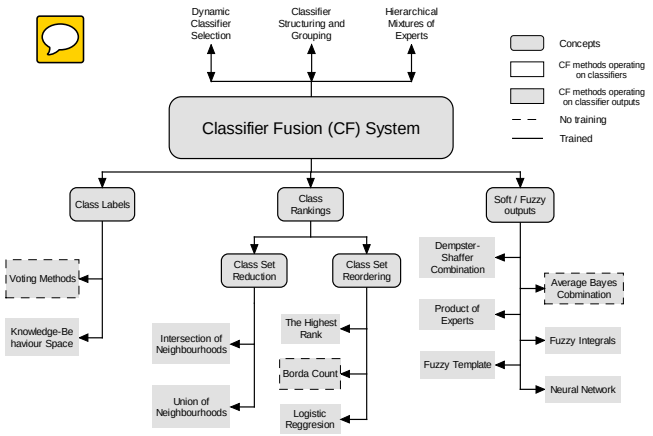
A number of classifier fusion methods have been recently developed opening an alternative approach leading to a potential improvement in the classification performance. As there is little theory of information fusion itself, currently we are faced with different methods designed for different problems and producing different results. This paper gives an overview of classifier fusion methods and attempts to identify new trends that may dominate this area of research in future. A taxonomy of fusion methods trying to bring some order into the existing "pudding of diversities" is also provided.

1. INTRODUCTION

The objective of all decision support systems (DSS) is to create a model, which given a minimum amount of input data/information, is able to produce correct decisions. Quite often, especially in safety critical systems, the correctness of the decisions taken is of crucial importance. In such cases the minimum information constraint is not that important as long as the derivation of the final decision is obtained in a reasonable time. According to one approach, the progress of DSS should be based on continuous development of existing methods as well as discovering new ones. Another approach suggests that as the limits of the existing individual method are approached and it is hard to develop a better one, the solution of the problem might be just to combine existing well performing methods, hoping that better results will be achieved. Such fusion of information seems to be worth applying in terms of uncertainty reduction. Each of individual methods produces some errors, not mentioning that the input information might be corrupted and incomplete. However, different methods performing on different data should produce different errors, and assuming that all individual methods perform well, combination of such multiple experts should reduce overall classification error and as a consequence emphasise correct outputs. Information fusion techniques have been intensively investigated in recent years and their applicability for classification domain has been widely tested [1]-[14].

The problem arouse naturally as a need of improvement of classification rates obtained from individual classifiers. Fusion of data/information can be carried out on three levels of abstraction closely connected with the flow of the classification process: *data level fusion, feature level fusion, and classifier fusion* [15]. There is little theory about the first two levels of information fusion. However, there have been successful attempts to transform the numerical, interval and linguistic data into a single space of symmetric trapezoidal fuzzy numbers [14], [15], and some heuristic methods have been successfully used for feature level fusion [15]. A number of methods have been developed for classifier fusion also referred to as decision fusion or mixture of experts. Essentially, there are two general groups of classifier fusion techniques. The methods subjectively associated with the first group generally operate on classifiers and put an emphasis on a development of the classifier structure. They do not do anything with classifiers outputs until combination process finds single best classifier or a selected group of classifiers and only then their outputs are taken as a final decision or for further processing [2], [9], [10]. Another group of methods operate mainly on classifiers outputs, and effectively the combination of classifiers outputs is calculated [1], [3]-[8], [11]-[15]. The methods operating on classifiers outputs can be further divided according to the type of the output produced by individual classifiers. A diagrammatic representation of the proposed taxonomy of classifier fusion methods is shown in Figure 1.

Fig 1. Classifier Fusion Methods



From the three possible types of outputs generated by individual classifiers the crisp labels offer the minimum amount of input information for fusion methods, as no information about potential alternatives is available. Some additional useful information can be gained from classification methods generating outputs in a form of class rankings.

However, fusion methods operating on classifiers with soft/fuzzy outputs can be expected to produce the greatest improvement in classification performance. The remaining of this paper is organised as follows. In section II methods operating on classifiers are briefly presented. The following three sections provide an overview of the classifier fusion methods operating on single class labels, class rankings and fuzzy measures respectively. Finally conclusions and suggestions for future work are presented.

2. METHODS OPERATING ON CLASSIFIERS

As mentioned in the introduction, a number of fusion methods operate on the classifiers rather than their outputs, trying to improve the classification rate by pushing classifiers into an optimised structure. Among these methods a dominant role is played by Dynamic Classifier Selection, which is sometimes referred to as an alternative approach to the classifier fusion. The other two approaches reviewed in this section include classifier structuring and grouping and hierarchical mixture of experts.

2.1 Dynamic Classifier Selection

Dynamic Classifier Selection (DCS) methods reflect the tendency to extract a single best classifier instead of mixing many different classifiers. DCS attempts to determine a single classifier, which is the most likely to produce the correct classification label for an input sample [2], [10]. As a result only the output of the selected classifier is taken as a final decision. Dynamic classifier selection process includes a partitioning of the input samples. There are a number of methods for the partition forming, starting from the classifiers agreement on the top choices, up to the grouping of features of input samples.

An example of DCS method is recently developed DCS by Local Accuracy (DCS-LA). Having defined the set of partitions, the best classifier for each partition is locally selected. Considering final classification process, an unknown sample is assigned to a partition, and the output of the best classifier for that partition is taken as a final decision. The idea of using DCS-LA is to estimate each classifier's accuracy in a local region of feature space and then the final decision is taken as an output from the most locally accurate classifier.

Another approach assumes estimating a local regression model (i.e. logistic regression [2]) for each partition. After the model is estimated for each partition, a relevant decision combination function is selected dynamically for each test case.

All DCS methods strongly rely on training data and by choosing only locally best classifier they seem to lose some useful information available from other well performing local classifiers. However, applying the DCS method sequentially, excluding the best classifier each time, it is possible to obtain a very reliable ranking of classifiers and eventually also class rankings. Such an approach could be treated as a good pre-processing stage before other methods operating on class rankings are used.

2.2 Classifier Structuring and Grouping

Classifiers and their combination functions may be organized in many different ways [2]. The standard approach is to organise them in parallel and simultaneously and separately get their outputs as an input for a combination function or alternatively sequentially apply several combination functions. According to another strategy, a more reasonable approach is to organise all classifiers into groups and to apply different fusion methods for each group. In general, classifiers may be arranged in a multistage structure. At each stage different fusion methods should be applied for different groups of classifiers. Additionally, DCS methods could be used at some stage for selection of the best classifier in each group.

There are a lot of different design options, which are likely to be specific for a particular application. However, at each stage of grouping, a very important factor is the level of diversity of classifier types, training data and methods involved [17]. Any possible classification improvement may only be achieved if the total information uncertainty is reduced [16]. This in turn depends on the diversity of information supporting different classification methods.

On the other hand, the same goal can be achieved by reduction of errors produced by individual classifiers. Decision combination process tries to minimise the final classification error. As most combination functions work on the basis of increased importance of repetitive inputs, the greater the diversity of the errors produced by individual classifiers, the lower their impact on the final decision and effectively the lower final error. This rule can be applied for any kind of groupings and structuring that might be used in the multiple classifier system.

2.3 Hierarchical mixture of experts

Hierarchical mixture of experts (HME) is an example of the fusion method, which strength comes from classifiers structure. HME represents a supervised learning technique based on the divide-and-conquer principle, which is broadly used throughout computer science and applied mathematics [9]. The HME is conceptually organised in a tree-like structure of leaves. Each leave represents an individual expert network, which given the input vector x tries to solve local supervised learning problem. The outputs of the

elements of the same node are partitioned and combined by the gating network and the total output of the node is given as a convex combination. The expert networks are trained to increase the posterior probability according to Bayes rule and then a number of learning algorithms can be applied to tune the mixture model.

Recently the EM algorithm was developed for the HME architecture [9]. The tests on the robot dynamics problem showed a substantial improvement in comparison with the back-propagation neural network. The HME technique does not seem to be applicable for a large dimensional data, as increase of the complexity of the tree-like architecture and associated input space subdivision lead to the increased variance and numerical instability.

3. FUSING SINGLE CLASS LABELS

Classifiers producing crisp, single class labels (SCL) provide the least amount of useful information for the combination process. However, they are still well performing classifiers, which could be applied to a variety of real-life problems. If some training data are available, it is possible to upgrade the outputs of these classifiers to the group operating on class rankings or even fuzzy measures. There are a number of methods to achieve this goal, for instance by performing an empirical probability distribution over a set of training data. The two most representative methods for fusing SCL classifiers, namely generalised voting method, and Knowledge-Bahaviour Space method, are now presented.

3.1 Voting Methods

Voting strategies can be applied to a multiple classifier system assuming that each classifier gives a single class label as an output and no training data are available. There are a number of approaches to combination of such uncertain information units in order to obtain the best final decision. However, they all lead to the generalised voting definition. For convenience let the output of the classifiers form the decision vector d defined as $d=[d_1,d_2,...,d_n]^T$ where $d_i \in \{c_1,c_2,...,c_m,r\}$, c_i denotes the label of the i-th class and r the rejection of assigning the input sample to any class. Let binary characteristic function be defined as follows:

$$B_j(c_i)=\begin{cases}1 & \text{if} \quad d_j=c_i \\ 0 & \text{if} \quad d_j\neq c_i\end{cases}$$

Then the general voting routine can be defined as:

$$E(d)=\begin{cases}c_i & \text{if} \quad \underset{t\in\{1,..,m\}}{\forall}\sum_{j=1}^{n}B_j(c_t)\le\sum_{j=1}^{n}B_j(c_i)\ge\alpha\cdot m+k(d) \\ r & \text{otherwise}\end{cases}$$

where α is a parameter and $k(d)$ is a function that provides additional voting constraints. The most conservative voting rule is given if $k(d)=0$ and $\alpha=1$, meaning that the class is chosen when all classifiers produce the same output. This rule can be liberalised by lowering the parameter α. The case where $\alpha=0.5$ is commonly known as the majority vote. Function $k(d)$ is usually interpreted as a level of abjection to the most often selected class and refers mainly to the score of the second ranked class. This option allows to adjust the level of collision that is still acceptable for giving correct decision.

3.2 Behaviour-Knowledge Space Method

Most fusion methods assume independence of the decisions made by individual classifiers. This is in fact not necessarily true and Behaviour-Knowledge Space method (BKS) does not require this condition [12]. It provides a knowledge space by collecting the records of the decisions of all classifiers for each learned sample. If the decision fusion problem is defined as a mapping of K classifiers: $e_1,...,e_K$ into M classes: $c_1,...,c_M$, the method operates on the K - dimensional space. Each dimension corresponds to an individual classifier, which can produce $M+1$ crisp decisions, M class labels and one rejection decision. A unit of BKS is an intersection of decisions of every single classifier. Each BKS unit contains three types of data: the total number of incoming samples: $T_{e_1,...,e_K}$, the best representative class: $R_{e_1,...,e_K}$, and the total number of incoming samples for each class: $n_{e_1,...,e_K}(m)$. In the first stage of BKS method the training data are extensively exploited to build the BKS. Then the final classification decision for an input sample is derived in the focal unit where the balance is estimated between the current classifiers decisions and the recorded behaviour information as shown in the following rule:

$$E(x)=\begin{cases}R_{e_1,...,e_K} & \text{if} \quad T_{e_1,...,e_K}>0\cap\dfrac{n_{e_1,...,e_K}(R_{e_1,...,e_K})}{T_{e_1,...,e_K}}\ge\lambda \\ \text{rejection} & \text{otherwise}\end{cases}$$

where λ is a threshold controlling the reliability of the final decision. The model tuning process should include automatic finding of the threshold λ.

4. CLASS RANKING BASED TECHNIQUES

Among the fusion methods operating on class rankings as the outputs from multiple classifiers, two main approaches are worth mentioning. The first is based on a class set reduction and its objective is to reduce the set of considered classes to as small a number as possible but ensuring that the correct class is still represented in the reduced set. Another

approach aims at a class set reordering in order to obtain the true class ranked as close to the top as possible. Interestingly, both approaches may be applied to the same problem, so that the set of the classes is first reduced and then reordered.

4.1 Class Set Reduction Methods

At an early stage of combining multiple classifiers, it is reasonable to try to reduce the set of possible classes. Two main criteria have to be taken into consideration while reducing the class set: the size of the set of classes and the probability of containing the true class in the reduced set of classes. The class set reduction (CSR) methods try to find the trade-off between the minimising of the class set and maximising of the probability of inclusion of the true class. Two different approaches are dominant in this type of analysis.

4.1.1 Intersection of Neighbourhoods

One CSR method computes an intersection of large neighbourhoods trying to find the threshold rank of a class, below which classes are removed [2]. To achieve this, firstly the neighbourhoods of all classifiers are determined by the ranks of true classes for the worst case in the training data set. The lowest rank ever given by any of the classifier is taken as the threshold and only the classes that are ranked above are used for further processing. This method also recognises redundant classifiers as the ones for which the thresholds are equal to the size of the class set. Intersection approach should only be applied to the classifiers with moderate worst-case performance.

4.1.2 Union of Neighbourhoods

Another method provides a union of small neighbourhoods taken from each classifier [2]. The threshold for each classifier is calculated as the maximum (worst) of the minimums (best) of ranks of true classes over the training data set. The redundant classifier can be easily determined, as its threshold equals to zero meaning that its output is always incorrect. This method is suitable for the classifiers with different types of inputs.

4.2 Class Set Reordering Methods

Class Set Reordering (CSRR) methods try to improve overall rank of the true class. The CSRR method is considered to be successful if it ranks the true class higher than any individual classifier. Three most commonly used techniques are here presented [2].

4.2.1 The Highest Rank Method

Assuming that each classifier produces a ranking list of classes, it is possible to make groups of rankings referring to each class. According to the Highest Rank (HR) method [2], the minimum from these groups of rankings is assigned to each class and then classes are sorted according to the new ranks. If an individual class has to be determined as a final decision, the one from the top of the reordered ranking is chosen. This method is particularly dedicated to cases with a large number of classes and few classifiers. An advantage of the HR method is that it utilises the strength of every single classifier, which means that as long as there is at least one classifier that performs well, the true class should always be near the top of the final ranking. The weakness is that combined ranking may have many ties, which have to be resolved by additional criteria.

4.2.2 The Borda Count Method

Borda Count (BC) is an example of *group consensus functions*, defined as a mapping from a set of individual rankings to a combined ranking leading to the most relevant decision [1], [2]. For a particular class c_k Borda Count $B(c_k)$ is defined as a sum of the number of classes ranked below class c_k by each classifier. The magnitude of the BC reflects the level of agreement that the input pattern belongs to the considered class. To a certain degree the BC can be treated as a generalization of the majority-voting rule and for a case of two classes problem it is exactly reduced to the majority vote.

The idea behind the BC method is based on the assumption of additive independence among the contributing classifiers. The method ignores the redundant classifiers, which reinforce errors made by other classifiers. The Borda Count method is easy to implement and does not require any training. Weak point of this technique is that it treats all classifiers equally and does not take into account individual classifiers capabilities. This disadvantage can be reduced to a certain degree by applying weights and calculation of BC as a weighted sum of a number of classes. The weights can be different for every classifier, which in turn requires additional training.

4.2.3 Logistic Regression

The Borda Count method does not recognise the quality of individual classifiers outputs. An improvement can be achieved by assigning the weights to each classifier reflecting their importance in a multiple decision system and performing so-called logistic regression [2]. An important thing at this stage is to distinguish the classification correctness and classifiers correlation, treating them as separate problems to be modelled. If we assume that the responses: $(x_1, x_2, ..., x_m)$ from m classifiers are highest for the classes ranked at the top of the ranking it is possible to use the logistic response function:

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)}{1 + \exp(\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)}$$

where $\alpha, \beta_1, \beta_2, ..., \beta_m$ are parameters, which are constant. The output of the transformation:

$$L(x) = \log \frac{\pi(x)}{1 - \pi(x)} = (\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m)$$

is referred to as *logit* and provides new value according to which combined rankings are created. The model parameters can be estimated using data fitting methods based on maximum likelihood. The logits or $\pi(x)$ values can be additionally treated as confidence measures. It is possible to determine a threshold value so that classes with confidence value below the threshold are rejected.

5. SOFT-OUTPUT CLASSIFIER FUSION METHODS

The largest group of classifier fusion methods operate on classifiers which produce so-called soft outputs. The outputs are the real values in the range $[0,1]$. These values are generally referred to as fuzzy measures, which cover all known measures of evidence: probability, possibility, necessity, belief and plausibility [16]. All these measures are used to describe different dimensions of information uncertainty. Effectively, the fusion methods in this group try to reduce the level of uncertainty maximising suitable measures of evidence.

5.1 Bayesian Fusion Methods

The Bayesian methods can be applied to the classifier fusion under the condition that the outputs of the classifier are expressed in posterior probabilities. Effectively combination of given likelihoods is also a probability of the same type, which is expected to be higher than the probability of the best individual classifier for the correct class. Two basic Bayesian fusion methods are introduced. The first one named *Bayes Average* is a simple average of posterior probabilities. The second method uses Bayesian methodology to provide a belief measure associated with each classifier output and eventually integrates all single beliefs resulting in a combined final belief.

5.1.1 Simple Bayes Average

If the outputs of the multiple classifier system are given as posterior probabilities that an input sample x comes from a particular class $C_i, i = 1,..,m$: $P(x \in C_i / x)$, it is possible to calculate an average posterior probability taken from all classifiers:

$$P_E(x \in C_i / x) = \frac{1}{K} \sum_{k=1}^{K} P_K(x \in C_i / x)$$

where $i = 1,..,m$. Such a Bayes decision, based on the newly estimated posterior probabilities is called an average Bayes classifier. This approach can be applied for the Bayes classifiers. For other classifiers there is a number of methods to estimate posterior probability. As an example for the k – NN classifier the transformation is given in the following form:

$$P_k(x \in C_i / x) = \frac{k_i}{k_{nn}}$$

where k_i denotes the number of prototype samples from class C_i out of all k_{nn} nearest prototype samples. The quality of the Bayes average classifier depends on how the posterior probabilities are estimated and the diversity of used classifiers.

5.1.2 Bayes Belief Integration

The approach mentioned above treats equally all the classifiers and does not explicitly consider different errors produced by each of them. These errors can be comprehensively described by means of confusion matrix given by:

$$PT_k = \begin{pmatrix} n_{11}^{(k)} & ... & n_{1(M+1)}^{(k)} \\ ... & ... & ... \\ n_{M1}^{(k)} & ... & n_{M(M+1)}^{(k)} \end{pmatrix}$$

where rows correspond to classes: $c_1, ..., c_M$ from which the input sample was drawn from and columns denote the classes to which the input sample was assigned by the classifier e_k. The values $n_{i,j}^{(k)}$ express how many input samples coming from class c_i were assigned to class c_j. On the basis of the confusion matrix PT_k it is possible to build the belief measure of correct assignment as given by:

$$Bel(x \in c_i / e_k(x)) = P(x \in c_i / e_k(x) = j_k)$$

where $i = 1,..., M$; $j = 1,..., M+1$ and

$$P(x \in c_i / e_k(x) = j) = \frac{n_{ij}^{(k)}}{\sum_{i=1}^{M} n_{ij}^{(k)}}$$

Having defined such a belief measure for each classifier we can combine them in order to create new belief measure of the multiple classifier system as follows:

$$Bel(i) = P(x \in c_i) \frac{\prod_{k=1}^{K} P(x \in c_i / e_k(x) = j_k)}{\prod_{k=1}^{K} P(x \in c_i)}$$

The probabilities used in the above formula can be easily estimated from the confusion matrix. The class

with the highest combined belief measure: Bel(i) is chosen as a final classification decision. Alternatively selection of any class may be rejected if the combined belief is smaller than a specified threshold value.

5.2 Fuzzy Integrals

Fuzzy integrals aim at searching for the maximal agreement between the real possibilities relating to objective evidence and the expectation g which defines the level of importance of a subset of sources. The concept of fuzzy integrals arises from the λ-fuzzy measure g_λ developed by Sugeno. It generalises the probability by adding parameter λ to the additive probability measure with respect to disjoint objects of measure:

$$\underset{\substack{A,B\subset X\\ A\cap B=0}}{\forall} g(A\cup B)=g(A)+g(B)+\lambda g(A)g(B)$$

From the normalization $g(X)=1$ we can derive value λ by solving the equation:

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda g^i)$$

where g^i are fuzzy densities which could be chosen subjectively or estimated through a training process. Thus, knowing the fuzzy densities g^i, $i=1,...,n$, one can construct the fuzzy measure g_λ for the set A. The fuzzy measures g_λ are a subclass of belief (for $\lambda\geq 0$) and plausibility (for $\lambda\leq 0$) measures defined by Shafer.

5.2.1 Sugeno Fuzzy Integral

Sugeno fuzzy integral combines objective evidence of an hypothesis with the prior expectation of the importance of the evidence to the hypothesis. If we introduce a measurable space (X,Ω) and a function $h: X\rightarrow[0,1]$ then the fuzzy integral over set $A\subseteq\Omega$ of the function h with respect to a fuzzy measure g is defined by:

$$\int_A h(x)\circ g(\cdot)=\sup_{E\subseteq X}[\min(\min_{x\in E} h(x), g(A\cap E))]$$

Calculation of the Sugeno fuzzy integral is unexpectedly easy if we reorder elements of a set $X=\{x_1,...,x_n\}$ so that the condition: $h(x_1)\geq h(x_2)\geq ...\geq h(x_n)$ is met. Then a fuzzy integral e with respect to the fuzzy measure g over X can be computed by:

$$e=\max_{i=1}^{n}[\min(h(x_i), g(A_i))]$$

where $A_i=\{x_1,...,x_i\}$. Note that when g is the λ-fuzzy measure, the values of $g(A_i)$ can be computed recursively as:

$$\begin{aligned} g(A_1)&=g(\{x_1\})=g^1 \\ g(A_i)&=g^i+g(A_{i-1})+\lambda g^i g(A_{i-1}) \end{aligned} \qquad \text{for } 1<i\leq n$$

For pattern recognition applications, the function $h_k(x_i)$ can be treated as a partial evaluation of the degree of belonging of the object A to class k, given by the classifier associated with a group of features x_i. Sugeno integral can be successfully applied to any multi-sensor systems by fusing the classifiers outputs in order to provide more accurate classification rates.

5.2.2 Choquet Fuzzy Integral

Choquet fuzzy integral, developed by Sugeno and Murofushi, provides an extension to the Lebesgue integral which Sugeno integral do not cover. The definition of the Choquet integral refers to the Choquet functional presented in a different context. The assumptions are the same as for the Sugeno integral. The Choquet integral over set $A\subseteq\Omega$ of the function $h: X\rightarrow[0,1]$ with respect to a fuzzy measure g is defined by:

$$\int_A h(x)\circ g(\cdot)=\int_0^{+\infty} g(A_\alpha)d\alpha$$

where $A_\alpha=\{x\mid h(x)>\alpha\}$. Calculation of the Choquet fuzzy integral is similar to the numerical methods of integral calculation:

$$e=\sum_{i=1}^{n}[h(x_i)-h(x_{i-1})]g_i^n$$

where $h(x_0)=0$ and $g_i^j=g(\{x_i,x_{i+1},...,x_j\})$. The Choquet integral reduces to the Lebesque integral for a probability measure when a probability density function is used for calculations. Comparative results from the fusion of handwritten word classifiers showed similar level of classification performance for both integrals.

5.2.3 Weber Fuzzy Integral

Weber fuzzy integral is a result of an attempt to improve the quality of fuzzy integrals based on the Sugeno fuzzy measure. Weber originally proposed a generalisation of the Sugeno integral and Keller and Tahani have extended this approach introducing a large family of measures called S-decomposable measures. Given a triangular co-norm S, the S-decomposable measure g has the property: $g(A\cup B)=S(g(A),g(B))$ if $A\cap B=0$. The possibility measure is an example of such a measure with S

being the maximum operator. With the above property, a set of information sources can be obtained after the fuzzy densities are determined. An example of the t-conorm is: $S_p(a,b) = (a^p + b^p - a^p b^p)^{1/p}$. Combining the S-decomposable measure with a t-conorm, the generalised fuzzy integral is defined as follows:

$$e_T = \bigcup_i [T(h(x_i), g(\{x_1,...,x_n\}))]$$

A number of t-conorms have been tested with respect to their applicability for information fusion. The use of some of them has resulted in a significant increase of the classification performance.

5.3 Dempster-Shaffer Combination

According to the Dempster-Shaffer theory the universe Θ consists of exhaustive and mutually exclusive logical statements called propositions: $A_i \subset \Theta$, $i = 1,...,M$. Each proposition is assigned a belief value from the range $[0,1]$, which is based on the presence of evidence e. The value of belief is derived from basic probability assignment (BPA) $m(A_i)$, $i = 1,...,M$, which defines an individual impact of each item of evidence on the subsets of the universal set: Θ. If a subset A is given as a disjunction of all elements in A, the belief value of the subset A is given by:

$$bel(A) = \sum_{B \subseteq A} m(b)$$

In the multiple classifier case the universal set of propositions is defined as: $\Theta = \{A_1,...,A_M\}$ and each proposition: $A_i = x \subset C_i$ means that the input sample x comes from class C_i. Supporting evidence is given by K classifiers as: $e_1,...,e_K$. Two parameters are associated with each classifier: recognition rate - $\varepsilon_r^{(k)}$ and substitution rate - $\varepsilon_s^{(k)}$, which represent the measures of an uncertain belief that given proposition is true or is not true respectively. Only non-rejecting classifiers are taken into account for the combination. Also classifiers with the substitution rate equal $\varepsilon_{ss}^{(k)} = 1$ should be removed from the system. If there is one classifier with the recognition rate $\varepsilon_r^{(k)} = 1$, it means that it classifies all input samples with absolute certainty and other classifiers are no longer needed. In a general case the classifier rates are in the range $(0,1)$ and for such classifiers the combination is calculated according to the combination rule:

$$m(A) = m_1 \oplus m_2(A) = k \sum_{X,Y \subseteq \Theta, X \cap Y = A, A \neq 0} m_1(X) m_2(Y)$$

$$\text{where } k^{-1} = 1 - \sum_{X,Y \subseteq \Theta, X \cap Y = 0} m_1(X) m_2(Y) = \sum_{X,Y \subseteq \Theta, X \cap Y \neq 0} m_1(X) m_2(Y)$$

To calculate this combination, all classifiers are grouped according to propositions they produce. Applying this rule sequentially for all classifiers in a group, new combined BPA values are formed for each group: $m_{E_1},...,m_{Ep}$. This is equivalent to obtaining new classifiers with recognition rate $\varepsilon_r^{(k)} = m_{E_k}(A_{j_k})$ and substitution rate $\varepsilon_s^{(k)} = m_{E_k}(\neg A_{j'_k})$ The next step is to combine the BPA values in order to obtain the final belief values. Firstly, for each proposition derived from all groups three constants are calculated:

$$A = \sum_{k=1}^{p} \frac{m_{E_k}(A_{j'_k})}{1 - m_{E_k}(A_{j'_k})}, \quad B = \prod_{k=1}^{p} [1 - m_{E_k}(A_{j_k})],$$

$$C = \prod_{k=1}^{p} m_{E_k}(\neg A_{j'_k}), \quad k^{-1} = \begin{cases} (1+A)B - C & \text{if} \quad p = M \\ (1+A)B & \text{if} \quad p < M \end{cases}$$

The final belief for a given proposition $A_{j'_k}$ is expressed by the following formula:

$$bel(A_{j'_k}) = \begin{cases} k\left[\dfrac{Bm_{E_k}(A_{j'_k}) + Cm_{E_k}(\Theta)}{1 - m_{E_k}(A_{j'_k})}\right] & \text{if} \quad \begin{array}{c}(p = M) \cup \\ [(p = K-1) \cap (k = M)]\end{array} \\ \dfrac{Bm_{E_k}(A_{j'_k})}{1 - m_{E_k}(A_{j'_k})} & \text{otherwise} \end{cases}$$

The decision rule is then very simply given by:

$$E(x) = \begin{cases} j & \text{if} \quad bel(A_j) = \max[bel(A_j)] \\ \text{rejection} & \text{otherwise} \end{cases}$$

Additionally a threshold may be added to accept a certain level of collision between the winning class and remaining alternatives. Extensive experiments have been performed to test the applicability of this method for the classification and a substantial increase in classification rate has been achieved.

5.4 Fuzzy Templates

Fuzzy template technique represents a very simple classifier fusion method that combines the outputs of multiple classifiers. Let $C = \{C_1,...,C_L\}$ be a set of classifiers. Each of the classifiers produces the output: $C_i(x) = [d_{i,1}(x),...,d_{i,c}(x)]^T$ where the value $d_{i,j}(x)$ refers to the degree of support given by classifier C_i that x comes from class j. The outputs of the classifiers

form so-called *decision profile* organised in a matrix holding all the $d_{i,j}(x)$ values:

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \dots & d_{1,c}(x) \\ \dots & \dots & \dots \\ d_{L,1}(x) & \dots & d_{L,c}(x) \end{bmatrix}$$

The fuzzy template definition is closely connected with the training data used. Let $Z = \{Z_1,...,Z_N\}$ be the crisply labelled set of training data. The fuzzy template of the class i is then defined as the $L \times c$ matrix $F_i = \{f_i(k,s)\}$ the elements of which are obtained from:

$$f_i(k,s) = \frac{\sum_{j=1}^{N} Ind(Z_j,i) d_{k,s}(Z_j)}{\sum_{j=1}^{N} Ind(Z_j,i)}$$

where $Ind(Z_j,i)$ is an indicator function with value 1 if Z_j comes from class i and 0 otherwise.

At this stage, the ranking of classes can be achieved by aggregating the columns of DP using a number of possible aggregating operators (minimum, maximum, average, product, weighted average etc). Another method calculates a soft class label vector with components expressing similarity between the decision profile matrix and the fuzzy template matrix. This is defined as follows:

$$CLV = [\mu_{\tilde{D}}^{1},...,\mu_{\tilde{D}}^{i},...,\mu_{\tilde{D}}^{c}]^T \quad \text{where} \quad \mu_{\tilde{D}}^{i} = S(F_i, DP(x))$$

and commonly used similarity operator S,

$$S(F_i, DP(x)) = 1 - \frac{1}{Lc} \sum_{k=1}^{L} \sum_{s=1}^{c} (f_i(k,s) - d_{k,s}(x))^2$$

Now, if the objective is to generate a crisp classification decision, x is assigned to the class with the largest μ value. The fuzzy template method has been tested with ELENA databases and outperformed minimum, maximum and average aggregation rules. The FT technique seems to be very flexible which is especially important while dealing with small training data sets. It is likely that through its flexibility and simplicity, the FT method may outperform other more complex fusion methods requiring substantially larger number of parameters (e.g. fuzzy integrals).

5.5 Product of Experts

A common way of combining different probabilistic models of the same data is to use a mixture by performing weighted average of individual probability distributions. However, this approach is inefficient in high-dimensional problems like faces recognition due to vast complexity and vaguer distribution of mixed models. Another alternative way of combining individual experts is to calculate the product of experts by multiplying individual probabilities and renormalizing. This can be expressed as:

$$p(d/\theta_1...\theta_n) = \frac{\prod_m p_m(d/\theta_m)}{\sum_i \prod_m p_m(c_i/\theta_m)}$$

where d is the data vector in a discrete space, θ_m represents all parameters of an individual model m, $p_m(d/\theta_m)$ is the probability of d obtained from the model m, and i is an index over all possible data vectors c_i. For an individual expert the objective is to assign a high probability to the region of observed data space, and waste as little as possible probability to the unobserved data space. To fit the product of experts (PoE) to the observed data vectors, the derivatives of the log likelihood of each observed data vector have to be computed as given by:

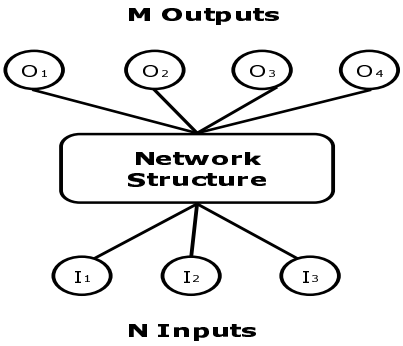
$$\frac{\partial \ln p(d/\theta_1..\theta_n)}{\partial \theta_m} = \frac{\partial \ln p_m(d/\theta_m)}{\partial \theta_m} - \sum_i p(c_i/\theta_1..\theta_n) \frac{\partial \ln p_m(c_i/\theta_m)}{\partial \theta_m}$$

As it can be seen from the above equation, assuming that each expert has tractable derivatives, the only problem remaining is to generate correctly distributed fantasy data. This can be achieved in various ways, but Gibbs sampling seems to be the best method for this purpose. Employing Kullback-Lieber divergence from the true distribution it can be shown that benefits of combining experts come from the disagreeing on the unobserved data. Therefore individual experts have to be initialised by different training data sets or different dimensions of these data. The aim is to force the experts to differ i.e. to teach them separately in order to raise individual probability distributions. Such mixture of experts provides the optimal exploitation of knowledge standing behind the data. PoE's is presented as an unsupervised learning technique and its potential strength has been confirmed by perfect image reconstructions. PoE's can be also adapted to classification problems by comparing the log probabilities under separate, class-specific PoE's.

5.6 Artificial Neural Networks

On a higher level of abstraction an artificial neural network (ANN) is usually viewed as a mapping of n inputs into m outputs as shown below:

Fig 2. Neural Network Scheme



ANNs with their ability to learn from examples and approximate any function to any degree of accuracy, represent a very promising approach to the classifier fusion problem. A neural network designed for the purpose of classifier fusion should have one crisp output or alternatively a number of soft outputs equal to the number of classes if there is a need to produce qualitative assignment values to each class. The input of such a network should be associated with individual classifier outputs.

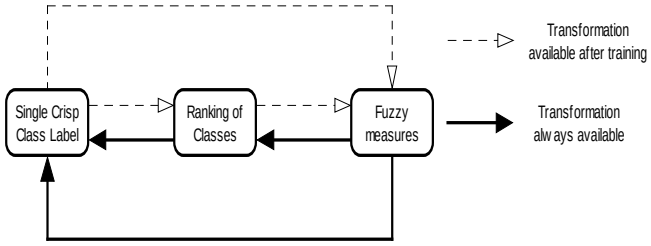
Let the neural network perform a mapping of n individual classifiers outputs (taken as an input) into m outputs corresponding to the level of assignment to each of m classes. If a crisp decision is required, the output with the highest value is chosen. The input-output mapping in ANNs is determined via an iterative learning process. During learning stage, weights between each pair of connected nodes of the network are adapted in such a way as to minimise the difference between the network outputs and expected outputs given in the training data.

It is quite common that a set of ANNs is combined using another ANN. Following this approach neural networks working as a mixture can be expanded to a higher dimension by fusing several neural networks [6] or arranging them in an efficient ANN-like structure [9].

6. CONCLUSIONS

The classifier fusion methods described in this paper cover a large variety of practical applications. The presented taxonomy of fusion techniques has been intended to help in understanding the current state of knowledge in this research area. Additionally we have attempted to identify the main directions leading towards the standardised procedure of multiple classifier system design. Basically, designer of such a system should first concentrate on a careful selection of the relevant classifiers structure. This might be a crucial part of the design in terms of applicability so the structure should reflect the specificity of the problem to be modelled. To do this properly, the types of outputs of the classifiers have to be first defined. If we are faced with different types of outputs, they could be transformed to a uniform type that covers the largest amount of information (preferably a fuzzy measure). In many cases the transformation should be possible through applying additional data from classifiers training process as shown in the following figure:

Fig 3. Classifier outputs transferability



Having optimised the input of the multiple classifier system, one can start the process of reducing redundant classifiers. Simultaneously to the reduction process the selection of the most appropriate groups of classifiers can be carried out. This selection should be guided by maximisation of the diversity among the selected classifiers in each group. In the next step the class set reduction process may take place. Finally a relevant combination method has to be applied for the structured multiple classifiers. If a multistage hierarchical system is to be designed, methods for partial combinations have to be specified. Surprisingly there have been very few attempts to combine the outputs from several fusion methods as a combination on a higher level of abstraction. If carefully selected, they might provide a reduction of information. Taking this line of thinking one step further an interesting question arises: Are the dimensions of information uncertainty really independent? Or as is commonly suspected the information uncertainty in an isolated system is preserved. This could be formulated as the following postulate: *If the information system is isolated from the rest of the information space, the total uncertainty associated with the knowledge of the system is constant*. Proving or otherwise of the above postulate remains an open research issue.

References

[1] P.D. Gader, M.A. Mohamed, J.M. Keller, "Fusion of handwritten word classifiers", *Pattern Recognition Letters*, vol. 17, pt. 6, pp. 577-584, 1996

[2] T.H. Ho, J. J. Hull, S.N. Srihari, "Decision Combination in Multiple Classifier System", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pt. 1, pp. 66-75, 1994

[3] H. Tahani, J.M. Keller "Information Fusion in Computer Vision Using Fuzzy Integral", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, pt. 3, pp. 733-741, 1990

[4] J.M. Keller, P. Gader, H. Tahani, J.H. Chiang, M. Mohamed "Advances in fuzzy integration for pattern recognition", *Fuzzy Sets and Systems*, vol. 65, pp. 273-283, 1994

[5] L.I. Kuncheva, J.C. Bezdek, M.A. Sutton, "On Combining Multiple Classifiers by Fuzzy Templates", *Proc. NAFIPS Conf. EDS.* pp. 193-197, 1998

[6] S.B. Cho, J.H. Kim, "Combining Multiple Neural Networks by Fuzzy Integral for Robust Classification", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, pt. 2, pp. 380-384, 1995

[7] D. Wang, J.M. Keller, C.A. Carson, K.K. McAdoo-Edwards, C.W. Bailey, "Use of Fuzzy-Logic-Inspired Features to Improve Bacterial Recognition Through Classifier Fusion", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, pt. 4, pp. 583-591, 1998

[8] G.E. Hinton, "Products of Experts", *Artificial Neural Networks, Conf.*, No. 470, pp. 1-9, 1999

[9] M.I. Jordan, R.A. Jacobs "Hierarchical mixtures of experts and the EM algorithms", *Neural Computations*, vol. 6, pp. 181-214, 1994

[10] K. Woods, W.P. Kegelmeyer, K. Bowyer, "Combination of Multiple Classifiers Using Local Accuracy Estimates", *IEEE Transactions on Pattern Analysis and Machine Intelligence"*, vol. 19, pt. 4, pp. 405-410, 1997

[11] L. Xu, A. Krzyzak, C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, pt. 3, pp. 418-435, 1992

[12] Y. S. Huang, C. Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals", *IEEE Transactions on Pattern Analysis and Machine Intelligence"*, vol. 17, pt. 1, pp. 90-94, 1995

[13] N.Ueda, R.Nakano, Z. Ghahramani, G.E. Hinton, "SMEM Algorithm for Mixture Models", to appear in *Neural Computations*

[14] W. Pedrycz, J.C. Bezdek, R.J. Hathaway, G.W. Rogers, "Two Nonparametric Models for Fusing Heterogenous Fuzzy Data", *IEEE Transactions on Fuzzy Systems"*, vol. 6, pt. 3, pp. 411-425, 1998

[15] J. Bezdek, "Fuzzy models and algorithms for pattern recognition and image processing ", J. - Boston : Kluwer Academic, 1999

[16] G.J.Klir, T.A. Folger, "Fuzzy Sets, Uncertainty, and Information", Prentice-Hall International Edition, 1988

[17] S. Theodoridis, "Pattern recognition", San Diego: Academic Press, 1999

[18] A.J.C. Sharkey, N.E. Sharkey, "Combining diverse neural nets", *The Knowledge Engineering Review*, vol 12, pt. 3, pp 231-247, 1997

D. Ruta is a Research Student and Dr. B Gabrys is a Lecturer at the University of Paisley