

Combining Classifiers and Learning Mixture-of-Experts

Lei Xu

Chinese University of Hong Kong, Hong Kong & Peking University, China

Shun-ichi Amari

Brain Science Institute, Japan

INTRODUCTION

Expert combination is a classic strategy that has been widely used in various problem solving tasks. A team of individuals with diverse and complementary skills tackle a task jointly such that a performance better than any single individual can make is achieved via integrating the strengths of individuals. Started from the late 1980' in the handwritten character recognition literature, studies have been made on combining multiple classifiers. Also from the early 1990' in the fields of neural networks and machine learning, efforts have been made under the name of ensemble learning or mixture of experts on how to learn jointly a mixture of experts (parametric models) and a combining strategy for integrating them in an optimal sense.

The article aims at a general sketch of two streams of studies, not only with a re-elaboration of essential tasks, basic ingredients, and typical combining rules, but also with a general combination framework (especially one concise and more useful one-parameter modulated special case, called α -integration) suggested to unify a number of typical classifier combination rules and several mixture based learning models, as well as max rule and min rule used in the literature on fuzzy system.

BACKGROUND

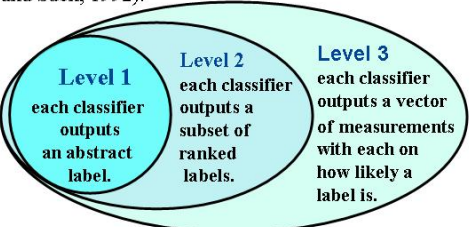
Both streams of studies are featured by two periods of developments. The first period is roughly from the late 1980s to the early 1990s. In the handwritten character recognition literature, various classifiers have been developed from different methodologies and different features, which motivate studies on combining multiple classifiers for a better performance. A systematical effort on the early stage of studies was made in (Xu,

Krzyzak & Suen, 1992), with an attempt of setting up a general framework for classifier combination. As re-elaborated in Tab.1, not only two essential tasks were identified and a framework of three level combination was presented for the second task to cope with different types of classifier's output information, but also several rules have been investigated towards two of the three levels, especially with Bayes voting rule, product rule, and Dempster-Shafer rule proposed. Subsequently, the rest one (i.e., rank level) was soon studied in (Ho, Hull, & Srihari, 1994) via Borda count.

Interestingly and complementarily, almost in the same period the first task happens to be the focus of studies in the neural networks learning literature. Encountering the problems that there are different choices for the same type of neural net by varying its scale (e.g., the number of hidden units in a three layer net), different local optimal results on the same neural net due to different initializations, studies have been made on how to train an ensemble of diverse and complementary networks via cross-validation-partitioning, correlation reduction pruning, performance guided re-sampling, etc, such that the resulted combination produces a better generalization performance (Hansen & Salamon, 1990; Xu, Krzyzak, & Suen, 1991; Wolpert, 1992; Baxt, 1992, Breiman, 1992&94; Drucker, et al, 1994). In addition to classification, this stream also handles function regression via integrating individual estimators by a linear combination (Perrone & Cooper, 1993). Furthermore, this stream progresses to consider the performance of two tasks in Tab.1 jointly in help of the mixture-of-expert (ME) models (Jacobs, et al, 1991; Jordan & Jacobs, 1994; Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994), which can learn either or both of the combining mechanism and individual experts in a maximum likelihood sense.

Two stream studies in the first period jointly set up a landscape of this emerging research area, together

Table 1. Essential tasks and their implementations

Two Tasks (a quotation from Xu, Krzyzak and Suen, 1992)					
<p><i>Task 1</i>: “How many and what type of classifiers should be used for a specific application?, and for each classifier what type of features should we use?, as well as other problems that are related to the construction of those individual and complementary classifier”.</p> <p><i>Task 2</i>: “How to combine the results from different existing classifiers so that a better result can be obtained?”</p>					
Two Styles of Implementations					
Two Stage Implementation	Joint Implementation				
<ul style="list-style-type: none"> • <i>Task 1</i> is completed in advance, with the resulted classifiers being diverse and complementary. • Perform <i>Task 2</i> in one of three levels (Xu, Krzyzak and Suen, 1992). 	<p>Two tasks made jointly or alternatively</p> <table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: center;">under a same criterion</th> <th style="text-align: center;">others</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • Mixture of experts (ME) (Jacobs, et al, 1991; Jordan & Jacobs, 1994); • Alternative ME (Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994); • EM-RBF (Xu, 1998) • Three layer nets, etc. </td> <td> <p>Stacking , Boosting, ... , etc (Breiman, 1992&94; Wolpert, 1992)</p> </td> </tr> </tbody> </table>	under a same criterion	others	<ul style="list-style-type: none"> • Mixture of experts (ME) (Jacobs, et al, 1991; Jordan & Jacobs, 1994); • Alternative ME (Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994); • EM-RBF (Xu, 1998) • Three layer nets, etc. 	<p>Stacking , Boosting, ... , etc (Breiman, 1992&94; Wolpert, 1992)</p>
under a same criterion	others				
<ul style="list-style-type: none"> • Mixture of experts (ME) (Jacobs, et al, 1991; Jordan & Jacobs, 1994); • Alternative ME (Xu & Jordan, 1993; Xu, Jordan & Hinton, 1994); • EM-RBF (Xu, 1998) • Three layer nets, etc. 	<p>Stacking , Boosting, ... , etc (Breiman, 1992&94; Wolpert, 1992)</p>				

with a number of typical topics or directions. Thereafter, further studies have been further conducted on each of these typical directions. First, theoretical analyses have been made for deep insights and improved performances. For examples, convergence analysis on the EM algorithm for the mixture based learning are conducted in (Jordan & Xu, 1995; Xu & Jordan, 1996). In Tumer & Ghosh (1996), the additive errors of posteriori probabilities by classifiers or experts are considered, with variances and correlations of these errors investigated for improving the performance of a sum based combination. In Kittler, et al (1998), the effect of these errors on the sensitivity of sum rule vs product rule are further investigated, with a conclusion that summation is much preferred. Also, a theoretical framework is suggested for taking several combining rules as special cases (Kittler, 1998), being unaware of that this framework is actually the mixture-of-experts model that was proposed firstly for combining multiple function regressions in (Jacobs, et al, 1991) and then for combining multiple classifiers in (Xu & Jordan, 1993). In addition, another theoretical study is made on six classifier fusion strategies in (Kuncheva, 2002). Second, there are further studies on Dempster-Shafer rule (Al-Ania, 2002) and other combining methods such as rank based, boosting based, as well as local

accuracy estimates (Woods, Kegelmeyer, & Bowyer, 1997). Third, there are a large number of applications. Due to space limit, details are referred to Ranawana & Palade (2006) and Sharkey & Sharkey (1999).

A GENERAL ARCHITECTURE, TWO TASKS, AND THREE INGREDIENTS

We consider a general architecture shown in Fig.1. There are $\{e_j(x)\}_{j=1}^k$ experts with each $e_j(x)$ as either a classifier or an estimator. As shown in Tab.2, a classifier outputs one of three types of information, on which we have three levels of combination. The first two can be regarded as special cases of the third one that outputs a vector of measurements. A typical example is $[p_j(1|x), \dots, p_j(m|x)]^T$ with each $1 \geq p_j(\ell|x) \geq 0$ expressing a posteriori probability that x is classified to the ℓ -th class. Also, $p_j(\ell|x) = p_j(y = \ell|x)$ can be further extended to $p_j(y|x)$ that describes a distribution for a regression $x \rightarrow y \in R^m$. In Figure 1, there is also a gating net that generates signals $\{\alpha_j(x)\}_{j=1}^k$ to modulate experts by a combining mechanism $M(x)$.

Figure 1. A general architecture for expert combination

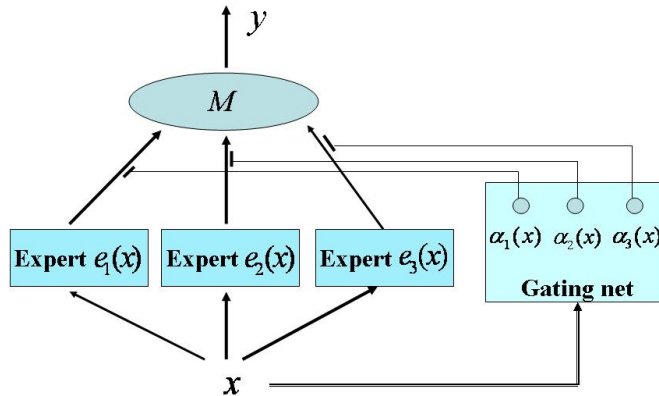


Table 2. Three levels of combination

Three Rules for Combination on Level 3		
Sum rule (Bayes Voting)	Product rule	Dempster-Shafer rule
<p>Given k classifiers, the j-th classifier classifies x to y with a probability $p_j(y x) = P(x \in C_y e_j(x))$, we sum up to get a combination</p> $P(y x) = \frac{1}{k} \sum_{j=1}^k p_j(y x)$ <p>See eqn.(4) in (Xu, Krzyzak and Suen, 1992)</p>	<p>If k classifiers are independent, another combination is given by</p> $P(x \in C_y) = \frac{\prod_{j=1}^k P(x \in C_y e_j(x))}{\prod_{j=1}^k P(x \in C_j)}$ <p>or concisely</p> $P(y x) = P^{1-k}(y) \prod_{j=1}^k P_j(y x)$ <p>See eqn.(31) in (Xu, Krzyzak and Suen, 1992)</p>	$bel(A) = \sum_{B \subseteq A} m(B)$ $m(A) = \frac{\sum_{X \cap Y = A} m_x(X) m_y(Y)}{\sum_{X \cap Y \neq \emptyset} m_x(X) m_y(Y)}$ <p>A_z denotes $x \in C_z$, $\theta = \{A_1, \dots, A_m\}$ if $e_j(x) = \ell$, $m_j(A_\ell) = \epsilon_j^{(\ell)}$ $\neg A_\ell = \theta - \{A_\ell\}$ $m_j(\neg A_\ell) = \epsilon_j^{(\ell)}$ $m_j(\theta) = 1 - \epsilon_j^{(\ell)} - \epsilon_j^{(\ell)}$</p> <p>See Sec.VI in (Xu, Krzyzak and Suen, 1992)</p>

Based on this architecture, two essential tasks of expert combination could be still quoted from (Xu, Krzyzak & Suen, 1992) with a slight modification on Task 1, as shown in Tab.1, that the phrase ‘for a specific application?’ should be deleted in consideration of the previously introduced studies (Hansen & Salamon, 1990; Xu, Krzyzak, & Suen, 1991; Wolpert, 1992; Baxt, 1992, Breiman, 1992&94; Drucker, et al, 1994; Tumer & Ghosh, 1996).

Insights can be obtained by considering three basic ingredients of two streams of studies, as shown in Fig.2. Combinatorial choices of different ingredients lead to different specific models for expert combination, and differences in the roles by each ingredient highlight the different focuses of two streams. In the stream

of neural networks and machine learning, provided with a structure for each $e_j(x)$, a gating structure, and a combining structure $M(x)$, all the rest unknowns are determined under guidance of a learning theory in term of minimizing an error cost. Such a minimization is implemented via an optimizing procedure by a learning algorithm, based on a training set $\{x_t, y_t\}_{t=1}^N$ that teaches a target y_t for each mapping $x_t \rightarrow R^m$. While in the stream of combing classifiers, all $\{p_j(y|x)\}_{j=1}^k$ are known without unknowns left to be specified. Also, M is designed according to certain heuristics or principles, with or without help of a training set, and studies are mainly placed on developing and analyzing different combining mechanisms, for which we will further dis-

cuss subsequently. The final combining performance is empirically evaluated by the misclassification rate, but there is no effort yet on developing a theory for one M that minimizes the misclassification rate or a cost function, though there are some investigations on how estimated posteriori probabilities can be improved by a sum rule and on error sensitivity of estimated posteriori probabilities (Tumer & Ghosh, 1996; Kittler, et al, 1998). This under-explored direction also motivate future studies subsequently.

***f*-COMBINATION**

The arithmetic, geometric, and harmonic mean of non-negative number $b_j \geq 0, j = 1, \dots, k$ has been further extended into one called:

$$f\text{-mean } m_f = f^{-1}\left(\sum_{j=1}^k \alpha_j f(b_j)\right),$$

where $f(r)$ is a monotonic scalar function, and

$$\alpha_j > 0 \sum_{j=1}^k \alpha_j = 1$$

(Hardy, Littlewood, & Polya, 1952).

We can further generalize this *f-mean* to the general architecture shown in Fig.1, resulting in the following *f-combination*:

$$M(x) = f^{-1}\left(\sum_{j=1}^k \alpha_j(x) f(p_j(y|x))\right), \text{ or}$$

$$f(M(x)) = \sum_{j=1}^k \alpha_j(x) f(p_j(y|x))$$

where

$$\alpha_j > 0 \sum_{j=1}^k \alpha_j(x) = 1.$$

In the following, we discuss to use it as a general framework to unify not only typical classifier combining rules but also mixture-of-expert learning and RBF net learning, as shown in Tab.3.

We observe the three columns for three special cases of $f(r)$. The first column is the case $f(r) = r$, we return to the ME model:

$$M(x) = \sum_{j=1}^k \alpha_j(x) p_j(y|x),$$

which was proposed firstly for combining multiple regressions in (Jacobs, et al, 1991) and then for combining classifiers in (Xu & Jordan, 1993). For different special cases of $\alpha_j(x)$, we are lead to a number of existing typical examples. As already pointed out in (Kittler, et al, 1998), the first three rows are four typical classifier combining rules (the 2nd row directly applies to the min-rule too). The next three rows are three types of ME learning models, and a rather systematic summary

Figure 2. Three basic ingredients

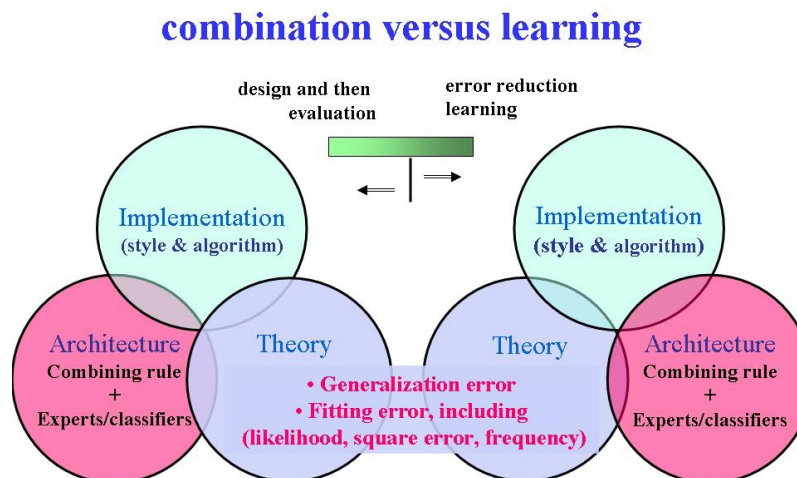
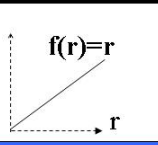
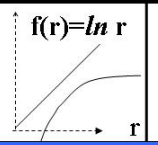
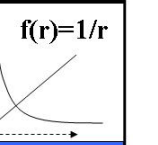


Table 3. Typical examples (including several existing rules and potential topics)

$f(M)$ $f(r)$ $\alpha_j(x)$			
$\alpha_j(x) = 0$, except $\alpha_j(x) = 1$ $j = \begin{cases} \operatorname{argmax}_j p_j(y x), & \text{(a)} \\ \operatorname{argmin}_j p_j(y x), & \text{(b)} \end{cases}$	(a) max-rule (b) min-rule	(a) max-rule (b) min-rule	(a) max-rule (b) min-rule
$\alpha_j(x) = 1/k$	Average Bayes or Bayes voting (Xu, Krzyzak & Suen, 1992)	Product rule (Xu, Krzyzak & Suen, 1992; Kittler, et al, 1998; Hinton, 2002)	Harmonic mean
$\alpha_j(x) = \alpha_j = \frac{\sigma_j^{-2}}{\sum_{j=1}^k \sigma_j^{-2}}$	Mixture using variances (MUV) (Perrone & Cooper, 1993)	To be explored	To be explored
$\alpha_j(x) = \frac{e^{g_j(x, \phi)}}{\sum_{j=1}^k e^{g_j(x, \phi)}}$	Mixture-of-experts (ME) (Jacobs, et al, 1991)	To be explored	To be explored
$\alpha_j(x) = \frac{\beta_j p(x \phi_j)}{\sum_{j=1}^k \beta_j p(x \phi_j)}$	Alternative ME (Xu, Jordan & Hinton, 1994)	To be explored	To be explored
$\alpha_j(x) = \frac{\beta_j G(x \mu_j, \Sigma_j)}{\sum_{j=1}^k \beta_j G(x \mu_j, \Sigma_j)}$, subject to $\beta_j / \sqrt{ \Sigma_j } = \text{const}$	Extended Normalized RBF (Xu, 1998)	To be explored	To be explored
$\alpha_j(x) = \sigma_j^{-2}(x) / \sum_{j=1}^k \sigma_j^{-2}(x)$	Belief net based MUV (Lee, et al, 2006)	To be explored	To be explored

is referred to Sec. 4.3 in (Xu, 2001). The last row is a recent development of the 3rd row.

The 2nd row of the 2nd column is the geometric mean:

$$M(x) = \sqrt[k]{\prod_{j=1}^k p_j(y|x)},$$

which is equal to the product rule (Xu, Krzyzak and Suen, 1992; Kittler, et al, 1998, Hinton, 2002) if each a priori is equal, i.e., $\alpha_j(x) = 1/m$. Generally if $\alpha_j(x) \neq 1/m$, there is a difference by a scaling factor $\alpha_j(x)^{1/k-1}$. The product rule works in a probability theory sense under a condition that classifiers are mutually independent. In (Kittler, et al, 1998), attempting to discuss a number of rules under a unified system, the sum rule is approximately derived from the product rule, under an extra condition that is usually difficult to satisfy. Actually, such an imposed link between the product rule and the sum rule is unnecessary, the sum:

$$M(x) = \sum_{j=1}^k \alpha_j(x) p_j(y|x)$$

is just a marginal probability

$$\sum_{j=1}^k p(y, j|x),$$

which is already in the framework of probability theory. That is, both the sum rule and the product rule already coexist in the framework of probability theory.

On the other hand, it can be observed that the sum:

$$\sum_{j=1}^k \alpha_j(x) \ln p_j(y|x)$$

is dominated by a $p_j(y|x)$ if it is close to 0. That is, this combination expects that every expert should cast enough votes, otherwise the combined votes will be still very low just because there is only one that casts a very low vote. In other words, this combination can

be regarded as a relaxed logical AND that is beyond the framework of probability theory when $\alpha_j(x) \neq 1/m$. However, staying within the framework of probability theory does not mean that it is better, not only because it requires that classifiers are mutually independent, but also because there lacks theoretical analysis on both rules in a sense of classification errors, for which further investigations are needed.

In Tab.2, the 2nd row of the third column is the harmonic mean. It can be observed that the problem of combining the degrees of support is changed into a problem of combining the degrees of disagree. This is interesting. Unfortunately, efforts of this kind are seldom found yet. Exceptionally, there are also examples that can not be included in the f -combination, such as Dempster-Shafer rule (Xu, Krzyzak and Suen, 1992; Al-Ania, 2002) and rank based rule (Ho, Hull, Srihari, 1994).

α -INTEGRATION

After completed the above f -combination, the first author becomes aware of the work by (Hardy, Littlewood, & Polya, 1952) through one coming paper (Amari, 2007) that studies a much concise and more useful one-parameter modulated special case called α -integration. With help of a concrete mathematical foundation from an information geometry perspective. Imposing an additional but reasonable nature that the f -mean should be linear scale-free, i.e.:

$$cm_f = f^{-1}(\sum_{j=1}^k \alpha_j f(cb_j))$$

for any scale c , alternative choices of $f(r)$ reduces into the following only one:

$$f_\alpha(r) = \begin{cases} r^{0.5(1-\alpha)}, & \alpha \neq 1, \\ \ln r, & \alpha = 1. \end{cases}$$

It is not difficult to check that

$$f_\alpha(r) = \begin{cases} r, & \alpha = -1, \\ \ln r, & \alpha = 1, \\ 1/r, & \alpha = 3. \end{cases}$$

Thus, the discussions on the examples in Tab.2 are applicable to this $f_\alpha(r)$. Moreover, the first row in Tab.2 holds when $\alpha = -\infty$ and $\alpha = +\infty$ for whatever a gating net, which thus includes two typical operators of the fuzzy system as special case too. Also, the family is systematically modulated by a parameter $-\infty \leq \alpha \leq +\infty$, which provides not only a spectrum from the most optimistic integration to the most pessimistic integration as varying from $\alpha = -\infty$ to $\alpha = +\infty$ but also a possibility of adapting α for a best combining performance.

Furthermore, Amari (2007) also provides a theoretical justification that α -integration is optimal in a sense of minimizing a weighted average of α -divergence. Moreover, it provides a potential road for studies on combining classifiers and learning mixture models from the perspective of information geometry.

FUTURE TRENDS

Further studies are expected along several directions as follows:

- Empirical and analytical comparisons on performance are needed for those unexplored or less explored items in Tab.2.
- Is there a best structure for $\alpha_j(x)$? comparisons need to be made on different types of $\alpha_j(x)$, especially the ones by the MUV type in the last row and the ME types from the 4th to the 7th rows,
- Is it necessary to relax the constraint:

$$\alpha_j(x) > 0, \sum_{j=1}^k \alpha_j(x) = 1,$$

e.g., removing non-negative requirement and to relax the distribution $p_j(y|x)$ to other types of functions ?

- How weights $\alpha_j(x)$ can be learned under a generalization error bound.
- As discussed in Fig.2, classifier combination and mixture based learning are two aspects with different features. How to let each part to take their best roles in an integrated system?

CONCLUSION

Updating the purpose of (Xu, Krzyzak & Suen, 1992), the article provides not only a general sketch of studies

on combining classifiers and learning mixture models, but also a general combination framework to unify a number of classifier combination rules and mixture based learning models, as well as a number of directions for further investigations.

ACKNOWLEDGMENT

The work is supported by Chang Jiang Scholars Program by Chinese Ministry of Education for Chang Jiang Chair Professorship in Peking University.

REFERENCES

Al-Ania, A., (2002), A New Technique for Combining Multiple Classifiers using The Dempster-Shafer Theory of Evidence, *Journal of Artificial Intelligence Research* 17, 333-361.

Amari, S., (2007), Integration of stochastic models by minimizing α -divergence, *Neural Computation* 19(10), 2780-2796.

Baxt, W. G. (1992), Improving the accuracy of an artificial neural network using multiple differently trained networks, *Neural Computation* 4, 772-780.

Breiman, L., (1992), Stacked Regression, Department of Statistics, Berkeley, TR-367.

Breiman, L., (1994), Bagging Predictors, Department of Statistics, Berkeley, TR-421.

Drucker, H., Cortes, C., Jackel, L.D., LeCun, Y. & Vapnik, V., (1994) Boosting and other ensemble methods, *Neural Computation* 6, 1289-1301.

Hardy, G.H., Littlewood, J.E., and Polya, G. (1952), *Inequalities*, 2nd edition, Cambridge University Press.

Hansen, L., K. & Salamon, P. (1990), Neural network ensembles. *IEEE Transactions Pattern Analysis Machine Intelligence* 12(10), 993-1001.

Hinton, G.E. (2002), Training products of experts by minimizing contrastive divergence, *Neural Computation* 14, 1771-1800.

Ho, T.K., Hull, J.J., Srihari, S., N., (1994), Decision combination in multiple classifier systems, *IEEE Transactions Pattern Analysis Machine Intelligence* 16(1), 66-75

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E., (1991), Adaptive mixtures of local experts, *Neural Computation* 3, 79-87.

Jordan, M. I., & Jacobs, R. A., (1994), Hierarchical mixtures of experts and the EM algorithm, *Neural Computation* 6, 181-214.

Jordan, M.I., & Xu, L., (1995), Convergence Results for The EM Approach to Mixtures of Experts Architectures, *Neural Networks* 8, 1409-1431.

Kittler, J., Hatef, M., Duinand, R., P., W., Matas, J., (1998) On combining classifiers, *IEEE Trans. Pattern Analysis Machine Intelligence* 20(3), 226-239.

Kittler, J., (1998), Combining classifiers: A theoretical framework, *Pattern Analysis and Applications* 1, 18-27.

Kuncheva, L., I., (2002), A theoretical study on six classifier fusion strategies, *IEEE Transactions Pattern Analysis Machine Intelligence* 24(2), 281-286.

Lee, C., Greiner, R., Wang, S., (2006), Using query-specific variance estimates to combine Bayesian classifiers, *Proc. of the 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, June (529-536).

Perrone, M.P., & Cooper, L.N., (1993), When networks disagree: Ensemble methods for neural networks, *Neural Networks for Speech and Image Processing*, Mammone, R., J., editor, Chapman Hall.

Ranawana, R. & Palade, V., (2006), Multi-Classifer Systems: Review and a roadmap for developers, *International Journal of Hybrid Intelligent Systems* 3(1), 35 - 61.

Shafer, G., (1976), *A mathematical theory of evidence*, Princeton University Press.

Sharkey, A.J. & Sharkey, N.E., (1997), Combining diverse neural nets, *Knowledge Engineering Review* 12(3), 231-247.

Sharkey, A.J. & Sharkey, N.E., (1999.), *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, Springer-Verlag, New York, Inc.

Tumer, K. & Ghosh, J. (1996), Error correlation and error reduction in ensemble classifiers, *Connection Science*, 8 (3/4), 385-404.

Wolpert, D.H., (1992), Stacked generalization, *Neural Networks* 5(2), 241-260.

Woods, K., Kegelmeyer, K.P., & Bowyer, K., (1997), Combination of multiple classifiers using local accuracy estimates, *IEEE Transactions Pattern Analysis Machine Intelligen* 19(4), 405-410.

Xu, L (2007), A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving, *Pattern Recognition* 40, 2129–2153.

Xu, L (2001), Best harmony, unified RPCL and automated model selection for unsupervised and supervised learning on Gaussian mixtures, three-layer nets and ME-RBF-SVM Models, *International Journal of Neural Systems* 11(1), 43-69.

Xu, L., (1998), RBF Nets, Mixture Experts, and Bayesian Ying-Yang Learning, *Neurocomputing* 19, 223-257.

Xu, L. & Jordan, M.I., (1996), On convergence properties of the EM algorithm for Gaussian mixtures, *Neural Computation*, 8(1), 129-151.

Xu, L., Jordan, M.I., & Hinton, G.E., (1995), An Alternative Model for Mixtures of Experts, *Advances in Neural Information Processing Systems 7*, Cowan, Tesauro, and Alspector, editors, MIT Press, 633-640

Xu, L., Jordan, M.I., & Hinton, G.E., (1994), A Modified Gating Network for the Mixtures of Experts Architecture, *Proc. WCNN94*, San Diego, CA, (2) 405-410.

Xu, L. & Jordan, M.I., (1993), EM Learning on A Generalized Finite Mixture Model for Combining Multiple Classifiers, *Proc. of WCNN93*, (IV) 227-230.

Xu, L., Krzyzak, A. & Sun, C.Y., (1992), Several methods for combining multiple classifiers and their applications in handwritten character recognition, *IEEE Transactions on System, Man and Cybernetics* 22, 418-435.

Xu, L., Krzyzak, A. & Sun, C.Y., (1991), Associative Switch for Combining Multiple Classifiers, *Proc. of IJCNN91*, July 8-12. Seattle, WA, (I) 43-48.

Xu, L. Oja, E., & Kultanen, P., (1990), A New Curve Detection Method Randomized Hough transform (RHT), *Pattern Recognition Letters* 11, 331-338.



KEY TERMS

Conditional Distribution $p(y|x)$: Describes the uncertainty that an input x is mapped into an output y that simply takes one of several labels. In this case, x is classified into the class label y with a probability $p(y|x)$. Also, y can be a real-valued vector, for which x is mapped into y according density distribution $p(y|x)$.

Classifier Combination: Given a number of classifiers, each classifies a same input x into a class label, and the labels maybe different for different classifiers. We seek a rule $M(x)$ that combines these classifiers as a new one that performs better than anyone of them.

Sum Rule (Bayes Voting): A classifier classifies x to a label y can be regarded as casting one vote to this label, a simplest combination is to count the votes received by every candidate label. The j -th classifier classifies x to a label y with a probability $p_j(y|x)$ means that one vote is divided to different candidates in fractions. We can sum up:

$$\sum_j p_j(y|x)$$

to count the votes on a candidate label y , which is called Bayes voting since $p(y|x)$ is usually called Bayes posteriori probability.

Product Rule: When k classifiers $\{e_j(x)\}_{j=1}^k$ are mutually independent, a combination is given by

$$p(x \in C_y | x) = p(x \in C_y) \frac{\prod_{j=1}^k p(x \in C_y | e_j(x))}{\prod_{j=1}^k p(x \in C_y)}$$

or concisely

$$p(y | x) = p^{1-k}(y) \prod_{j=1}^k p_j(y | x),$$

which is also called product rule.

Mixture of Experts: Each expert is described by a conditional distribution $p_j(y | x)$ either with y taking one of several labels for a classification problem or with y being a real-valued vector for a regression problem. A combination of experts is given by:

$$M(x) = \sum_{j=1}^k \alpha_j(x) p_j(y | x), \alpha_j(x) = p(j|x) > 0, \sum_{j=1}^k \alpha_j(x) = 1,$$

which is called a mixture-of-experts model. Particularly, for y in a real-valued vector, its regression form is

$$E(y | x) = \sum_{j=1}^k \alpha_j(x) f_j(x), f_j(y) = \int y p_j(y | x) dy.$$

f-Mean: Given a set of non-negative numbers $b_j \geq 0, j = 1, \dots, k$, the f -mean is given by:

$$m_f = f^{-1}(\sum_{j=1}^k \alpha_j f(b_j)),$$

where $f(r)$ is a monotonic scalar function and

$$\alpha_j > 0 \sum_{j=1}^k \alpha_j = 1.$$

Particularly, one most interesting special case is that $f(r)$ satisfies

$$cm_f = f^{-1}(\sum_{j=1}^k \alpha_j f(cb_j))$$

for any scale c , which is called f_a -mean.

Performance Evaluation Approach: It usually works in the literature on classifier. Combination, with a chart flow that considering a set of classifiers $\{e_j(x)\}_{j=1}^k \rightarrow$ designing a combining mechanism $M(x)$ according to certain principles \rightarrow evaluating performances of combination empirically via misclassification rates, in help of samples with known correct labels.

Error-Reduction Approach: It usually works in the literature on mixture based learning, where what needs to be pre-designed is the structures of classifiers or experts, as well as the combining structure $M(x)$ with unknown parameters. A cost or error measure is evaluated via a set of training samples, and then minimized through learning all the unknown parameters.