Convergence of Generalized Probability Mixtures That Describe Adaptive Fuzzy Rule-based Systems

Bart Kosko
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, California 90089-2564
kosko@usc.edu

***Abstract*—A generalized probability mixture density governs the rule-base structure of an additive fuzzy system. A new theorem allows such a mixture to absorb any bounded real function by mixing two normal densities. We further show that a sequence of adaptive fuzzy systems defines a corresponding sequence of uniformly convergent generalized Gaussian mixtures. The result applies to any uniformly convergent sequence of function approximators. It gives a practical way to define approximating mixtures with adaptive fuzzy systems or neural networks. Users can combine any number of these rule-based systems by mixing their generalized mixtures.**

***Index Terms*—additive fuzzy system, generalized probability mixture, Bayes rule posterior, Watkins representation**

I. Generalized Mixtures and Fuzzy Systems

There is a deep connection between generalized probability measures and fuzzy rule-based systems. A set of m fuzzy if-then rules $R_{A_1 \to B_1}, \ldots, R_{A_m \to B_m}$ defines a generalized probability mixture $p(y|x) = p_1(x)\ p_{B_1}(y|x) + \cdots + p_m(x)\ p_{B_m}(y|x)$ if the fuzzy system $F : \mathbb{R}^n \to \mathbb{R}^p$ is *additive* [1]. The mixture arises directly from summing the the m fired then-part sets $B_1, \ldots, B_m$ in (5) below. This mixture structure does not hold for earlier min-max fuzzy systems [2]–[4].

The mixture is generalized because the convex mixing weights $p_j(x)$ depend on the input x. The m mixture weights $p_j(x)$ and likelihood densities $p_{B_j}(y|x)$ fully absorb the structure of the m rules. Such mixtures allow the user to combine any number of rule-based systems into a common rule base because mixing mixtures always produces a mixture. It also allows the user to sample from a virtual rule continuum by drawing rule samples from the mixture $p(y|x)$. The sample creates a fresh and statistically representative set of rules for each input x and can also help mitigate rule explosion [1].

A more general result holds in the other direction: A given generalized mixture $p(y|x)$ gives rise to an additive fuzzy system F and all its higher-order statistical moments [1]. The second moment gives the conditional variance $V[Y|X = x]$ in (17) that describes the confidence of a given fuzzy-system output $F(x)$. This variance also endows a sampled neural network with a confidence measure when an adaptive fuzzy system approximates the network [5], [6].

Theorem 1 shows that a stronger result also holds: Mixing just 2 normal densities gives a generalized mixture $p(y|x)$ whose conditional expectation $E[Y|X = x]$ exactly represents any bounded non-constant real function $f : \mathbb{R}^n \to \mathbb{R}$ in the sense that $f(x) = E[Y|X = x]$ for all x. The representation is *exact* and not a mere approximation. This mixture result generalizes the earlier Watkins Representation Theorem that showed that an additive fuzzy system with just 2 rules can represent any such bounded function [7], [8]. Figure 1 shows the generalized mixture $p(y|x)$ that absorbs the target function $f(x) = \sin x$ by mixing two unit-variance normal bell curves centered at the infimum and supremum of the bounded function f. A similar mixture absorbs any other bounded function.

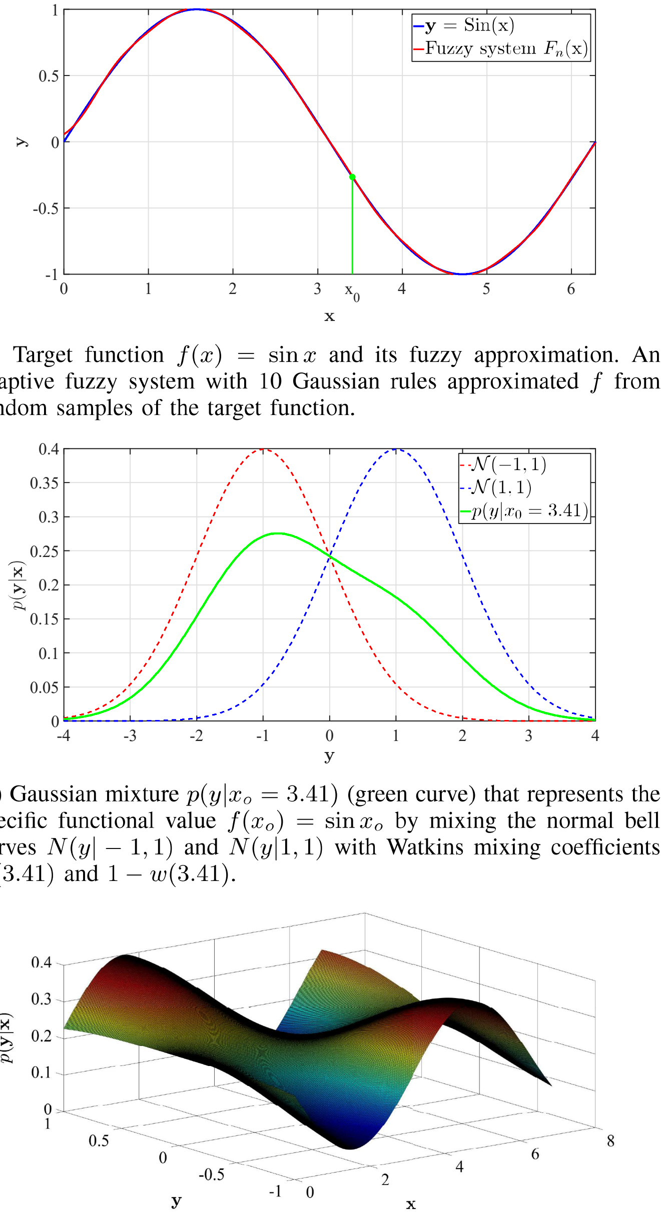
Theorem 2 shows further that adapting an additive fuzzy system leads to a convergent sequence of generalized mixtures $q_n(y|x)$ for all x. The generalized mixtures $q_n(y|x)$ mix the same two normal bell curves. They converge uniformly to the mixture $p(y|x)$ as the underlying fuzzy systems F_n converge to f: $q_n(y|x) \to p(y|x)$ uniformly in y and x as $F_n \to f$ uniformly in x.

Theorem 2 gives a practical way to define mixture representations using any kind of uniform function approximator F_n. This includes adaptive fuzzy systems and deep neural networks and even Bernstein polynomials. The user can simply insert the approximator F_n into the mixture's two modified Watkins mixing coefficients in (31) - (32).

Figure 1 illustrates the mixture representation of Theorem 1. It shows how mixing the two unit-variance normal bell curves $N(y| - 1, 1)$ and $N(y|1, 1)$ defines a generalized mixture $p(y|x)$ that gives back $f(x) = \sin x$ in the first panel upon averaging over $p(y|x)$: $\sin x = E[Y|X = x]$. The second panel shows the mixture $p(y|x_o)$ that results for the fixed value $x_o = 3.41$. The third panel shows the complete mixture $p(y|x)$ surface for all x and y values.

Figure 2 illustrates the uniform convergence of Theorem 2. It shows a snapshot of two different mixture representations after their corresponding additive fuzzy systems F_n have learned for $n = 10,000$ iterations. The first panel shows the approximating mixture $q_n(y|x)$ for an additive fuzzy system with 10 Gaussian rules. The second panel shows the corresponding mixture when F_n replaces f in Theorem 1. The last two panels show the two mixtures for an additive fuzzy system with 10 sinc rules. The approximated mixtures in the second and fourth panels are indistinguishable from the mixture $p(y|x)$ in Figure 1 that exactly represents $f(x) = \sin x$.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE



(a) Target function $f(x) = \sin x$ and its fuzzy approximation. An adaptive fuzzy system with 10 Gaussian rules approximated f from random samples of the target function.

(b) Gaussian mixture $p(y|x_o = 3.41)$ (green curve) that represents the specific functional value $f(x_o) = \sin x_o$ by mixing the normal bell curves $N(y| -1, 1)$ and $N(y|1, 1)$ with Watkins mixing coefficients $w(3.41)$ and $1 - w(3.41)$.

(c) Fuzzy mixture with Watkins coefficients $q_n(y|x_0 = 3.41)$

Fig. 1: Generalized Gaussian mixture representation of a bounded function in accord with Theorem 1. The three panels show the generalized probability mixture $p(y|x)$ that represents the bounded function $f(x) = \sin x$ using just the two mixed unit-variance normal bell curves $N(y| -1, 1)$ and $N(y|1, 1)$ centered at the respective infimum $\alpha = \inf \sin x = -1$ and supremum $\beta = \sup \sin x = 1$.

II. Mixture Structure of Additive Systems

We first review the mixture structure that underlies additive fuzzy systems. The mixture gives a Bayes posterior over the rules for each input-output pair (x, y). It also gives a conditional variance $V[Y|X = x]$ that describes the inherent uncertainty of a fuzzy system's answer to a question. We present this in formal detail because the main theoretical results below depend on it. We begin with the convex structure of additive fuzzy systems and how an input fires the m stored rules.

A. Rule Firing as Delta-Spike Convolution

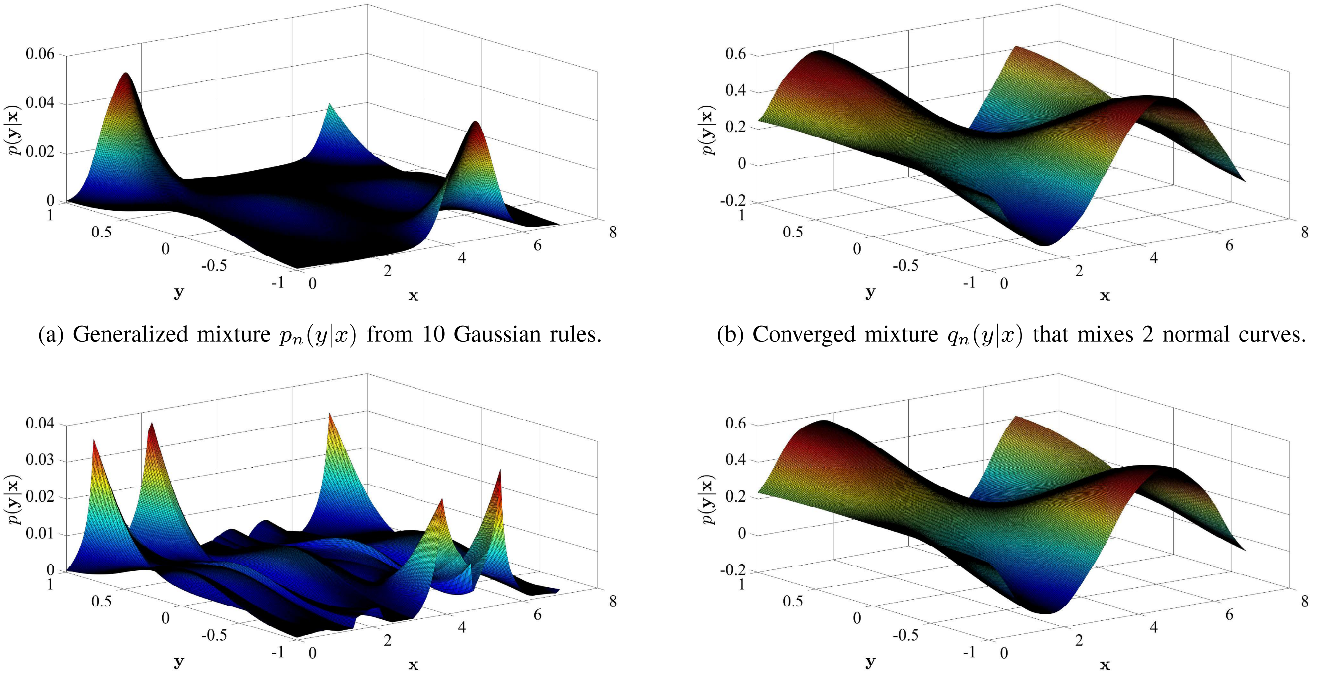
An additive fuzzy system $F : \mathbb{R}^n \to \mathbb{R}^p$ adds and then averages the fired then-part fuzzy sets $B_j \subset \mathbb{R}^p$ of its m rules $R_{A_1 \to B_1}, \ldots, R_{A_m \to B_m}$. The fuzzy system defines a map or process that takes an input vector $x \in \mathbb{R}^n$ through the rule base and produces a system output $F(x) \in \mathbb{R}^p$. The output turns out to be a convex combination of the m centroids $c_1, \ldots, c_m$ of the respective m then-part fuzzy sets $B_1, \ldots, B_m$.

The process begins when vector input $x \in \mathbb{R}^n$ fires each of the m rules $R_{A_j \to B_j}$ in parallel. The if-part fuzzy set $A_j \subset \mathbb{R}^n$ has membership or set function $a_j : \mathbb{R}^n \to [0, 1]$ [9], [10]. The then-part fuzzy set $B_j \subset \mathbb{R}^n$ has set function $b : \mathbb{R}^p \to [0, 1]$. We will often take the then-part sets B_j to be scalar $(p = 1)$ for simplicity and with no loss of generality. The naive view of rule firing is that the input x_o fires the rule $R_{A_j \to B_j}$ when it "picks off" the if-part value $a_j(x_o)$ and then somehow affects the corresponding then-part set B_j to produce the x-fired version $B_j(x_o)$. Denote the input-fired jth rule as $R_{A_j \to B_j}(x_o)$. We show now that a rule fires formally when the rule convolves with an input.

Denote the functional form of the fuzzy-rule set function as $r_{A_j \to B_j} : \mathbb{R}^n \times \mathbb{R}^p \to [0, 1]$. A practical definition views the if-then rule "If $X = A_j$ then $Y = B_j$" as the Cartesian product $A_j \times B_j$ with multiplicative (not pairwise minimum) definition $r_{A_j \to B_j}(x, y) = a_j(x) b_j(y)$. We show that this product defines a *standard* additive model or SAM fuzzy system: $R_{A_j \to B_j}(x_o) = a_j(x_o)\ B_j$.

Define next the input x_o as the vector Dirac delta function $\delta(x - x_o)$. A simpler discrete version would denote the input x_o as a unit bit vector. Then the input x_o *fires* the jth rule $R_{A_j \to B_j}$ when the rule convolves with the delta-spike input $\delta(x - x_o)$ to produce the fired then-part set $B_j(x_o)$: $b_j(y|x_o) = r_{A_j \to B_j} * \delta(x - x_o)(y)$ for all $y \in \mathbb{R}^p$. This gives the key result that $b_j(y|x_o) = r_{A_j \to B_j}(x_o, y) = a_j(x_0)\ b_j(y)$. So it gives the fired then-part set $B_j(x_o)$ as the input-scaled then-part set $a_j(x_o)\ B_j$ because

$$
\begin{aligned}
b_j(y|x_o) &= \int_{\mathbb{R}^n} \delta(x - x_o)\ r_{A_j \to B_j}(x, y)\ dx && (1)\\
&= \int_{\mathbb{R}^n} \delta(x - x_o)\ a_j(x)\ b_j(y)\ dx && (2)\\
&= b_j(y) \int_{\mathbb{R}^n} \delta(x - x_o)\ a_j(x)\ dx && (3)\\
&= b_j(y)\ a_j(x_o) && (4)
\end{aligned}
$$



(a) Generalized mixture $p_n(y|x)$ from 10 Gaussian rules.

(b) Converged mixture $q_n(y|x)$ that mixes 2 normal curves.

(c) Generalized mixture $p_n(y|x)$ from 10 sinc rules.

(d) Converged mixture $q_n(y|x_0)$ that mixes 2 normal curves.

Fig. 2: Mixture representations for the adaptive fuzzy approximation of the bounded target function $f(x) = \sin x$ in Figure 1. The underlying fuzzy approximators $F_n(x)$ used 10 rules with either Gaussian or sinc if-part set functions. Panel (a) shows the generalized mixture $p_n(y|x)$ of the Gaussian fuzzy system after $n = 10,000$ epochs of learning. The mixture combined 10 Gaussian likelihood functions. Panel (b) shows the mixture $q_n(y|x)$ that Theorem 2 describes when the fuzzy system F_n directly defines the mixing weight of the two normal bell curves in Figure 1. Panels (c) and (d) show the related mixtures for an underlying fuzzy approximator F_n with 10 sinc rules. The final converged mixtures in (b) and (d) are indistinguishable from the generalized mixture $p(y|x)$ in Figure 1 that represents $f(x) = \sin x$ with two unit-variance normal densities.

for all y from the sifting property of the Dirac delta. We can rewrite the firing sequence (1) - (4) as the fuzzy set $R_{A_j \to B_j}(x_o) = a_j(x_o)\, B_j$.

The same argument extends to an input fuzzy set $A \subset \mathbb{R}^n$ that fires the rules if we replace the delta-spike convolution with the correlation $a_j(A) = \int_{\mathbb{R}^n} a(x)\; a_j(x)\; dx$ where a is the set function of A and where the integral exists [11]. This gives the fired rule's then-part set as $R_{A_j \to B_j}(A) = a_j(A)\; B_j$. The additive-model results below still hold in this more general case of a set-valued input but for simplicity we present only the common case of vector-valued inputs.

Each rule $R_{A_j \to B_j}$ also has a nonnegative rule weight $w_j \geq 0$. The weight can reflect a rule's credibility or relative importance and gives rise to a straightforward supervised learning law [11]. We define more generally an m-by-m matrix W of rule weights. We consider here only a diagonal W and thus ignore cross rules of the form $R_{A_i \to B_j}$ where $i \neq j$. Cross rules simply require an extra rule sum over i.

The vector input x_o fires all m rules $R_{A_j \to B_j}$ in parallel: $\phi(x_o) = \phi(R_{A_1 \to B_1}, \ldots, R_{A_m \to B_m})(x_o) = \phi(R_{A_1 \to B_1}(x_o), \ldots, R_{A_m \to B_m}(x_o))$ for some rule combination function ϕ and weight matrix W. This produces a final combined generalized set $B(x_o)$. An *additive* fuzzy system sums the weighted rules: $\phi(x_o) = \sum_{j=1}^{m} w_j R_{A_j \to B_j}(x)$. Then the convolution argument (1) - (4) gives the key result for the total rule firing given input x_o:

$$B(x_o) = \sum_{j=1}^{m} w_j a_j(x_o) B_j. \tag{5}$$

A "defuzzification" step converts the summed rule firings $B(x_o)$ into a final output $F(x_o)$ by taking the centroid of this generalized fuzzy set. So it does not matter that some or all values $b(y|x_o)$ can exceed unity. Earlier fuzzy systems appeared to have instead combined rules with pairwise maximum to prevent this [2]–[4] even though then the final result $B(x_o)$ would tend to approach a unit rectangle for large m. So the max-based combination technique would tend to give poorer performance as the size m of the rule base increased.

B. Generalized Mixtures from Additive Combination

The generalized mixture probability density $p(y|x)$ now follows from the additive combination $b(y|x_o)$ in (5). This follows from the nonnegativity of $b(y|x_o)$ and from its implied integrability: $p(y|x_o) = \frac{b(y|x_o)}{\int b(y|x_O)dy}$. So $p(y|x_o) \geq 0$ and $\int p(y|x_o)dy = 1$ for all x_o. Hence $p(y|x_o)$ defines a family of probability densities with index x_o.

A mixture density is a convex sum of densities. Define the finite positive volume $V_j(x)$ of the jth fired then-part set $B_j(x)$ as $V_j(x) = \int b_j(y|x) dy > 0$. Then the total-firing structure of the generalized set function $b(y|x_o)$ in (5) gives just such a convex sum:

$$p(y|x) = \frac{\sum_{j=1}^{m} w_j \; b_j(y|x)}{\int \sum_{k=1}^{m} w_k \; b_k(y|x) \; dy} \tag{6}$$
$$= \frac{\sum_{j=1}^{m} w_j \; V_j(x) p_{B_j}(y|x)}{\sum_{k=1}^{m} w_k \; V_k(x)} \tag{7}$$
$$= \sum_{j=1}^{m} \Big(\frac{w_j \; V_j(x)}{\sum_{k=1}^{m} w_k \; V_k(x)} \Big) p_{B_j}(y|x) \tag{8}$$
$$= \sum_{j=1}^{m} p_j(x) \; p_{B_j}(y|x) \tag{9}$$

for then-part probability density function $p_{B_j}(y|x) = \frac{b_j(y|x)}{V_j(x)}$ and the generalized mixing weight

$$p_j(x) = \frac{w_j \; V_j(x)}{\sum_{k=1}^{m} w_k \; V_k(x)} \; . \tag{10}$$

The SAM structure $b_j(y|x) = b_j(y) a_j(x)$ in (4) simplifies the mixture because then the then-part likelihoods p_{B_j} no longer depend on the input x: $p_{B_j}(y|x) = p_{B_j}(y)$. Assume that $a_j(x) > 0$ for simplicity as holds in a SAM with Gaussian or Cauchy if-part sets. Then $p_{B_j}(y|x) = \frac{b_j(y|x)}{V_j(x)} = \frac{a_j(x) b_j(y)}{a_j(x) V_j} = \frac{b_j(y)}{V_j} = p_{B_j}(y)$. Then

$$p(y|x) = \sum_{j=1}^{m} p_j(x) \; p_{B_j}(y) \tag{11}$$

with simplified generalized convex mixture coefficients

$$p_j(x) = \frac{w_j \; a_j(x) \; V_j}{\sum_{k=1}^{m} w_k \; a_k(x) \; V_k} \; . \tag{12}$$

The convex structure of the mixture $p(y|x)$ in (11) reflects that it is just the theorem on total probability from elementary probability. The mixture weights $p_j(x)$ define m generalized priors over the "hidden" random variable Z that takes values in $\{1, \ldots, m\}$: $p_j(x) = P(Z = j|X = x)$. The then-part densities $p_{B_j}(y)$ define m generalized likelihoods: $p_{B_j}(y) = P(Y = y|Z = j)$. So (11) gives a Bayesian posterior density p over the m rules given the input x and the observed output $y = F(x)$:

$$p(j|y, x) = \frac{p_j(x) \; p_{B_j}(y)}{\sum_{k=1}^{m} p_k(x) \; p_{B_k}(y)} \; . \tag{13}$$

This Bayesian posterior helps interpret neural black boxes when an adaptive fuzzy system trains on a classifier or other neural network and thereby converts the neural mapping to a fuzzy rule-based system [5].

C. Mixture Moments: Fuzzy Systems as Convex Sums

The generalized mixture $p(y|x)$ in (11) gives rise to uncountably many central and noncentral higher-order moments [1]. The first two integer moments are the most important because they give the respective conditional mean $E[Y|X = x]$ and the variance or covariance matrix $V[Y|X = x]$. The former moment defines the fuzzy system F itself. The latter defines its basic input-by-input uncertainty measure given its stored rules.

The ordinary SAM fuzzy system $F : \mathbb{R}^n \to \mathbb{R}^p$ of m rules $R_{A_j \to B_j}$ is just the first central moment of the system's generalized mixture $p(y|x)$ in (11):

$$F(x) = E[Y|X = x] = \int y \; p(y|x) \; dy \tag{14}$$
$$= \frac{\sum_{j=1}^{m} w_j \; a_j(x) \; V_j \; c_j}{\sum_{k=1}^{m} w_k \; a_k(x) \; V_k} \tag{15}$$
$$= \sum_{j=1}^{m} p_j(x) \; c_j \tag{16}$$

for then-part set centroids $c_j = \frac{\int y \; p_{B_j}(y) dy}{V_j}$. Most fuzzy applications use some version of this additive system [12], [13].

The convex sum of centroids in (16) follows naturally from the mixture $p(y|x)$. A direct calculation shows that it also follows in the usual but *ad hoc* way by taking the centroid of the fired then-part sets: $F(x) = \text{Centroid}(B(x))$.

The second non-central moment of $p(y|x)$ gives the conditional variance $V[Y|X = x]$ in the SAM case as

$$V[Y|X = x] = \sum_{j=1}^{m} p_j(x) \; \sigma^2_{B_j} + \sum_{j=1}^{m} p_j(x) \; [c_j - F(x)]^2 \tag{17}$$

for then-part-set variance $\sigma^2_{B_j} = \int (y - c_j)^2 p_{B_j}(y) \; dy$ in the scalar case of $F : \mathbb{R}^n \to \mathbb{R}$. The first term on the right-hand side describes the uncertainty due to the size and structure of the m then-part sets B_j. The second term penalizes the output $F(x)$ based on how much the system interpolates over its m rules. This conditional variance defines an uncertainty surface over the input space. It can help interpret misclassifications when the fuzzy system approximates a neural classifier [5], [6].

III. Mixture Representations of Functions with Watkins Coefficients

We next state and prove a simpler and more practical version of Theorem 4 in [1]. The next section uses this result to prove the main mixture uniform-convergence result.

Suppose the real function $f : X \to \mathbb{R}$ is bounded and not constant. The input vector space X is $\mathbb{R}^n$ in practice but can have infinite dimension. The output $f(x)$ can also be a point in $\mathbb{R}^p$ so long as each component function is bounded and not constant. Let α denote the infimum of the bounded real function f: $\alpha = \inf_{x \in X} f(x)$. Let β denote the supremum: $\beta = \sup_{x \in X} f(x)$. Then $\alpha < \beta$ because f is not constant.

We will use two normal bell curves to form a canonical mixture representation of the bounded function f. The trick is to center the bell curves over the infimum α and supremum β. The corresponding variances $\sigma^2_\alpha > 0$ and $\sigma^2_\beta > 0$ can be arbitrary finite positive values for purposes of the next theorem. The two variances are unity in Figure 1(a) where the two likelihood normal curves have respective centers $\alpha = -1$ and $\beta = 1$. Denote these basis-like bell curves as $N_\alpha(y|\alpha, \sigma_\alpha^2)$ and $N_\beta(y|\beta, \sigma_\beta^2)$. So the normal density n_α has the form

$$n_\alpha(y) = \frac{1}{\sqrt{2\pi\sigma_\alpha^2}} \exp\left[-\frac{1}{2}\left(\frac{y-\alpha}{\sigma_\alpha}\right)^2\right] \tag{18}$$

and thus $n_\alpha(y) = N_\alpha(y|\alpha, \sigma_\alpha^2)$.

The last step defines the convex *Watkins coefficients* $w(x) \geq 0$ and $1 - w(x) \geq 0$ that control the mixture representation of the bounded real function f:

$$w(x) = \frac{\beta - f(x)}{\beta - \alpha} \tag{19}$$

$$1 - w(x) = \frac{f(x) - \alpha}{\beta - \alpha} \tag{20}$$

for infimum α and supremum β. Then $0 \leq w(x) \leq 1$ holds and so $w(x)$ and $1-w(x)$ define generalized mixture weights. The weights are differentiable or integrable with respect to variable x if and only if f is.

Watkins first showed that a SAM additive fuzzy system F can exactly represent any bounded nonconstant real function $f : (R) \to \mathbb{R}$ with just *two* rules if it uses w and $1-w$ for the respective if-part set functions of the two rules: $F(x) = f(x)$ for all $x \in \mathbb{R}$ [7], [8]. This Watkins Representation Theorem is a much stronger result than the uniform fuzzy approximation theorem that says some additive fuzzy system with a finite number m of rules can uniformly approximate any continuous function on a compact set [14]–[16]. But the Watkins result requires that the user know the bounded function f in advance and not just approximate it from data.

The Watkins representation is quite useful when we do know a closed-form bounded function and want to represent it in a rule base. This holds for almost all closed-form probability densities in Bayesian analysis [17] as well as many other math models and even many trained neural systems. The next theorem shows that a generalized Gaussian mixture $p(y|x)$ can also absorb such an f by mixing just the two likelihood densities $N_\alpha(y|\alpha, \sigma_\alpha^2)$ and $N_\beta(y|\beta, \sigma_\beta^2)$.

Theorem 1: Generalized Gaussian Mixture Representation of Bounded Functions.

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is bounded and not a constant with $\alpha = \inf_{x\in X} f(x)$ and $\beta = \sup_{x \in X} f(x)$. Suppose that the generalized mixture density $p(y|x)$ mixes two normal probability density functions with Watkins coefficients (19):

$$p(y|x) = w(x)N_\alpha(y|\alpha, \sigma_\alpha^2) + (1 - w(x))N_\beta(y|\beta, \sigma_\beta^2) \ . \tag{21}$$

Then $p(y|x)$ represents f exactly on average: $E[Y|X = x] = f(x)$ for all x. In particular: $F(x) = f(x)$ for all x for the additive SAM system F in (16).

Proof : Substitute the Watkins coefficients in (19) - (20) into $p(y|x)$ and then integrate and use (18):

$$\begin{aligned} E[Y|X=x] &= \int y\ p(y|x)\ dy & (22)\\ &= (\frac{\beta - f(x)}{\beta-\alpha}) \int y\ n_\alpha(y)\ dy \\ &\quad + (\frac{f(x)-\alpha}{\beta-\alpha}) \int y\ n_\beta(y)\ dy & (23)\\ &= (\frac{\beta - f(x)}{\beta-\alpha})\alpha + (\frac{f(x)-\alpha}{\beta-\alpha})\beta & (24)\\ &= \frac{f(x)[\beta-\alpha]}{\beta-\alpha} & (25)\\ &= f(x)\ . & (26)\end{aligned}$$

This follows because the centroid of a normal random variable $Y \sim N_\alpha(y|\alpha, \sigma_\alpha^2)$ is just its location parameter α and similarly if $Y \sim N_\beta(y|\beta, \sigma_\beta^2)$. **Q.E.D.**

Figure 1(c) shows this generalized Gaussian-mixture representation of $f(x) = \sin x$ over a 2π interval of its domain in Figure 1(a). The mixture has the form

$$p(y|x) = \frac{1-\sin x}{2}N(y|-1,1) + \frac{\sin x + 1}{2}N(y|1,1)\ . \tag{27}$$

The green curve in Figure 1(b) shows the Gaussian-mixture slice $p(y|3.41)$ for input $x_o = 3.41$.

Theorem 1 holds for any two likelihood densities that have the infimum α and the supremum β as their respective centroids. It holds approximately for linear or multiplicative perturbations of α and β as may occur in practice when one estimates the function f with data as in [1].

The theorem also reduces to the original Watkins 2-rule case if the likelihood variances σ_α and σ_β go to zero. Then the normal bell curves become Dirac delta functions:

$$p(y|x) = w(x)\delta(y-\alpha) + (1-w(x))\delta(y-\beta)\ . \tag{28}$$

This result reflects the earlier practice of simply defining the fuzzy system's rule then-part sets B_j as their centroids c_j [18]. This strong assumption removes all uncertainty from the then-part portions of the rules but does not affect function approximation to first order. It does affect the system's second-order uncertainty in accord with the first term on the right-hand side of (17).

The canonical Gaussian-mixture representation (21) also implies an integral representation for the k-th derivative $f^{(k)}$ of the bounded function f if the derivative exists:

$$\begin{aligned} f^{(k)}(x) &= \int \frac{d^k p(y|x)}{dx^k} dy & (29)\\ &= w^{(k)}(x)\int y\ n_\alpha(y)dy - w^{(k)}(x)\int y\ n_\beta(y)dy & (30)\end{aligned}$$

because $w^{(k)}(x) = -\frac{f^{(k)}(x)}{\beta-\alpha}$.

IV. Mixture Approximation Theorem

We can now present the main result: The approximator mixture $q_n(y|x)$ converges uniformly in x *and* in y to the canonical generalized Gaussian mixture $p(y|x)$ if the fuzzy or other function approximator F_n converges uniformly to the bounded non-constant function f. The theorem holds for any uniform function approximator F_n of f. Figure 2 shows that a user can simply plug an adaptive fuzzy system F_n into the generalized Gaussian-mixture framework to produce such a uniformly converging sequence of Gaussian mixtures $q_n(y|x)$.

We start with the definition of uniform convergence. The sequence of functions F_n converges uniformly in x to f if for all $\epsilon > 0$ there exists a positive integer n_0 such that for all integers $n \geq n_0$: $|F_n(x) - f(x)| < \epsilon$ for all $x \in X$.

Let $v_n(x)$ and $1 - v_n(x)$ define the approximator Watkins coefficients that result from replacing the target function f with the uniform approximator F_n:

$$v_n(x) = \frac{\beta - F_n(x)}{\beta - \alpha} \tag{31}$$

$$1 - v_n(x) = \frac{F_n(x) - \alpha}{\beta - \alpha} \tag{32}$$

such that $\alpha \leq F_n \leq \beta$ for all n. Define the *approximating Gaussian mixture* $q_n(y|x)$ as

$$q_n(y|x) = v_n(x)N_\alpha(y|\alpha, \sigma_\alpha^2) + (1 - v_n(x))N_\beta(y|\beta, \sigma_\beta^2)\ . \tag{33}$$

Then the next theorem shows that $q_n(y|x)$ converges uniformly to the generalized Gaussian mixture $p(y|x)$. This means that for all $\epsilon > 0$ there exists a positive integer n_0 such that for all integers $n \geq n_0$: $|q_n(y|x) - p(y|x)| < \epsilon$ for all x and for all y.

Theorem 2: Uniform Gaussian Mixture Convergence.

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is bounded and not a constant with $\alpha = \inf_{x\in X} f(x)$ and $\beta = \sup_{x\in X} f(x)$. Suppose that the generalized mixture density $p(y|x)$ mixes two normal probability density functions with Watkins coefficients (31):

$$p(y|x) = w(x)N_\alpha(y|\alpha, \sigma_\alpha^2) + (1 - w(x))N_\beta(y|\beta, \sigma_\beta^2)\ . \tag{34}$$

Suppose that F_n uniformly approximates f and that $\alpha \leq F_n \leq \beta$ for all n. Define the approximating mixture $q_n(y|x)$ with (31) - (33). Then the generalized Gaussian mixture $q_n(y|x)$ converges uniformly in x and y to the generalized Gaussian mixture $p(y|x)$:

$$\lim_{n\to\infty} q_n(y|x) = p(y|x)\ . \tag{35}$$

Proof : Pick any y. Then Figure 1(b) shows that the distance between the normal-curve values $n_\alpha(y)$ and $n_\beta(y)$ cannot exceed the larger of the two mode values $n_\alpha(\alpha)$ and $n_\beta(\beta)$:

$$|n_\alpha(y) - n_\beta(y)| \leq \max(n_\alpha(\alpha), n_\beta(\beta)) \tag{36}$$

$$< \max(n_\alpha(\alpha), n_\beta(\beta)) + 1 \equiv c\ . \tag{37}$$

Suppose that F_n converges uniformly to f. Then for all $\epsilon > 0$ there exists a positive integer n_0 such that for all integers $n \geq n_0$: $|F_n(x) - f(x)| < \frac{\beta-\alpha}{c}\epsilon$ for all $x \in X$ since f is bounded and not constant. Then for all such large $n \geq n_0$:

$$\begin{aligned}|q_n(y|x) - p(y|x)| &= \frac{1}{\beta - \alpha}|(\beta - F_n(x))n_\alpha(y) \\ &\quad + (F_n(x) - \alpha)n_\beta(y) - (\beta - f(x))n_\alpha(y) \\ &\quad - (f(x) - \alpha)n_\beta(y)| \qquad (38) \\ &= \frac{1}{\beta - \alpha}|F_n(x)(n_\beta(y) - n_\alpha(y)) \\ &\quad - f(x)(n_\beta(y) - n_\alpha(y))| \qquad (39) \\ &= \frac{1}{\beta - \alpha}|F_n(x) - f(x)||n_\alpha(y) - n_\beta(y)| \qquad (40) \\ &< \frac{c}{\beta - \alpha}|F_n(x) - f(x)| \qquad (41) \\ &< \frac{c}{\beta - \alpha}\frac{\beta - \alpha}{c}\epsilon \qquad (42) \\ &< \epsilon \qquad (43)\end{aligned}$$

for all x and all y. So $q_n(y|x)$ converges uniformly to $p(y|x)$. **Q.E.D.**

A more general result holds: The approximator F_n can obey the bounds $\alpha_n \leq F_n \leq \beta_n$ so long as $\lim_{n\to\infty} \alpha_n = \alpha$ and $\lim_{n\to\infty} \beta_n = \beta$. The new bounds α_n and β_n now replace α and β in the modified Watkins coefficients $v_n(x)$ and $1 - v_n(x)$ in (31) and (32). Then the uniform convergence in (35) still holds because the mixed normals' exponentials and quadratics are continuous. So the limits pass through without incident.

A. Convergent Mixtures with Adaptive Fuzzy Approximators

The mixture convergence in (35) allows the user to insert *any* uniform approximator F_n in (31) - (33) to produce a corresponding sequence of generalized Gaussian mixtures $q_n(y|x)$.

A classic example in real analysis of a uniform approximator is the sequence of Bernstein polynomials B_n on the unit interval that uniformly approximates a continuous function $f : [0, 1] \to \mathbb{R}$ [19]. A Bernstein polynomial B_n takes $n + 1$ uniform samples of f and forms a binomial average:

$$B_n(x) = \sum_{k=0}^{n} f(\frac{k}{n})\binom{n}{k}x^k(1 - x)^{n-k} \tag{44}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Suppose that $\alpha = 0$ and $\beta = 1$ for f although in general these values may differ. The extreme value theorem ensures that the continuous function f attains its minimum α and maximum β on $[0, 1]$ because the unit interval is compact. Then put $v_n(x) = 1 - B_n(x)$ and $1 - v_n(x) = B_n(x)$. This gives the uniformly converging mixture sequence $q_n(y|x)$:

$$q_n(y|x) = (1 - B_n(x))N(y|0, 1) + B_n(x)N(y|1, 1)\ . \tag{45}$$

The basic uniform approximation theorem for additive fuzzy systems [11], [14], [20] states that additive fuzzy systems F

are dense in the space of continuous function f on a compact set. These and related theorems [15], [16] do not actually exhibit a uniformly convergent sequence of additive fuzzy systems F_n as in the case of the Bernstein polynomial (44). We often assume that efficient supervised learning produces such convergent sequences. The same holds with the basic theorems on neural universal approximation: Multilayer neural networks N with a large but finite number of hidden logistic sigmoid neurons are dense in the space of continuous functions [21], [22]. We often assume that a successfully trained neural classifier or regressor defines such a predicted uniformly convergent sequence of neural networks N_n for the nth training epoch.

An adaptive SAM system F did learn the sine wave in Figure 1(a). It used 10 Gaussian rules and fully converged to the target function $f(x) = \sin x$ after just a few iterations. The scalar if-part sets had the form

$$a_j(x) = \exp[-\Big(\frac{x - m_j}{d_j}\Big)^2] \tag{46}$$

for the respective mean m_j and dispersion d_j. This Gaussian set function gives rise to the supervised learning laws [11], [17], [23]

$$m_j(n+1) = m_j(n) + \mu_n \varepsilon(x) p_j(x)[c_j - F_n(x)]\frac{x - m_j(n)}{d_n^2(x)} \tag{47}$$

$$d_j(n+1) = d_j(n) + \mu_n \varepsilon(x) p_j(x)[c_j - F_n(x)]\frac{(x - m_j(n))^2}{d_n^3(x)} \tag{48}$$

where μ_n is a decreasing sequence of learning constants. We often use a linearly decreasing sequence in practice both because of its effectiveness and because of its robust status in the theory of stochastic approximation [24]. The error ε is the desired-minus-actual difference $\varepsilon(x) = f(x) - F_n(x)$. The result produces the 10-rule SAM sequence F_n that rapidly converges to $f(x) = \sin x$.

Figure 2(a) shows the generalized mixture $p_n(y|x)$ that results from using this SAM's coefficients in (7) - (12) after $n = 10,000$ iterations or learning epochs. Figure 2(b) shows the generalized mixture $q_n(y|x)$ that results from inserting this Gaussian SAM F_n into (31) - (33) to rapidly approximate the generalized mixture $p(y|x)$ in Figure 1(c) that represents the target function $f(x) = \sin x$.

The sinc wavelet often performs better as an if-part set than does a Gaussian or other set structure [17], [23], [25]. The sinc if-part set function a_j^k has center m_j^k and has dispersion or width d_j^k:

$$a_j^k(x^k) = \sin\big(\frac{x^k - m_j^k}{d_j^k}\big) / \big(\frac{x^k - m_j^k}{d_j^k}\big)\ . \tag{49}$$

The sinc set function is neither monotonic nor unimodal. But it still defines a working generalized set function even though it takes values in $[-.217, 1]$ (the software must zero-out the occasional negative value that this can produce in the mixing weights). The sinc set function gives rise to the adaptive SAM learning laws

$$\begin{aligned} m_j^k(t+1) &= m_j^k(t) \\ &\quad - \mu_t \varepsilon_t \frac{p_j(x)}{a_j^k(x^k)}[c_j - F(x)] \\ &\quad \times \big(a_j^k(x^k) - \cos\big(\frac{x^k - m_j^k}{d_j^k}\big)\big)\frac{1}{x^k - m_j^k} \end{aligned} \tag{50}$$

$$\begin{aligned} d_j^k(t+1) &= d_j^k(t) \\ &\quad - \mu_t \varepsilon_t \frac{p_j(x)}{a_j^k(x^k)}[c_j - F(x)] \\ &\quad \times \big(a_j^k(x^k) - \cos\big(\frac{x^k - m_j^k}{d_j^k}\big)\big)\frac{1}{d_j^k}\ . \end{aligned} \tag{51}$$

Figure 2(c) shows a 10-rule sinc SAM F_n after $n = 10,000$ iterations. Figure 2(d) shows the generalized mixture $q_n(y|x)$ that also results from inserting this sinc SAM F_n into (31) - (33) to rapidly approximate the generalized mixture $p(y|x)$ in Figure 1(c) that represents the target function $f(x) = \sin x$. The converged mixtures in panels (b) and (d) are indistinguishable from $p(y|x)$.

Converging neural networks N_n likewise define a sequence of generalized mixtures $q_n(y|x)$ by replacing F_n with N_n in the Watkins coefficients (31) - (33). The K output neurons are bounded softmax or logistic neurons in a typical deep neural network [26]–[29]. Then the above results apply componentwise in this vector-valued case.

Mixtures can also represent the bounded dynamical models in some control systems [12], [13], [30]. The vector approach further allows mixtures to apply to nonlinear feedback networks such as fuzzy cognitive maps [31], [32] used in causal modeling [33]–[35] because the concept nodes are bounded.

V. Conclusion

A sequence of converging fuzzy or neural systems defines a sequence of converging generalized probability mixtures. The convergence is uniform both in the input and in the output variables. These convergent mixtures completely describe the fuzzy system and all its higher-order moments. They also give a new statistical description of neural networks. They allow users to naturally combine any number of rule bases into a single representative rule base because mixing mixtures results in a new mixture. Then this hierarchical mixture's own Bayes theorem gives a complete posterior-density probability description of the firing of the system's rule-based subsystems and the firing of the if-then rules in each subsystem.

References

[1] B. Kosko, "Additive fuzzy systems: From generalized mixtures to rule continua," *International Journal of Intelligent Systems*, vol. 33, no. 8, pp. 1573–1623, 2018.

[2] T. Terano, K. Asai, and M. Sugeno, *Fuzzy systems theory and its applications.* Academic Press Professional, Inc., 1992.

[3] H.-J. Zimmermann, *Fuzzy set theory and its applications.* Springer Science & Business Media, 2011.

[4] M. Sugeno, "An introductory survey of fuzzy control," *Information sciences*, vol. 36, no. 1, pp. 59–83, 1985.

[5] A. K. Panda and B. Kosko, "Converting neural networks to rule foam," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2019, pp. 519–525.

[6] ——, "Random fuzzy-rule foams for explainable AI," in *Fuzzy Information Processing 2020, Proceedings of NAFIPS-2020*. Springer, 2020.

[7] F. A. Watkins, "Fuzzy engineering," Ph.D. dissertation, University of California at Irvine, 1994.

[8] F. Watkins, "The representation problem for additive fuzzy systems," in *Proceedings of the International Conference on Fuzzy Systems (IEEE FUZZ-95)*, 1995, pp. 117–122.

[9] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[10] R. Bellman, R. Kalaba, and L. Zadeh, "Abstraction and pattern classification," *Journal of Mathematical Analysis and Applications*, vol. 13, no. 1, pp. 1–7, 1966.

[11] B. Kosko, *Fuzzy Engineering*. Prentice Hall, 1996.

[12] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *Fuzzy systems, IEEE Transactions on*, vol. 14, no. 5, pp. 676–697, 2006.

[13] A.-T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, "Fuzzy control systems: Past, present and future," *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 56–68, 2019.

[14] B. Kosko, "Fuzzy Systems as Universal Approximators," *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1329–1333, November 1994.

[15] V. Kreinovich, G. C. Mouzouris, and H. T. Nguyen, "Fuzzy rule based modeling as a universal approximation tool," in *Fuzzy Systems*. Springer, 1998, pp. 135–195.

[16] V. Kreinovich, H. T. Nguyen, and Y. Yam, "Fuzzy systems are universal approximators for a smooth function and its derivatives," *International Journal of Intelligent Systems*, vol. 15, no. 6, pp. 565–574, 2000.

[17] O. Osoba, S. Mitaim, and B. Kosko, "Bayesian inference with adaptive fuzzy priors and likelihoods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 5, pp. 1183 –1197, Oct. 2011.

[18] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," *IEEE Transactions on Fuzzy systems*, vol. 7, no. 2, pp. 201–224, 1999.

[19] P. L. Duren and P. Duren, *Invitation to classical analysis*. American Mathematical Soc., 2012, vol. 17.

[20] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, 1991.

[21] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[22] K. Hornik, M. Stinchcombe, H. White *et al.*, "Multilayer feedforward networks are universal approximators." *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[23] S. Mitaim and B. Kosko, "The shape of fuzzy sets in adaptive function approximation," *IEEE Transactions on fuzzy systems*, vol. 9, no. 4, pp. 637–656, 2001.

[24] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[25] Q. Luo, W. Yang, and D. Yi, "Kernel shapes of fuzzy sets in fuzzy systems for function approximation," *Information sciences*, vol. 178, no. 3, pp. 836–857, 2008.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[27] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[28] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[29] O. Adigun and B. Kosko, "Noise-boosted bidirectional backpropagation and adversarial learning," *Neural Networks*, vol. 120, pp. 9–31, 2019.

[30] H. Hamdi, C. B. Regaya, and A. Zaafouri, "A sliding-neural network control of induction-motor-pump supplied by photovoltaic generator," *Protection and Control of Modern Power Systems*, vol. 5, no. 1, p. 1, 2020.

[31] B. Kosko, "Fuzzy cognitive maps," *International journal of man-machine studies*, vol. 24, no. 1, pp. 65–75, 1986.

[32] O. A. Osoba and B. Kosko, "Fuzzy cognitive maps of public support for insurgency and terrorism," *The Journal of Defense Modeling and Simulation*, vol. 14, no. 1, pp. 17–32, 2017.

[33] G. Ziv, E. Watson, D. Young, D. C. Howard, S. T. Larcom, and A. J. Tanentzap, "The potential impact of Brexit on the energy, water and food nexus in the uk: A fuzzy cognitive mapping approach," *Applied energy*, vol. 210, pp. 487–498, 2018.

[34] M. Glykas, *Fuzzy cognitive maps: Advances in theory, methodologies, tools and applications*. Springer, 2010, vol. 247.

[35] E. I. Papageorgiou, *Fuzzy cognitive maps for applied sciences and engineering: from fundamentals to extensions and learning algorithms*. Springer Science & Business Media, 2013, vol. 54.