Fuzzy Aggregation Techniques in Situations Without Experts: Towards A New Justification

Hung T. Nguyen
Department of Mathematical Sciences
New Mexico State University
Las Cruces, New Mexico, 88003, USA

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
El Paso, Texas 79968, USA
contact email vladik@utep.edu

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

1. Brief Outline

- *Original idea:* fuzzy techniques were invented to describe expert knowledge.
- *Successes:* there are many successful applications of fuzzy methodology to situations with expert knowledge.
- *Situation:* fuzzy methodology is sometimes successful even when no expert knowledge exists.
- *What we plan to do:* give a mathematical explanation for this phenomenon.

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

2. Detailed Introduction

- *Origins of fuzzy techniques – a brief reminder:* they were invented to translate the knowledge of experts:
  - expert knowledge is often formulated in terms of natural language;
  - we need a precise computer-implementable form.
- *Expected application:* formalizing rules of human operators in intelligent control.
- *Unexpected application:* fuzzy techniques often work well without expert knowledge.
- *Question:* why?

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

3. Formulation of the Problem

- *Situation:* fuzzy techniques often work without expert knowledge.
- *Usual explanation – universal approximation result:*
  - for every continuous function $f(x_1, \ldots, x_n)$ on a bounded set, and for every $\varepsilon > 0$,
  - there exists a set of rules for which the corresponding input-output function is ε-close to f.
- *Comment:* this result explains why fuzzy techniques work.
- *Challenge:* polynomials or splines also have universal approximation property.
- *Question:* why are fuzzy methods *better* in many practical situations?

Our Main Idea: ...
Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments

4. Our Main Idea: In Many Practical Applications, Data Processing Speed Is Important

- *One of the main applications of fuzzy methodology:* intelligent control.
- *Specific demand of control:* in many practical problems, we need to produce a control value very fast.
- *Example:* before the chemical reaction deviates seriously veers off course).
- *Parallel computing is an answer:* a natural way to increase the computations speed is to perform computations *in parallel* on several processors.
- *Fastest possible parallelization* is when we divide the algorithm into parallelizable steps, each of which requires the smallest possible amount of computation time.

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

5. Towards the Fastest Control-Oriented Parallel Computer

- *Fact:* control algorithms are computationally complicated since they process many inputs.
- *Example:* a plane travels in 3-D, so we need more parameters to measure and to process than for the car.
- *Conclusion:* the fewer inputs the faster.
- *Ideal case:* every processor processes only one input.
- *Problem:* a composition of functions of 1 variable is still a function of one variable.
- *Conclusion:* we need functions of 2 variables.
- *Fact:* directly hardware supported operations are the fastest: $a+b$, $a-b$, $a \cdot b$, a/b, $\min(a,b)$, and $\max(a,b)$.
- *Question:* which functions should we choose?

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

6. Toward Fastest Processors

- *We have already argued that:* processors should compute functions of 1 variable and hardware-supported functions of 2 variables.
- *Fact:* directly hardware supported operations are the fastest: $a+b$, $a-b$, $a \cdot b$, a/b, $\min(a,b)$, and $\max(a,b)$.
- *Comparison:*
  - a/b is done by successive multiplication and $-$, so it is slower than $-$;
  - $a \cdot b$ consists of many $+$, so it is slower than $+$;
  - $+$ of two n-bit numbers means $2n$ additions: n for bits, n for adding carries;
  - min and max require only $\le n$ bit operations.
- *Conclusion:* min and max are fastest.

Our Main Idea: If . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

7. How to Combine These Processors?

- *What we have done:* we have selected processors.
- *What we still need to do:* combine these processors so that the resulting computations be as fast as possible.
- *Idea:* computation time is crudely proportional to the number of sequential steps (layers):
  - at the beginning, the input numbers are processed by some processors; these processors form the *first layer* of computations;
  - the results of this processing may then go into different processors, that form the *second layer*;
  - the results of the second layer of processing go into the *third layer*, etc.
- *Conclusion:* the fewer layers the computer has, the faster it is.

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

8. Formal Definition

- We say that a function $f(x_1, \ldots, x_n)$ is *computable by a 1-layer computer* if either $n = 1$, or the function f coincides with min or with max.
- Let $k \ge 1$ be an integer. We say that a function $f(x_1, \ldots, x_n)$ is *computable by a* $(k+1)$*-layer computer* if one of the following three statements is true:
  - $f = g(h(x_1, \ldots, x_n))$, where g is a function of one variable, and $h(x_1, \ldots, x_n)$ is computable by a k-layer computer;
  - $f = \min(g_1(x_1, \ldots, x_n), \ldots, g_l(x_1, \ldots, x_n))$, where all functions g_i are computed by a k-layer computer;
  - $f = \max(g_1(x_1, \ldots, x_n), \ldots, g_l(x_1, \ldots, x_n))$, where all functions g_i are computed by a k-layer computer.

Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments
Title Page

9. Results

- **Proposition.** $\exists T, \varepsilon > 0$, *and* $\exists$ *continuous* $f : [-T, T]^n \to R$ *such that no function* ε*-close to* f *on* $[-T, T]^n$ *can be computed on a 2-layer computer.*
- **Theorem.** $\forall\ T$ *and* $\varepsilon > 0$, *and* $\forall$ *continuous* $f : [-T, T]^n \to R$, $\exists$ *a function* $\widetilde{f}$ *that is* ε*-close to* f *on* $[-T, T]^n$ *and that is computable on a 3-layer computer.*
- *Mathematical comment.* In other words:
  - functions computed by a 3-layer computer are universal approximators, but
  - 2 layers are not enough.
- *Conclusion:* the fastest possible parallel computations are performed by a 3-layer computer.
- *What we will show:* that the corresponding functions are exactly fuzzy control functions.

Our Main Idea: ...
Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments
Title Page

10. Relation to Fuzzy Control

- *According to the proof:* $\widetilde{f}$ is of the type $\max(A_1, \ldots, A_k)$, where $A_j = \min(f_{j1}(x_1), \ldots, f_{jn}(x_n))$.
- *Auxiliary quantities:* $U = \max\limits_{i,j,x_i \in [-T,T]} |f_{ji}(x_i)|$, and

$$\mu_{ji}(x_i) = \frac{f_{ji}(x_i) - (-U)}{U - (-U)}.$$

- *Rules:*
  - "if $C_{j1}(x_1)$ & $\ldots$ & $C_{jn}(x_n)$ then $u = U$"; $(j = 1, \ldots, n)$
  - "else, $u = -U$".
- So, $u = U \leftrightarrow (C_{11} \& \ldots \& C_{1n}) \vee \ldots \vee (C_{k1} \& \ldots \& C_{kn})$.
- If we use min for &, and max for $\vee$, then $\mu_u(U) =$

$\max[\min(\mu_{11}(x_1), \ldots, \mu_{1n}(x_n)), \ldots, \min(\mu_{k1}(x_1), \ldots, \mu_{kn}(x_n))]$.

and $\mu_u(-U) = 1 - \mu_u(U)$.
- Defuzzification leads to $u = \widetilde{f}(x_1, \ldots, x_n)$.

Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments

11. Conclusions

- *Conclusion:* for control problems, the fastest possible universal computation scheme corresponds to using fuzzy methodology.
- *This is exactly what we wanted:* this result explains why fuzzy methodology is sometimes successfully used without any expert knowledge.
- *Comments.*
  - We have considered *digital* parallel computers.
  - If we use *analog* processors instead, then min and max stop being the simplest functions.
  - Instead, the sum is the simplest: joining two wires.
  - For sum (and linear combination) instead of min and max, 3-layer computers are also universal approximators – corresponding to *neural networks*.

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

12. Proof of the Proposition: Main Idea

- *What we prove:* 2-layer computers are not sufficient.
- *Specific example:* sum $f(x_1, x_2) = x_1 + x_2$ cannot be approximated by 2-layer computers.
- *Precise formulation:*
  - if a function $\widetilde{f}(x_1, x_2)$ is 0.4−close to f on $[-1, 1]^2$,
  - then $\widetilde{f}$ cannot be computed on a 2-layer computer.
- *Method of proof:* reduction to a contradiction.
- *Idea:* if $\widetilde{f}(x_1, x_2)$, then:
  - either $g(h(x_1, x_2))$, where h is 1-layer computable;
  - or $\min(g_1(x_1, x_2), \ldots, g_k(x_1, x_2))$, where all g_i are 1-layer computable;
  - or $\max(g_1(x_1, x_2), \ldots, g_k(x_1, x_2))$.
- We will show case-by-case that all these three cases are impossible.

Our Main Idea: In ...
Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments

13. Proof of The Proposition: Case 1

- *Situation:* $\widetilde{f}$ is 0.4-close to $x_1 + x_2$ and $\widetilde{f}(x_1, x_2) = g(h(x_1, x_2))$, where h is computable on a 1-layer computer.
- *Possible subcases:* h is a function of one variable $h(x_1)$, min, or max.
- *Subcase 1:*
  - $\widetilde{f}(x_1, x_2) = g(h(x_1))$, hence $\widetilde{f}(0, -1) = \widetilde{f}(0, 1)$;
  - however, $\widetilde{f}(0, -1)$ is 0.4-close to $0 + (-1) = -1$, so $\widetilde{f}(0, -1) \le -0.6$;
  - similarly, $\widetilde{f}(0, -1) \ge 0.6$ – a contradiction.
- *Subcase 2:* $\widetilde{f}(x_1, x_2) = g(\min(x_1, x_2))$ hence $\widetilde{f}(-1, -1) = \widetilde{f}(-1, 1)$ – also a similar contradiction.
- *Subcase 3:* $\widetilde{f}(x_1, x_2) = g(\max(x_1, x_2))$ hence $\widetilde{f}(1, 1) = \widetilde{f}(-1, 1)$ – a similar contradiction.

Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

14. Proof of The Proposition: Case 2

- *Situation:* $\widetilde{f}$ is 0.4-close to $x_1 + x_2$ and
$$\widetilde{f}(x_1, x_2) = \min(g_1(x_1, x_2), \ldots, g_k(x_1, x_2)),$$
where all g_i are 1-layer computable.
- *Comment:* Case 3 (with max) is similar.
- *Subcase 1:* if $g_i = \min(x_1, x_2)$, we can replace it with two terms x_1 and x_2.
- *Conclusion:* every g_i is either max or a function of 1 variable.
- *Subcase 2:* one of g_i is max. Then:
  - $\max(1,1) = 1$ hence $\widetilde{f}(1,1) = \min(\ldots, 1, \ldots) \le 1$;
  - but $\widetilde{f}(1,1)$ must be 0.4-close to $1 + 1 = 2$ hence $\widetilde{f}(1,1) \ge 1.6$.
- *Remaining subcase:* all g_i are functions of 1 variable.

Our Main Idea: ...
Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments
Title Page

15. Proof of The Proposition: Case 2 (cont-d)

- *Situation:* $\widetilde{f}$ is 0.4-close to $x_1 + x_2$ and $\widetilde{f}(x_1, x_2) = \min(g_1, \ldots, g_k)$, where all g_i are functions of 1 variable.
- Some g_i depend on x_1, some on x_2.
- So, we get $\widetilde{f}(x_1, x_2) = \min(h_1(x_1), h_2(x_2))$.
- For $x_1 = x_2 = 1$, we get $\min(h_1(1), h_2(1)) \ge 1.6$, hence $h_1(1) \ge 1.6$.
- For $x_1 =$ and $x_2 = -1$, we get $\min(h_1(1), h_2(-1)) \le 0.4$.
- Since $h_1(1) \ge 1.6$, we have $\min(h_1(1), h_2(-1)) = h_2(-1)$.
- So, from $\min(h_1(1), h_2(-1)) \ge -0.4$, we conclude that $h_2(-1) \ge -0.4$. Similarly, $h_1(-1) \ge -0.4$.
- So, $\widetilde{f}(-1, -1) = \min(h_1(-1), h_2(-1)) \ge -0.4$.
- Thus, $\widetilde{f}(-1, -1)$ is not 0.4-close to $(-1) + (-1) = -2$.

Our Main Idea: In . . .
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

16. Proofs of the Theorem

- Since f is continuous, $\exists\,\delta > 0$ such that if $|x_i - y_i| \le \delta$, then $|f(x_1, \ldots, x_n) - f(y_1, \ldots, y_n)| \le \varepsilon$.
- Pick points $P_j = (x_{j1}, \ldots, x_{jn})$ on a grid of linear size δ.
- Let $m = \min\{f(x_1, \ldots, x_n) : x_i \in [-T, T]\}$.
- Let $f_{ji}(x_i)$ be a small cone centered in x_{ji}:
  - $f_{ji}(x_i) = m$ when $|x_i - x_{ji}| \ge 0.7 \cdot \delta$;
  - $f_{ji}(x_i) = f(P_j)$ when $|x_i - x_{ji}| \le 0.6 \cdot \delta$;
  - with linear extrapolation in between.
- Then, $\widetilde{f}(x_1, \ldots, x_n) = \max(A_1, \ldots, A_k)$, where

  $$A_j = \min(f_{j1}(x_1), \ldots, f_{jn}(x_n)),$$

  is ε-close to the function $f(x_1, \ldots, x_n)$.
- Indeed, $\widetilde{f}(P_j) = f(P_j)$ for every j.
- One can show that $\widetilde{f}(x)$ and $f(x)$ are ε-close for all x.

Our Main Idea: ...
Towards the Fastest . . .
Toward Fastest Processors
How to Combine . . .
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the . . .
Proof of The . . .
Proof of The . . .
Proof of The . . .
Proofs of the Theorem
Acknowledgments

17. Acknowledgments

This work was supported in part:

- by NASA under cooperative agreement NCC5-209,
- by NSF grants EAR-0225670 and DMS-0532645,
- by Star Award from the University of Texas System, and
- by Texas Department of Transportation grant No. 0-5453.

Detailed Introduction
Formulation of the ...
Our Main Idea: In ...
Towards the Fastest ...
Toward Fastest Processors
How to Combine ...
Formal Definition
Results
Relation to Fuzzy Control
Conclusions
Proof of the ...
Proof of The ...
Proof of The ...
Proof of The ...
Proofs of the Theorem
Acknowledgments