

## Explains for the first time how "computing with words" can aid in making subjective judgments

Lotfi Zadeh, the father of fuzzy logic, coined the phrase "computing with words" (CWW) to describe a methodology in which the objects of computation are words and propositions drawn from a natural language. *Perceptual Computing* explains how to implement CWW to aid in the important area of making subjective judgments, using a methodology that leads to an interactive device—a "Perceptual Computer"—that propagates random and linguistic uncertainties into the subjective judgment in a way that can be modeled and observed by the judgment maker.

This book focuses on the three components of a *Perceptual Computer*—encoder, CWW engines, and decoder—and then provides detailed applications for each. It uses interval type-2 fuzzy sets (IT2 FSs) and fuzzy logic as the mathematical vehicle for perceptual computing, because such fuzzy sets can model first-order linguistic uncertainties whereas the usual kind of fuzzy sets cannot. Drawing upon the work on subjective judgments that Jerry Mendel and his students completed over the past decade, *Perceptual Computing* shows readers how to:

- Map word-data with its inherent uncertainties into an IT2 FS that captures these uncertainties
- Use uncertainty measures to quantify linguistic uncertainties
- Compare IT2 FSs by using similarity and rank
- Compute the subethood of one IT2 FS in another such set
- Aggregate disparate data, ranging from numbers to uniformly weighted intervals to nonuniformly weighted intervals to words
- Aggregate multiple-fired IF-THEN rules so that the integrity of word IT2 FS models is preserved

Free MATLAB-based software is also available online so readers can apply the methodology of perceptual computing immediately, and even try to improve upon it. *Perceptual Computing* is an important go-to for researchers and students in the fields of artificial intelligence and fuzzy logic, as well as for operations researchers, decision makers, psychologists, computer scientists, and computational intelligence experts.

**JERRY M. MENDEL** is Professor of Electrical Engineering at the University of Southern California. A Life Fellow of the IEEE and a Distinguished Member of the IEEE Control Systems Society, Mendel is also the recipient of many awards for his diverse research, including the IEEE Centennial Medal, the Fuzzy Systems Pioneer Award from the IEEE Computational Intelligence Society, and the IEEE Third Millennium Medal.

**DONGRUI WU** is a Postdoctoral Research Associate at the University of Southern California, where he recently obtained his PhD in electrical engineering.



Subscribe to our free Electrical Engineering eNewsletter at [wiley.com/enewsletters](http://wiley.com/enewsletters)

Visit [wiley.com/ieee](http://wiley.com/ieee)



Cover Illustration: John Woodcock/iStockphoto



MENDEL • WU

PERCEPTUAL COMPUTING

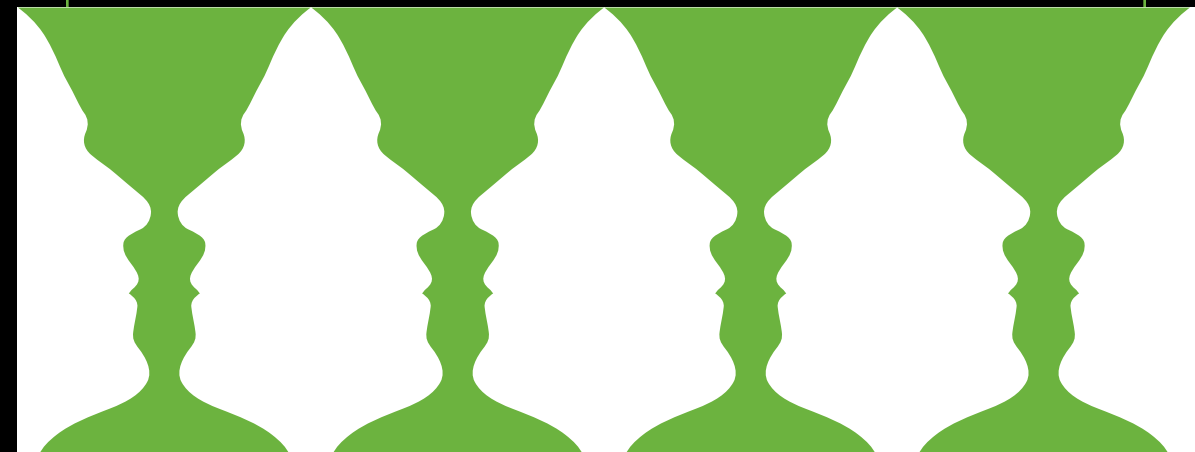
AIDING PEOPLE IN MAKING SUBJECTIVE JUDGMENTS



IEEE Press Series on Computational Intelligence  
David B. Fogel, Series Editor

# PERCEPTUAL COMPUTING

## AIDING PEOPLE IN MAKING SUBJECTIVE JUDGMENTS



JERRY M. MENDEL  
DONGRUI WU



---

# PERCEPTUAL COMPUTING

## Aiding People in Making Subjective Judgments

---

**JERRY M. MENDEL**  
**DONGRUI WU**

**IEEE Computational Intelligence Society, Sponsor**

**IEEE Press Series on Computational Intelligence**  
David B. Fogel, *Series Editor*



**IEEE PRESS**



**A JOHN WILEY & SONS, INC., PUBLICATION**

# Contents

---

<b>Preface</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Perceptual Computing	1
1.2 Examples	3
1.2.1 Investment Decision Making	3
1.2.2 Social Judgment Making	5
1.2.3 Hierarchical Decision Making	7
1.2.4 Hierarchical and Distributed Decision Making	9
1.3 Historical Origins of Perceptual Computing	11
1.4 How to Validate the Perceptual Computer	15
1.5 The Choice of Fuzzy Set Models for the Per-C	16
1.6 Keeping the Per-C as Simple as Possible	19
1.7 Coverage of the Book	20
1.8 High-Level Synopses of Technical Details	24
1.8.1 Chapter 2: Interval Type-2 Fuzzy Sets	24
1.8.2 Chapter 3: Encoding: From a Word to a Model—The Codebook	26
1.8.3 Chapter 4: Decoding: From FOU's to a Recommendation	27
1.8.4 Chapter 5: Novel Weighted Averages as a CWW Engine	29
1.8.5 Chapter 6: If-Then Rules as a CWW Engine	29
References	31
<b>2 Interval Type-2 Fuzzy Sets</b>	<b>35</b>
2.1 A Brief Review of Type-1 Fuzzy Sets	35
2.2 Introduction to Interval Type-2 Fuzzy Sets	38
2.3 Definitions	42
2.4 Wavy-Slice Representation Theorem	45
2.5 Set-Theoretic Operations	45
2.6 Centroid of an IT2 FS	46
2.6.1 General Results	46
2.6.2 Properties of the Centroid	50
2.7 KM Algorithms	52
	<b>vii</b>

**viii** CONTENTS

2.7.1	Derivation of KM Algorithms	52
2.7.2	Statements of KM Algorithms	53
2.7.3	Properties of KM Algorithms	54
2.8	Cardinality and Average Cardinality of an IT2 FS	56
2.9	Final Remark	58
	Appendix 2A. Derivation of the Union of Two IT2 FSs	58
	Appendix 2B. Enhanced KM (EKM) Algorithms	59
	References	61
<b>3</b>	<b>Encoding: From a Word to a Model—The Codebook</b>	<b>65</b>
3.1	Introduction	65
3.2	Person FOU Approach for a Group of Subjects	67
3.3	Collecting Interval End-Point Data	77
3.3.1	Methodology	77
3.3.2	Establishing End-Point Statistics For the Data	81
3.4	Interval End-Points Approach	82
3.5	Interval Approach	83
3.5.1	Data Part	84
3.5.2	Fuzzy Set Part	89
3.5.3	Observations	99
3.5.4	Codebook Example	101
3.5.5	Software	104
3.5.6	Concluding Remarks	105
3.6	Hedges	105
	Appendix 3A. Methods for Eliciting T1 MF Information From Subjects	107
3A.1	Introduction	107
3A.2	Description of the Methods	107
3A.3	Discussion	110
	Appendix 3B. Derivation of Reasonable Interval Test	111
	References	114
<b>4</b>	<b>Decoding: From FOUs to a Recommendation</b>	<b>117</b>
4.1	Introduction	117
4.2	Similarity Measure Used as a Decoder	118
4.2.1	Definitions	118
4.2.2	Desirable Properties for an IT2 FS Similarity Measure Used as a Decoder	119
4.2.3	Problems with Existing IT2 FS Similarity Measures	120
4.2.4	Jaccard Similarity Measure for IT2 FSs	121
4.2.5	Simulation Results	122
4.3	Ranking Method Used as a Decoder	123
4.3.1	Reasonable Ordering Properties for IT2 FSs	128
4.3.2	Mitchell's Method for Ranking IT2 FSs	128
4.3.3	A New Centroid-Based Ranking Method	129

4.3.4	Simulation Results	129
4.4	Classifier Used as a Decoder	130
4.4.1	Desirable Properties for Subsethood Measure as a Decoder	130
4.4.2	Problems with Four Existing IT2 FS Subsethood Measures	131
4.4.3	Vlachos and Sergiadis's IT2 FS Subsethood Measure	131
4.4.4	Simulation Results	132
Appendix 4A		135
4A.1	Compatibility Measures for T1 FSs	135
4A.2	Ranking Methods for T1 FSs	137
Appendix 4B		137
4B.1	Proof of Theorem 4.1	137
4B.2	Proof of Theorem 4.2	139
4B.3	Proof of Theorem 4.3	140
References		141
<b>5</b>	<b>Novel Weighted Averages as a CWW Engine</b>	<b>145</b>
5.1	Introduction	145
5.2	Novel Weighted Averages	146
5.3	Interval Weighted Average	147
5.4	Fuzzy Weighted Average	149
5.4.1	$\alpha$ -cuts and a Decomposition Theorem	149
5.4.2	Functions of T1 FSs	151
5.4.3	Computing the FWA	152
5.5	Linguistic Weighted Average	154
5.5.1	Introduction	154
5.5.2	Computing the LWA	157
5.5.3	Algorithms	160
5.6	A Special Case of the LWA	163
5.7	Fuzzy Extensions of Ordered Weighted Averages	165
5.7.1	Ordered Fuzzy Weighted Averages (OFWAs)	166
5.7.2	Ordered Linguistic Weighted Averages (OLWAs)	166
Appendix 5A		167
5A.1	Extension Principle	167
5A.2	Decomposition of a Function of T1 FSs Using $\alpha$ -cuts	169
5A.3	Proof of Theorem 5.2	171
References		173
<b>6</b>	<b>IF-THEN Rules as a CWW Engine—Perceptual Reasoning</b>	<b>175</b>
6.1	Introduction	175
6.2	A Brief Overview of Interval Type-2 Fuzzy Logic Systems	177
6.2.1	Firing Interval	177
6.2.2	Fired-Rule Output FOU	178
6.2.3	Aggregation of Fired-Rule Output FOUs	178
6.2.4	Type-Reduction and Defuzzification	178

## x CONTENTS

6.2.5	Observations	179
6.2.6	A Different Way to Aggregate Fired Rules by Blending Attributes	180
6.3	Perceptual Reasoning: Computations	180
6.3.1	Computing Firing Levels	181
6.3.2	Computing $\tilde{Y}_{PR}$	182
6.4	Perceptual Reasoning: Properties	184
6.4.1	General Properties About the Shape of $\tilde{Y}_{PR}$	185
6.4.2	Properties of $\tilde{Y}_{PR}$ FOU's	186
6.5	Examples	187
	Appendix 6A	191
6A.1	Proof of Theorem 6.1	191
6A.2	Proof of Theorem 6.2	191
6A.3	Proof of Theorem 6.3	191
6A.4	Proof of Theorem 6.4	192
6A.5	Proof of Theorem 6.5	192
6A.6	Proof of Theorem 6.6	192
6A.7	Proof of Theorem 6.7	193
6A.8	Proof of Theorem 6.8	194
	References	195
<b>7</b>	<b>Assisting in Making Investment Choices—Investment Judgment Advisor (IJA)</b>	<b>199</b>
7.1	Introduction	199
7.2	Encoder for the IJA	202
7.2.1	Vocabulary	202
7.2.2	Word FOU's and Codebooks	203
7.3	Reduction of the Codebooks to User-Friendly Codebooks	204
7.4	CWW Engine for the IJA	214
7.5	Decoder for the IJA	215
7.6	Examples	216
7.6.1	Example 1: Comparisons for Three Kinds of Investors	216
7.6.2	Example 2: Sensitivity of IJA to the Linguistic Ratings	221
7.7	Interactive Software for the IJA	228
7.8	Conclusions	228
	References	233
<b>8</b>	<b>Assisting in Making Social Judgments—Social Judgment Advisor (SJA)</b>	<b>235</b>
8.1	Introduction	235
8.2	Design an SJA	235
8.2.1	Methodology	236
8.2.2	Some Survey Results	238
8.2.3	Data Pre-Processing	238

8.2.4	Rulebase Generation	242
8.2.5	Computing the Output of the SJA	245
8.3	Using an SJA	246
8.3.1	Single Antecedent Rules: Touching and Flirtation	247
8.3.2	Single Antecedent Rules: Eye Contact and Flirtation	249
8.3.3	Two-Antecedent Rules: Touching/Eye Contact and Flirtation	251
8.3.4	On Multiple Indicators	253
8.3.5	On First and Succeeding Encounters	255
8.4	Discussion	255
8.5	Conclusions	256
	References	257
<b>9</b>	<b>Assisting in Hierarchical Decision Making—Procurement Judgment Advisor (PJA)</b>	<b>259</b>
9.1	Introduction	259
9.2	Missile Evaluation Problem Statement	260
9.3	Per-C for Missile Evaluation: Design	263
9.3.1	Encoder	263
9.3.2	CWW ENGINE	265
9.3.3	Decoder	266
9.4	Per-C for Missile Evaluation: Examples	266
9.5	Comparison with Previous Approaches	273
9.5.1	Comparison with Mon et al.'s Approach	273
9.5.2	Comparison with Chen's First Approach	276
9.5.3	Comparison with Chen's Second Approach	278
9.5.4	Discussion	279
9.6	Conclusions	280
	Appendix 9A: Some Hierarchical Multicriteria Decision-Making Applications	280
	References	282
<b>10</b>	<b>Assisting in Hierarchical and Distributed Decision Making—Journal Publication Judgment Advisor (JPJA)</b>	<b>283</b>
10.1	Introduction	283
10.2	The Journal Publication Judgment Advisor (JPJA)	284
10.3	Per-C for the JPJA	285
10.3.1	Modified Paper Review Form	285
10.3.2	Encoder	286
10.3.3	CWW ENGINE	288
10.3.4	Decoder	290
10.4	Examples	291
10.4.1	Aggregation of Technical Merit Subcriteria	291
10.4.2	Aggregation of Presentation Subcriteria	294

**xii** CONTENTS

10.4.3	Aggregation at the Reviewer Level	299
10.4.4	Aggregation at the AE Level	300
10.4.5	Complete Reviews	304
10.5	Conclusions	309
	Reference	310
<b>11</b>	<b>Conclusions</b>	<b>311</b>
11.1	Perceptual Computing Methodology	311
11.2	Proposed Guidelines for Calling Something <i>CWW</i>	312
	<b>Index</b>	<b>315</b>



## **Preface**

---

Life is full of subjective judgments: those we make that affect others and those that others make that affect us. Such judgments are personal opinions that have been influenced by one's personal views, experience, or background, and can also be interpreted as personal assessments of the levels of variables of interest. They are made using a mixture of qualitative and quantitative information. Emotions, feelings, perceptions, and words are examples of qualitative information that share a common attribute: they cannot be directly measured; for example, eye contact, touching, fear, beauty, cloudiness, technical content, importance, aggressiveness, and wisdom. Data (one- or multidimensional) and possibly numerical summarizations of them (e.g., statistics) are examples of quantitative information that share a common attribute: they can be directly measured or computed from direct measurements; for example, daily temperature and its mean value and standard deviation over a fixed number of days; volume of water in a lake estimated on a weekly basis, as well as the mean value and standard deviation of the estimates over a window of years; stock price or stock-index value on a minute-to-minute basis; and medical data, such as blood pressure, electrocardiograms, electroencephalograms, X-rays, and MRIs.

Regardless of the kind of information—qualitative or quantitative—there is uncertainty about it, and more often than not the amount of uncertainty can range from small to large. Qualitative uncertainty is different from quantitative uncertainty; for example, words mean different things to different people and, therefore, there are linguistic uncertainties associated with them. On the other hand, measurements may be unpredictable—random—because either the quantity being measured is random or it is corrupted by unpredictable measurement uncertainties such as noise (measuring devices are not perfect), or it is simultaneously random and corrupted by measurement noise.

Yet, in the face of uncertain qualitative and quantitative information one is able to make subjective judgments. Unfortunately, the uncertainties about the information propagate so that the subjective judgments are uncertain, and many times this happens in ways that cannot be fathomed, because these judgments are a result of things going on in our brains that are not quantifiable.

**xiv** PREFACE

It would be wonderful to have an interactive device that could aid people in making subjective judgments, a device that would propagate random and linguistic uncertainties into the subjective judgment, but in a way that could be modeled and observed by the judgment maker. This book is about a methodology, perceptual computing, that leads to such a device: a perceptual computer (Per-C, for short). The Per-C is not a single device for all problems, but is instead a device that must be designed for each specific problem by using the methodology of perceptual computing.

In 1996, Lotfi Zadeh, the father of fuzzy logic, published a paper with the very provocative title “Fuzzy Logic = Computing With Words.” Recalling the song, “Is That All There Is?,” his article’s title might lead one to incorrectly believe that, since fuzzy logic is a very well-developed body of mathematics (with lots of real-world application), it is straightforward to implement his paradigm of computing with words. The senior author and his students have been working on one class of applications for computing with words for more than 10 years, namely, subjective judgments. The result is the perceptual computer, which, as just mentioned, is not a single device for all subjective judgment applications, but is instead very much application dependent. This book explains how to design such a device within the framework of perceptual computing.

We agree with Zadeh, so fuzzy logic is used in this book as the mathematical vehicle for perceptual computing, but not the ordinary fuzzy logic. Instead, interval type-2 fuzzy sets (IT2 FSs) and fuzzy logic are used because such fuzzy sets can model first-order linguistic uncertainties (remember, words mean different things to different people), whereas the usual kind of fuzzy sets (called type-1 fuzzy sets) cannot.

Type-1 fuzzy sets and fuzzy logic have been around now for more than 40 years. Interestingly enough, type-2 fuzzy sets first appeared in 1975 in a paper by Zadeh; however, they have only been actively studied and applied for about the last 10 years. The most widely studied kind of a type-2 fuzzy set is an IT2 FS. Both type-1 and IT2 FSs have found great applicability in function approximation kinds of problems in which the output of a fuzzy system is a number, for example, time-series forecasting, control, and so on. Because the outputs of a perceptual computer are words and possibly numbers, it was not possible for us to just use what had already been developed for IT2 FSs and systems for its designs. Many gaps had to be filled in, and it has taken 10 years to do this. This does not mean that the penultimate perceptual computer has been achieved. It does mean that enough gaps have been filled in so that it is now possible to implement one kind of computing with words class of applications.

Some of the gaps that have been filled in are:

- A method was needed to map word data with its inherent uncertainties into an IT2 FS that captures these uncertainties. The *interval approach* that is described in Chapter 3 is such a method.
- Uncertainty measures were needed to quantify linguistic uncertainties. Some uncertainty measures are described in Chapter 2.

- How to compare IT2 FSs by using similarity was needed. This is described in Chapter 4.
- How to rank IT2 FSs had to be solved. A simple ranking method is also described in Chapter 4.
- How to compute the subsethood of one IT2 FS in another such set had to be determined. This is described in Chapter 4.
- How to aggregate disparate data, ranging from numbers to uniformly weighted intervals to nonuniformly weighted intervals to words, had to be determined. Novel weighted averages are a method for doing this. They include the interval weighted average, fuzzy weighted average and the linguistic weighted average, and are described in Chapter 5.
- How to aggregate multiple-fired if-then rules so that the integrity of word IT2 FS models is preserved had to be determined. Perceptual reasoning, which is described in Chapter 6, does this.

We hope that this book will inspire its readers to not only try its methodology, but to improve upon it.

So that people will start using perceptual computing as soon as possible, we have made free software available online for implementing everything that is in this book. It is MATLAB-based (MATLAB<sup>®</sup> is a registered trademark of The Mathworks, Inc.) and was developed by the second author, Feilong Liu, and Jhiin Joo, and can be obtained at <http://sipi.usc.edu/~mendel/software> in folders called “Perceptual Computing Programs (PCP)” and “IJA Demo.” In the PCP folder, the reader will find separate folders for Chapters 2–10. Each of these folders is self-contained, so if a program is used in more than one chapter it is included in the folder for each chapter. The IJA Demo is an interactive demonstration for Chapter 7.

We want to take this opportunity to thank the following individuals who either directly contributed to the perceptual computer or indirectly influenced its development: Lotfi A. Zadeh for type-1 and type-2 fuzzy sets and logic and for the inspiration that “fuzzy logic = computing with words,” the importance of whose contributions to our work is so large that we have dedicated the book to him; Feilong Liu for codeveloping the interval approach (Chapter 3); Nilesh Karnik for codeveloping the KM algorithms; Bob John for codeveloping the wavy slice representation theorem; Jhiin Joo for developing the interactive software for the investment judgment advisor (Chapter 7); Terry Rickard for getting us interested in subsethood; and Nikhil R. Pal for interacting with us on the journal publication judgment advisor.

The authors gratefully acknowledge material quoted from books or journals published by Elsevier, IEEE, Prentice-Hall, and Springer-Verlag. For a complete listing of quoted books or articles, please see the References. The authors also gratefully acknowledge Lotfi Zadeh and David Tuk for permission to publish some quotes from private e-mail correspondences.

The first author wants to thank his wife Letty, to whom this book is also dedicated, for providing him, for more than 50 years, with a wonderful and supportive



**xvi** PREFACE

environment that has made the writing of this book possible. The second author wants to thank his parents, Shunyou Wu and Shenglian Luo, and his wife, Ying Li, to whom this book is also dedicated, for their continuous encouragement and support.

JERRY M. MENDEL  
DONGRUI WU

*Los Angeles, California*  
*September 2009*

