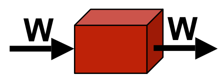
Perceptual Computing: One Implementation of Zadeh's Computing With Words Paradigm

Jerry M. Mendel
Ming Hsieh Department of Electrical Engineering
University of Southern California
mendel@sipi.usc.edu

Computing With Words (CWW)



- Zadeh (1996): "Fuzzy Logic = Computing With Words"
- Our viewpoint: CWW is a *paradigm*
- CWW can be approached from different points of view and levels of abstraction
- **Our Focus: How do you actually do it?**

2

• Lotfi Zadeh, the father of fuzzy logic (FL), is also the father of "Computing With Words (CWW)." His 1996 article (and his later 1999 article) that equated FL with CWW may have made it seem like all of the work was done, since by 1996 FL had been around more than 30 years; however, this was not the case.

• By CWW, Zadeh did not mean that computers would actually compute using words. What he meant is that one should use FL to design a device that accepts words at its inputs and provides words at its output.

• This is not as easy as it sounds.

• Our viewpoint about CWW is that it is a **high-level paradigm**, one that can be approached from different points of view, from abstract to very practical.

• Our focus in this talk is on **how do you actually implement CWW**?

Our Focus

- CWW for aiding people in making *subjective judgments*
  - **Subjective Judgment**: Personal opinions influenced by personal views, experience or background—personal assessments of levels of variables of interest, made using mixtures of qualitative and quantitative information

3

• To answer this question we have focused on how to implement CWW for a broad class of applications that we call **CWW for aiding people in making subjective judgments**.

• What is meant by a *subjective judgment* is explained on the slide.

• Some examples of subjective judgments are given on the next slide.

Our Focus

- CWW for assisting people in making subjective judgments
  - **Subjective Judgment**: Personal opinions influenced by personal views, experience or background—personal assessments of levels of variables of interest, made using mixtures of qualitative and quantitative information
  - **Examples**
    - Investment alternative judgments
    - Publication judgments
    - Competition evaluation judgments
    - Social judgments (flirtation, honesty, reliability)

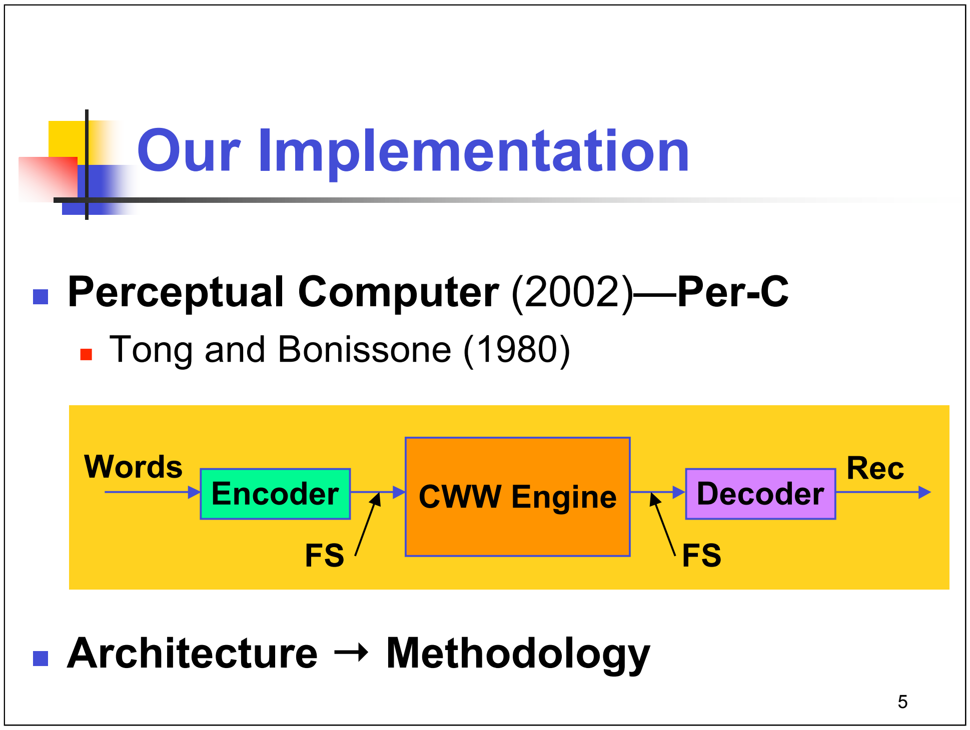
4

• 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.

• 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.

• This talk is about a device that can propagate random and linguistic uncertainties into the subjective judgment, but in a way that can be modeled and observed by the judgment maker.



• We call this device a *Perceptual Computer*, *Per-C* for short.

• A search of the literature has revealed that in 1980 Tong and Bonissone were the first to propose the essence of the Per-C, although they did not use the same terminology that we will use.

• The Per-C has three components:

1. **Encoder**: It maps words in a pre-specified application-dependent vocabulary into a fuzzy set model; the result is the **codebook** for the application.

2. **CWW Engine**: It maps fuzzy sets into fuzzy sets, and can have different forms, again depending upon the application.

3. **Decoder**: It maps fuzzy sets into a recommendation, e.g. word, rank, or class, where the kind of recommendation again depends upon the specific application.

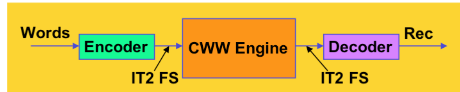
• In addition to the **architecture** of the Per-C, a **methodology** has evolved for how to design its three components. It will be explained later in this lecture.

This Talk

Obstacles (**Challenges**) to implementing the Per-C and how they have been overcome

6

- The main purpose of this talk is to focus on the obstacles (some might call them *challenges*) to implementing the Per-C and how they have been overcome.
- It has taken us more than 10 years to do this.
- Instead of listing all of the obstacles, I will develop them one at a time, after which I will summarize all of them in a list.



Which FSs to Use?: 1

- Interval T2 FSs—what else? ☺

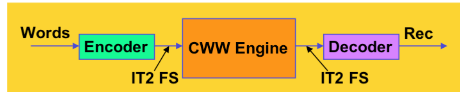
7

ISSUE #1: Which Fuzzy Sets (FSs) should be used to model a word?

• Although Zadeh equated CWW with fuzzy logic (FL), he did not tell us what kind of fuzzy sets we should use.

• For those of you who are familiar with my past works on type-2 fuzzy sets, it will not come as a big surprise when I tell you that we are going to use interval type-2 fuzzy sets (IT2 FSs).

• But why?



Which FSs to Use?: 2

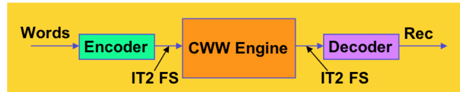
- Interval T2 FSs—what else? ☺
- Karl Popper's ***Falsificationism*** used to demonstrate that to model a word by a T1 FS is scientifically incorrect! (2003)

8

• In 2003, at FUZZ-IEEE in St. Louis, I presented a paper (Mendel, 2003) in which I demonstrated that it is scientifically incorrect to model a word using a type-1 FS.

• My arguments were based on Sir Karl Popper's **Falsificationism**.

• I have published this in a number of articles since then, e.g. Mendel (2007a, c).



Which FSs to Use?: 3

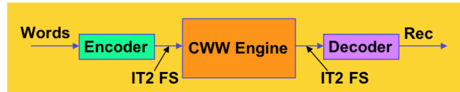
- T2 FSs—what else? ☺
- Karl Popper's ***Falsificationism*** used to demonstrate that to model a word by a T1 FS is scientifically incorrect! (2003)
  - *Words mean different things to different people*
  - *Once parameters of a T1 FS are fixed there is nothing uncertain about the T1 FS*
  - *It's a contradiction to model something that is uncertain by something that is certain*

9

• Whereas a T1 FS may be a scientific model for a word, because one can collect data about the word from subjects and then map that data into a T1 FS, it is a scientifically incorrect model for a word.

• My three-line proof of this is on the slide.

• You may raise objections to this proof. Please see my article in the CIS Magazine (Mendel, 2007c) where such objections are presented and refuted.



Which FSs to Use?: 3

- T2 FSs—what else? ☺
- Karl Popper's ***Falsificationism*** used to demonstrate that to model a word by a T1 FS is scientifically incorrect! (2003)
  - *Words mean different things to different people*
  - *Once parameters of a T1 FS are fixed there is nothing uncertain about the T1 FS*
  - *It's a contradiction to model something that is uncertain by something that is certain*
- **IT2FSs**—first-order uncertainty model of a word

10

• Our alternative to a T1 FS is to use an **interval type-2 FS** (IT2 FS).

• This is a special, but very important kind of T2 FS, one in which all of the secondary membership grades equal one.

• Hence, it is called a *first-order uncertainty model* of a word.

• A *second-order uncertainty model* is one in which all of the secondary membership grades are not the same.

• The reason this is called a *second-order uncertainty model* is because there is uncertainty about how to choose such secondary membership grades.

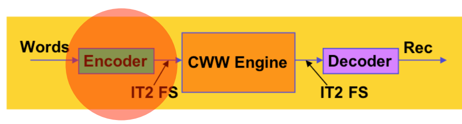
• In an IT2 FS the uncertainty is spread uniformly across its footprint of uncertainty (FOU).

• In the future, it may be possible to obtain second-order (and even higher-order) uncertainty models for a word, but as of this date, this is not possible.

• For readers who are unfamiliar with an IT2 FS, see e.g., Mendel (2007b).

How to Map Data About Words into a FS Model?

- Collect data about words from a group of subjects
  - *Words mean different things to different people*
    - Intra- and inter- uncertainties
  - To capture such uncertainties we collect data about the words across a group of subjects
    - *"On a scale of 0–10, where would you locate the end-points of an interval (range) that you associate with the word?"*
  - Data are application dependent



ISSUE #2: How do we map data collected about a word into an IT2 FS, I.e., how do we design the Encoder?

• This is an inverse problem.

• We must collect data about a word from a group of subjects and then map that data into its FOU.

• Because "Words mean different things to different people," we must collect data from people so that each person's uncertainty about the word— the **intra-uncertainty**— as well as the uncertainty across the group of subjects—the **inter-uncertainty**— is captured.

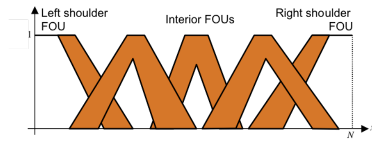
• This data should be collected in such a manner that methodological uncertainties are not introduced. Such uncertainties cannot be unraveled from the word uncertainties.

• It is our conviction that to ask people to provide MF values for a word introduces methodological uncertainties, because most people have no idea what a MF is (you are the exception).

• So, we ask them a very simple question that is stated on the slide.

• Our experience with carrying out surveys using this kind of question is that anyone an answer it.

• Although we have published articles showing data collected about words in a context-free environment, this was for illustration purposes only. Data needs to be collected in an application-dependent environment—**context matters**!



Encoder: 1

- **Interval Approach** (**IA**) (Liu and Mendel, 2008)
  - After some pre-processing of data intervals, use a derived *classification diagram* to establish the kind of FOU for the word—LS, INT or RS FOU
  - Assume each interval is uniformly distributed
  - Map mean and variance of each interval into a respective T1 FS (LS, INT or RS)
  - Aggregate T1 FSs into FOU of IT2 FS—Uses T2 FS Representation Theorem in reverse!
  - Bound FOU using piecewise-linear functions—LMF and UMF
  - Only three kinds of FOU shapes occur: LS, INT and RS

• In 2008, Feilong Liu and I published an article (Liu and Mendel, 2008) in which we presented a method called the **Interval Approach** (IA), that provides a complete method for going from data intervals that are collected from a group of subjects to the FOU of the word.

• It uses probability, statistics, classification and the Representation Theorem (Mendel, 2007b) for an IT2 FS.

• First, data pre-processing is done to remove “bad intervals.”

• The remaining intervals are collectively classified into either a left-shoulder, interior or right-shoulder FOU (see figure at top of this slide)—only these three shapes are possible (this will be described on the next slide).

• Each remaining data interval is assumed to be uniformly distributed, so that we can compute its mean and standard deviation (SD).

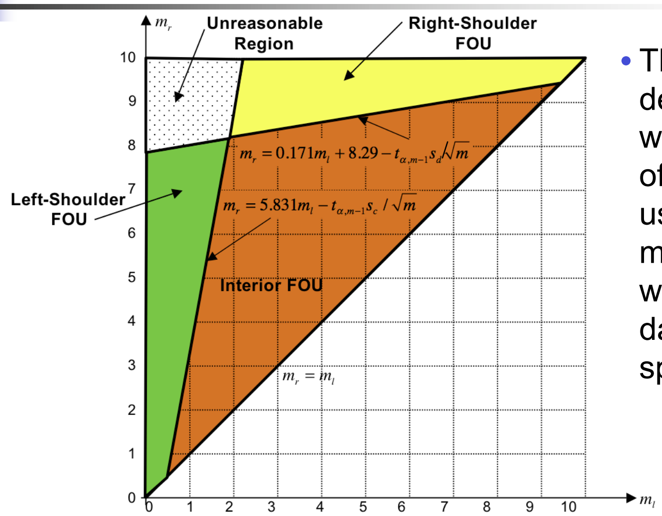
• By equating the mean and SD of each interval to the mean and SD of the just established T1 FS, we obtain the parameters of the T1 FS in terms of the interval end-points.

• In this way, uncertainty about each person’s interval is transferred into a T1 FS.

• After doing this for all of the data intervals, the collection of T1 FSs are upper and lower bounded.

• The result is the upper MF (UMF) and lower MF (LMF) for the FOU of the word.

Encoder: 2—Classification Diagram



- The data determine which kind of FOU is used to model the word—the data speaks!

13

• This slide shows the classification diagram.

• It has four regions, but only three are important.

• Average values and SD's for the left and right end-points of a group's data intervals locate a point on this diagram, from which one classifies the word's FOU as a left shoulder, interior or right shoulder.

• The diagram's decision lines are very easy to obtain and are for three "obvious" data requirements: (1) the left-end of the data interval must be greater than zero; (2) the right-end of the data interval must be less than ten; and, (3) the left-end of the data interval must be to the left of the right-end of the data interval.

• Small-sample statistics have been used, so the results are valid for small and large samples.

• What is really important is that in the IA the kind of FOU is not chosen ahead of time.

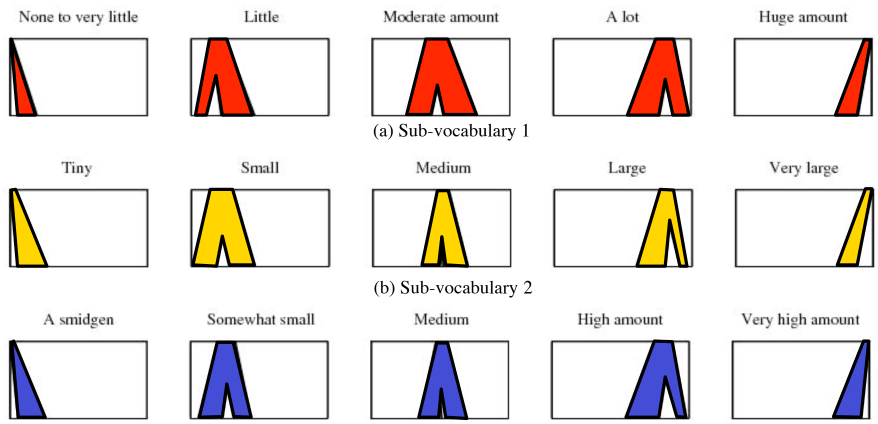
• Instead, *the data speaks*, and it establishes the kind of FOU for a word.

• By using the group's data statistics in the classification diagram, we are making use of the **inter-uncertainties** about the word.

• By then using each interval's statistics to obtain a T1 FS, we are making use of the **intra-uncertainties** about the word.

• By aggregating and bounding all of the T1 FSs, we are again making use of the **inter-uncertainties** about the word.

Encoder: 3—Examples



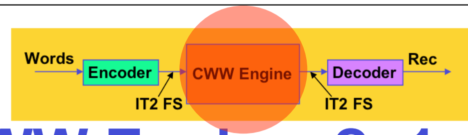
(a) Sub-vocabulary 1

(b) Sub-vocabulary 2

(c) Sub-vocabulary 3

14

- Here are some examples of FOUs obtained for three sub-vocabularies, each having five words.
- These five FOUs cover the entire domain 0-10, and they overlap.
- The five words were chosen from results that were obtained for a 32-word vocabulary [see the Lu and Mendel (2008) paper].
- Observe that in general these FOUs do not have parallel lines and usually one side has more area than the other side.
- We call the collection of words and IT2 FS Models a **Codebook**, because each FOU is coded by the parameters of its LMF and UMF.
- In summary, **the IA is our Encoder.**



What are CWW Engines?: 1

- **CWW Engines** are constrained by a new requirement that their outputs must resemble the IT2 FS models in the codebook

15

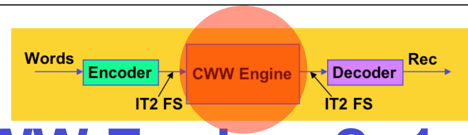
ISSUE #3: What are CWW Engines and How are they Designed?

• In addition to *words meaning different things to different people*, they must also *mean similar things to different people*, or else communication is not possible.

• Consequently, we have interpreted this to mean that the outputs from a CWW Engine must be an FOU that resembles the FOUs in the Codebook.

• This is a constraint, i.e. a new requirement on the output of a CWW engine, one that is not needed when FSs are used in other engineering applications.

• This makes the design of a CWW Engine challenging and interesting.



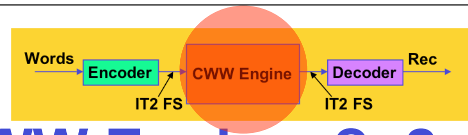
What are CWW Engines?: 1

- **CWW Engines** are constrained by a new requirement that their outputs must resemble the IT2 FS models in the codebook
  - CWW mappings of codebook IT2 FSs—non-linear mappings—must resemble the judgment words into which they will be decoded
    - We do not (yet) impose this constraint on the design of the CWW Engine
    - Instead we demonstrate by analysis that the output from each CWW Engine satisfies this constraint

16

• We do not yet know how to design *constrained* CWW Engines.

• So our approach has been to design what we believe is a useful CWW Engine and to then analyze its possible outputs to see if they satisfy this new requirement.



What are CWW Engines?: 2

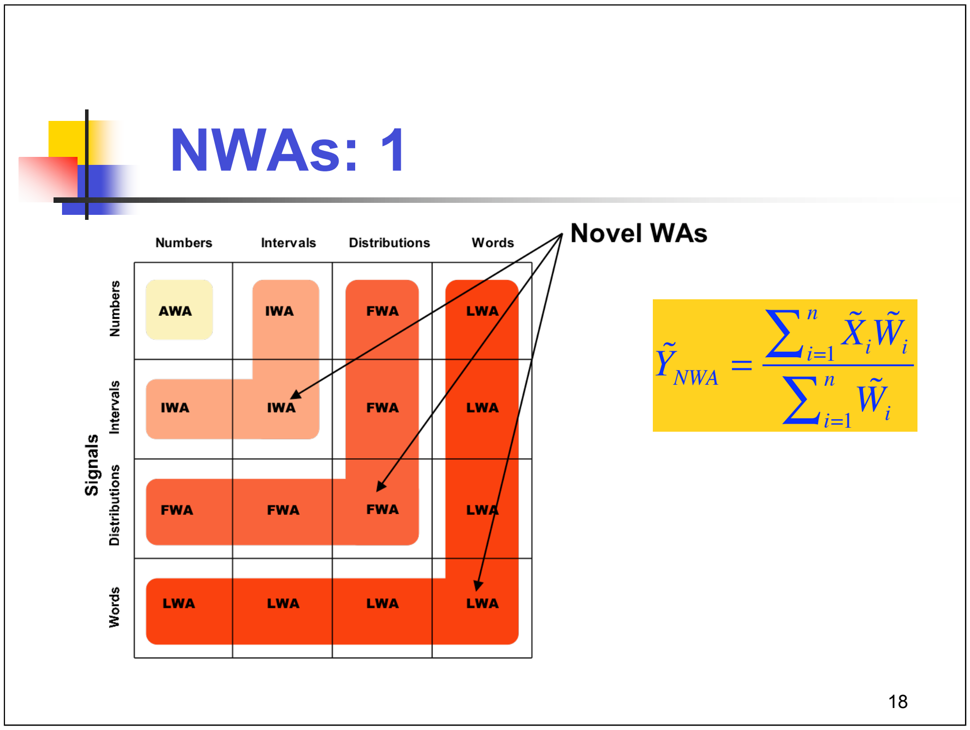
- Two **CWW Engines** (to-date)
  - Novel weighted averages (NWAs)
    - IWA
    - FWA
    - LWA
    - O-NWAs
  - IF-THEN rules

17

• To-date, we have only designed two CWW Engines, Novel Weighted Averages (NWAs) and IF-THEN rules,

• We will explain what NWAs are and what IF-THEN rules are.

• Designing other CWW Engines is a very fruitful research area.



Novel Weighted Averages

• Everyone is familiar with the arithmetic weighted average—it is not novel.

• The equation that is on the slide, for a NWA, should be interpreted as an *expressive equation*, i.e. the sub-criteria, *X-tilde*$_i$, and the weights, *W-tilde*$_i$, do not get multiplied, summed and divided.

• Computation of a NWA is beyond the scope of this talk, but it is done using alpha-cut techniques and KM Algorithms (Mendel, 2007b).

• The sub-criteria and weights can range from uniformly-weighted intervals, to non-uniformly-weighted intervals (i.e., T1 FSs) to words (i.e., IT2 FSs).

NWAs: 2

$$\tilde{Y}_{NWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i}$$

- Can aggregate everything from numbers to uniformly-weighted intervals to non-uniformly-weighted interval to words
- Weights can also be any of the above
- Linguistic Weighted Average (LWA) is most general NWA—IWA and FWA are special but important cases of LWA
- Alpha-cuts and KM Algorithms used to compute NWAs
- *FOU*(*LWA*) resembles word FOUs in a codebook
  - Constraint is satisfied by the LWA

19

• When at most uniformly-weighted intervals are used, the resulting NWA is called an **Interval Weighted Average** (IWA).

• When at most non-uniformly-weighted intervals are used, the resulting NWA is called a **Fuzzy Weighted Average** (FWA).

• When words are used, the resulting NWA is called a **Linguistic Weighted Average** (LWA).

• KM algorithms directly compute an IWA.

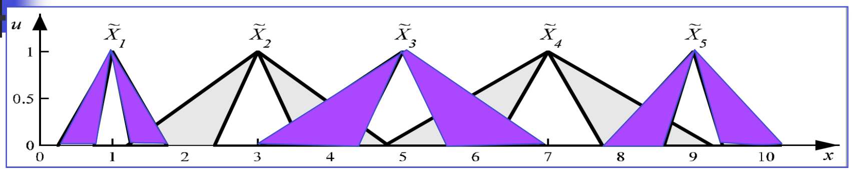
• A FWA is computed as a collection of IWAs.

• A LWA is computed as two FWAs, one for its LMF and one for its UMF.

• We have proven that FOU(LWA) resembles the word FOUs in a codebook, i.e. they can only look like left-shoulder, interior or right-shoulder FOU.

$$\tilde{Y}_{NWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i}$$

NWAs: 3—Example

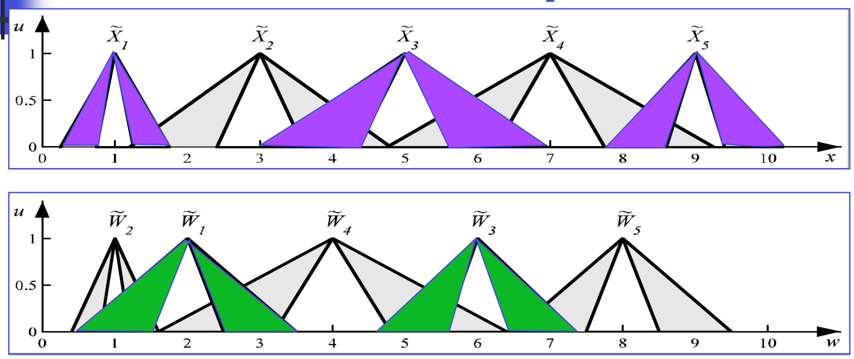


20

- In this example, both the sub-criteria and the weights are IT2 FSs and we will compute the LWA.
- The FOUs are not from an actual codebook. They are all triangular.
- The two colors are used just for “special” effects.
- In this LWA there are five sub-criteria, *X-tilde*$_1$, *X-tilde*$_2$, *X-tilde*$_3$, *X-tilde*$_4$, and *X-tilde*$_5$.

$$\tilde{Y}_{NWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i}$$

NWAs: 3—Example



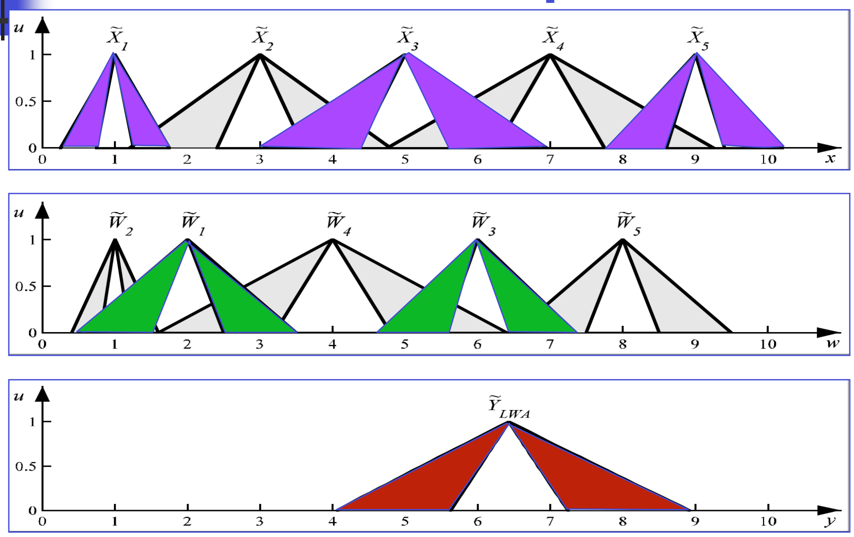
21

• Observe that the weight FOUs are not in chronological order.

• This means that the five sub-criteria are weighted in the following rank-ordering: *X-tilde*$_5$, *X-tilde*$_3$, *X-tilde*$_4$ *X-tilde*$_1$, and *X-tilde*$_2$.

NWAs: 3—Example

$$\tilde{Y}_{NWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i}$$



• The most important thing to observe from the LWA FOU is that it resembles the FOUs of the sub-criteria and the weights, supporting our earlier statement that FOU(LWA) resembles the word FOUs in a codebook.

• The uncertainties contained in the sub-criteria and weight FOUs have been mapped into the uncertainty of the LWA FOU.

• Even though the LWA FOU looks symmetrical, it is not.

• The centroid of the LWA FOU can be computed using KM algorithms.

• It will be an interval whose length provides a measure of the uncertainty for the LWA FOU.

IF-THEN Rules:

- Rules: IF $\tilde{X}$ is $\tilde{F}$, THEN y is $\tilde{G}$

23

IF-THEN Rules

• Our second CWW Engine is IF-THEN rules, an example of which is given on the slide.

• To keep the discussion simple, I am only showing a single antecedent rule.

• All of you should be familiar with such rules from your knowledge about T1 FL and T1 fuzzy logic systems.

• Recall, that when an input activates the rule, first a **firing activation** must be computed and then the **fired-rule output set** must be computed.

• And, when there is more than one rule—the usual case—and more than one rule is fired by the input(s) then fired-rule output sets have to be combined.

IF-THEN Rules:

- Rules: IF $\tilde{X}$ is $\tilde{F}$, THEN y is $\tilde{G}$
- Issues
  - How to compute firing activation?
    - Words (IT2 FSs) activate each antecedent
    - Non-singleton fuzzification—too complicated
    - Mamdani activation—leads to firing intervals

24

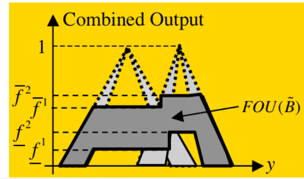
• There are two major issues concerning rules, the first of which is on this slide, namely: **How to compute the firing activation?**

• It is very tempting to use Mamdani (inference) activation, because it is so widely used in non-computing with word IF-THEN rule applications, e.g., fuzzy logic control, signal processing, etc..

• When a word activates a rule, as in the Per-C, this is equivalent to what is called **non-singleton IT2 fuzzification**, and calculating the sup-star composition (as is required by Mamdani inference), which is needed in order to compute the firing activation, is the most complicated calculation for all of the kinds of fuzzification possibilities.

• Regardless, in the case of IT2 FSs, this calculation leads to a **firing interval** for the firing activation.

• However, even if only one rule is fired (e.g., when minimum t-norm is used) the fired-rule output set is a clipped version of the FOU of the rule's consequent, and this FOU does not resemble the three kinds of FOUs in the Per-C's Codebook.



IF-THEN Rules:

- Rules: IF $\tilde{X}$ is $\tilde{F}$, THEN y is $\tilde{G}$
- Issues
  - How to compute firing activation?
    - Words (IT2 FSs) activate each antecedent
    - Non-singleton fuzzification—too complicated
    - Mamdani activation—leads to firing intervals
  - How to combine multiple fired rules?
    - Mamdani combining leads to FOUs that don't resemble FOUs in codebook (e.g., they are not normal and look clipped—see above figure)

25

• Things are even worse when more than one rule is fired.

• If, e.g., the fired-rule output sets are combined by taking their union, then the resulting IT2 FS bares no resemblance whatsoever to the three kinds of FOUs in the Codebook.

• This is illustrated in the figure that is at the top of this slide, which is for two fired rules whose fired-rule output sets have been combined using union.

• The result is FOU(*B-tilde*).

• So, when IF-THEN rules are used as a CWW Engine, it is no longer business as usual.

• New approaches are needed, and they are described next.

IF-THEN Rules: Perceptual Reasoning

- **Similarity** used to compute a firing level (rather than an interval)
  - Better than computing a firing interval
    - Aggregated multiple fired rules more closely resemble codebook FOUs
  - Jaccard similarity measure is used for IT2 FSs

$$\text{IF } \tilde{X} \text{ is } \tilde{F}, \text{THEN } y \text{ is } \tilde{G}$$

$$s_J(\tilde{X},\tilde{F}) = \frac{\sum_{i=1}^{N} \min(\bar{\mu}_{\tilde{X}}(x_i), \bar{\mu}_{\tilde{F}}(x_i)) + \sum_{i=1}^{N} \min(\underline{\mu}_{\tilde{X}}(x_i), \underline{\mu}_{\tilde{F}}(x_i))}{\sum_{i=1}^{N} \max(\bar{\mu}_{\tilde{X}}(x_i), \bar{\mu}_{\tilde{F}}(x_i)) + \sum_{i=1}^{N} \max(\underline{\mu}_{\tilde{X}}(x_i), \underline{\mu}_{\tilde{F}}(x_i))}$$

26

• We have called the new approach for reasoning with IF-THEN rules in a CWW Engine **Perceptual Reasoning** (Mendel and Wu, 2008).

• This slide focuses on how the firing activation is computed in Perceptual Reasoning.

• Instead of using a sup-star way to compute a firing interval, we prefer to compute a firing level—a number.

• This is done using the **Jaccard similarity measure** for IT2 FSs.

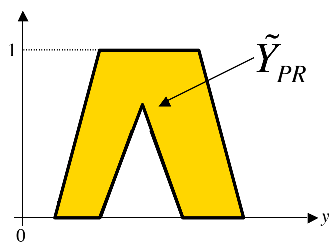
• The formula for the Jaccard similarity measure between input word *X-tilde* and its associated rule antecedent *F-tilde* is shown on this slide.

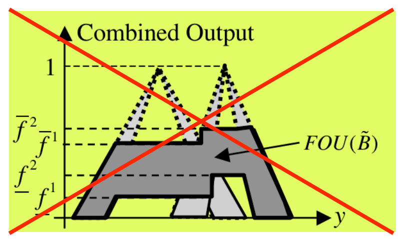
• All of these similarities can be **pre-computed** and stored as a table, because input words and antecedent words can only come from the established Codebook.

• This again is another major difference between CWW calculations for IF-THEN rules and non-CWW calculations (for which the inputs to the rules are not restricted to a codebook, and are not usually known ahead of time).

IF-THEN Rules: Perceptual Reasoning

- Multiple fired rules combined using special **NWA** of IT2 FSs of consequents of fired rules
- *FOU*(*Combined fired rules*) resembles word FOUs in codebook





27

• This slide focuses on how multiple-fired rules are combined in Perceptual Reasoning (PR).

• Union is NOT used, nor is type-reduction (Mendel, 2007b) performed.

• Fired rules are combined—aggregated—by using a special case of our NWA!

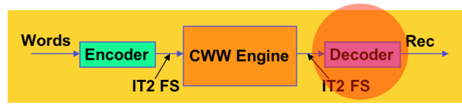
• In this NWA, we use a mixture of consequent FOUs as the “sub-criteria” and firing levels (numbers) as the “weights.”

• One could say that **this NWA is a weighted arithmetic average of the consequents of the fired rules**.

• We have proved that the FOU from this NWA resembles the three kinds of FOUs in the Codebook.

• This is exemplified by the two figures at the bottom of this slide.

• The one on the left is for PR.



What are the Decoders?

Three kinds of Recommendations—Decoders

- **1. Words**
  - Jaccard similarity measure developed for IT2 FSs
    - Captures proximity and shapes of FOUs

28

ISSUE #4: How do we design the Decoder?

• As was mentioned earlier, the output of the Decoder is a **Recommendation**, and to-date there can be three kinds of recommendations: word, rank and class.

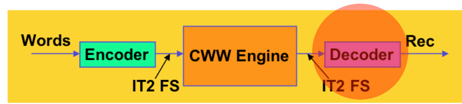
Words

• When the recommendation is a **word**, we compute the similarity between the FOU at the output of the CWW Engine with all of the words that are in the Codebook, and choose the winning word as the one with the largest similarity.

• Note that, for CWW, an exact shape of two FOUs is not sufficient for a word to be similar to another word, because these same shapes may be located far from each other on our scale of 0–10.

• We use the Jaccard similarity measure for IT2 FSs, because it simultaneously captures similarity of shape and proximity between two IT2 FSs.

• You might say that it “kills two birds with one stone.”



What are the Decoders?

Three kinds of Recommendations—Decoders

- 1. Words
  - Jaccard similarity measure developed for IT2 FSs
    - Captures proximity and shapes of FOUs
- **2. Rankings**
  - Average of centroid of an IT2 FS used
  - Seems to rank very well
  - No optimal ranking method exists

29

Rankings

• We are unaware of any *optimal* way to rank fuzzy sets.

• Ranking FOUs can be thought of as ranking a group of pictures, that lie on a scale (0–10) because each FOU is a picture.

• Our **procedure for ranking a group of IT2 FSs** is:

1. Compute the centroid of each IT2 FS, by using the KM algorithms. This will be an interval, one with smallest and largest values.

2. Compute the average value of the centroids, i.e. the average value of the smallest and largest values of the centroid.

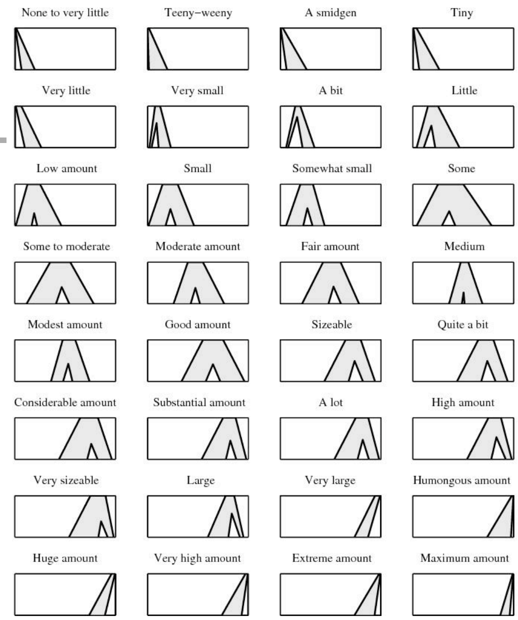
3. Rank the IT2 FSs by ranking the average value of the centroids.

• This ad hoc procedure seems to rank word FOUs very well.

• If you don't like the rankings of some of the FOUs, then just change them—remember, there is nothing sacrosanct about this ranking procedure.

Decoders

- **Example—**
  Ranking of FOUs for words



30

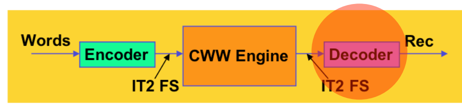
• The 32 FOUs shown on this slide are from the Liu & Mendel IA paper (2008).

• The figure should be read (lexicographically) by starting at its top row and scanning from left to right and then going down to the next row and repeating this process.

• Observe that our ranking of the 32 FOUs seems to give “correct” results, in that the word FOUs flow from left to right in a sensible linguistic ordering.

• Observe, also, that all of the left-shoulder FOUs appear before all of the interior FOUs, which appear before all of the right-shoulder FOUs.

• If this does not always occur, then just move a left shoulder FOU in with the other left-shoulder FOUs, etc.



Decoders

- **3. Class**
  - Subsethood of IT2 FSs developed by Terry Rickard (uses Kosko's subsethood for T1 FSs and the RT)

$$SS(\tilde{A},\tilde{B}) = \bigcup_{\forall A_e, B_e} ss(A_e, B_e)$$

$$= \bigcup_{\forall A_e, B_e} \frac{\sum_{i=1}^{N} \min(\mu_A(x_i), \mu_A(x_i))}{\sum_{i=1}^{N} \mu_A(x_i)}$$

$$= [ss_l(\tilde{A},\tilde{B}), ss_r(\tilde{A},\tilde{B})]$$

31

Class

• Classification of the **output FOU** from the CWW Engine is used when that FOU is not actually a word in the codebook for the recommendation.

• As an **example**: In our **Journal Publication Judgment Advisor** (Mendel and Wu, 2010), the output of the CWW Engine is an FOU that is obtained by aggregating FOUs from a group of reviewers. This aggregated FOU has to then be classified as either *Accept*, *Revise* or *Reject*.

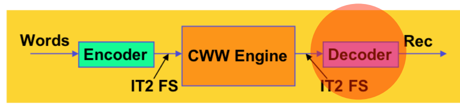
• **Subsethood** (actually average subsethood) is used by us to perform this classification.

• We compute the subsethood of the output FOU from the CWW Engine with all of the words in the Recommendation Codebook, and choose the winning recommendation as the class-word with the largest average subsethood.

• Recently, Terry Rickard et al. (2009) extended Kosko's definition of subsethood from T1 FSs to IT2 FSs.

• We use this extension to first compute the subset for an IT2 FS.

• It is an interval of numbers.



Decoders

- **Class** (Continued)
  - Average subsethood used

$$ss(\tilde{A},\tilde{B}) = \frac{ss_l(\tilde{A},\tilde{B}) + ss_r(\tilde{A},\tilde{B})}{2}$$

  - Axiomatic properties proved for average subsethood
  - More work is underway about subsethood

32

• Because the subsethood of an IT2 FS is an interval, we then compute the average value of this subsethood, and then use it to make our final classification decision.

• Although we have proven that average subsethood has some very desirable axiomatic properties, we have more recently switched from average subsethood to another scalar measure of subsethood, proposed by Vlachos and Sergiadis (2007), because it has even better properties than does average subsethood.

• These details are beyond the scope of this talk.

Summary of Obstacles

- Which fuzzy sets to use?
  - IT2
- How to map data about words into a FS model?
  - IA
- What are some CWW engines?
  - NWAs
  - IF-THEN Rules + Perceptual Reasoning
- What are some decoders?
  - Words—similarity
  - Rankings
  - Class—subsethood

• I have now competed discussions about the obstacles to the Perceptual Computer, and how they have been overcome.

• They are summarized on this slide.

Applications for the Per-C

- Investment Judgment Advisor
- Social Judgment Advisor
- Procurement Judgment Advisor
  - Hierarchical decision making
- Journal Publication Judgment Advisor
  - Hierarchical and distributed decision making

34

• To-date, we have applied the Per-C to the four applications that are listed on this slide.

• Before I explain the Investment Judgment Advisor—the only application that I have time for in this talk—let me describe the methodology for designing a Per-C for a specific application.

• This methodology was established as a result of our applying the Per-C to theses four applications, and is one that can be followed when applying the Per-C to a new application.

• For details about all of these applications, see Mendel and Wu (2010).

Methodology: 1

- Focus on an application (*A*)
- Establish a vocabulary (or vocabularies) for *A*
- Collect interval end-point data from a group of subjects (representative of subjects who will use the Per-C) for all words in the vocabulary

35

• The items on this slide are self-explanatory.

• Note, however, that the **vocabulary** should be established in consultation with application experts.

• The same word that is used across different applications may have a different FOU for each application.

Methodology: 2

- Map collected word data into word-FOUs using the IA
  - Result is codebook or codebooks for *A*
  - Completes design of Encoder
- Choose appropriate CWW Engine for *A*
  - It maps IT2 FSs into one or more IT2 FS

36

• The items on this slide are also self-explanatory.

• Multiple codebooks can occur when, e.g., there are different sets of words used for different groups of sub-criteria or antecedents, and for the recommendations.

• So far, only two CWW Engines have been developed by us— NWAs and IF-THEN rules.

Methodology: 3

- If an existing CWW Engine is available for *A*, use its available mathematics to compute its output(s)
- Otherwise, develop such mathematics for your new kind of CWW Engine
  - The new CWW Engine should be constrained so that its output(s) resemble the FOUs in the codebook

37

• Developing new CWW Engines (subject to the new codebook requirement) is a fertile area for research.

Methodology: 4

- Map the IT2 FS outputs from the CWW Engine into a recommendation at the output of the decoder
  - If recommendation is word, rank or class, use existing mathematics to accomplish this mapping
  - Otherwise, develop such mathematics for your new kind of decoder

38

• The items on this slide are also self-explanatory.

An Application: Investment Judgment Advisor

39

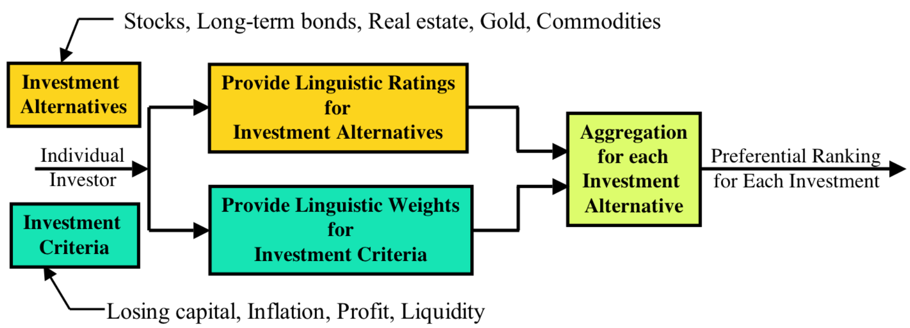
- On the subject of financial advisors ….

40

- A bit of humor.
- Hope you got it!

Investment Judgment Advisor (IJA): 1

Which investment alternatives should be in my portfolio?



41

• This advisor is based on a paper that was first published by Tong and Bonissone (1980).

• An individual has a sum of money that she wants to invest, but not all in one investment alternative.

• She is informed of five investment alternatives: stocks, long-term bonds, real estate, gold and commodities.

• She does not want to invest in so many alternatives, and wants to choose two or three of them for her portfolio.

• She does this by:

1. Providing a linguistic rating for four investment criteria (having to do with losing capital, inflation, profit and liquidity); and,

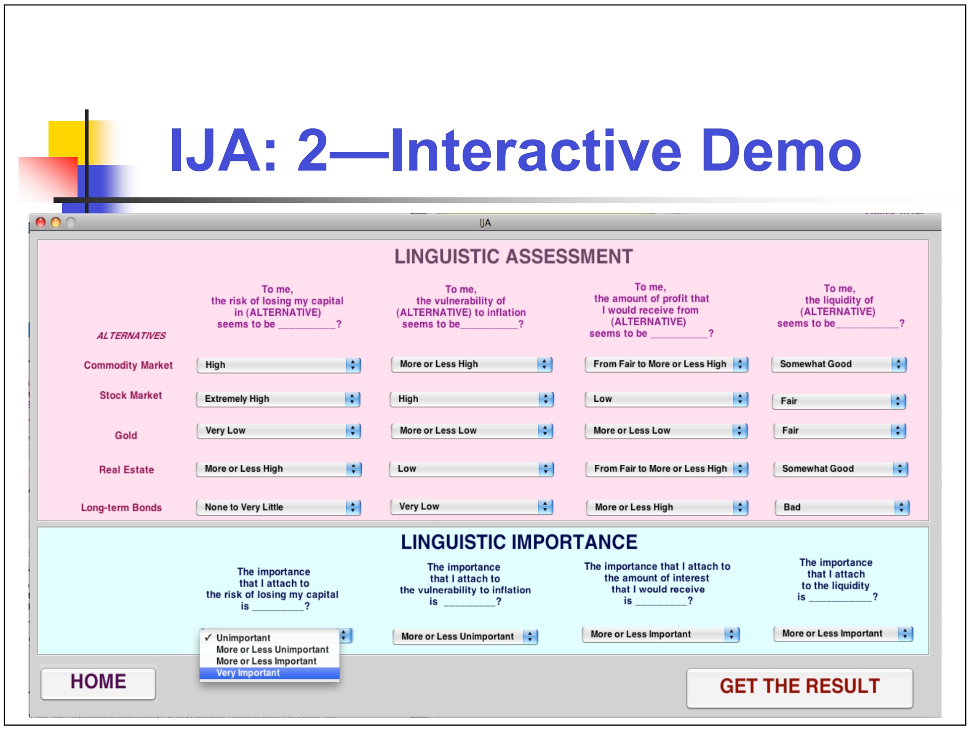
2. Providing linguistic weights for these four investment criteria.

• Item 1 is where she must use her knowledge about each investment alternative.

• Item 2 is where she displays her style as an investor (e.g., speculative, conservative, etc.)

• A LWA is used to aggregate the linguistic ratings with their linguistic weights.

• The resulting 5 LWA FOUs are then ranked.



• We have developed an interactive demo—IJA—one of whose screens is shown on this slide.

• There are pull-down menus from which an individual chooses the word that best reflects either her linguistic assessment or linguistic importance.

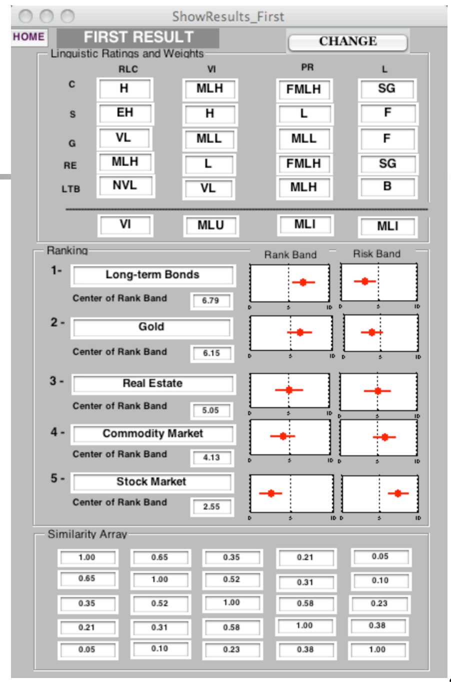
• An example of such a pull-down menu is shown for one of the investment criteria—it has four words.

• The pull-down menus for Linguistic assessment have eight words.

• After the individual completes the two arrays, she hits “GET THE RESULT.”

IJA: 3

- Each investment alternative has an LWA FOU
- Centroid of FOU is computed and is the Rank Band
- Risk considered to be antonym of rank
- If investments are similar above degree *x*, then they can be interchanged



43

• This is another screen shot and is what the individual sees after hitting "GET THE RESULT."

• The top part is a summary of the two linguistic arrays, and uses acronyms for the words.

• The middle part has a lot of information (people not only want a linguistic output from the IJA but also want data to back up the recommendations):

1. Linguistic ranking of the five investment alternatives.

2. Rank Band—the centroid of the FOU of the LWA for each investment alternative.

3. Center of Rank Band, shown as a larger dot in the middle of the Rank Band, and also as a numerical values just below the name of the ranked investment alternative; and,

4. Risk Band—we have assumed that risk can be interpreted as the **antonym** of rank, e.g., high rank implies low risk (the plot or the Risk Band is obtained from that of Rank Band by using *10-Rank Band*).

• Finally, the bottom part is a similarity array, obtained by computing the Jaccard similarity measure for each of the five LWA FOUs with the other FOUs.

• An investment alternative may be similar enough to another one so that it does not really matter which one the investor chooses for her portfolio.

IJA: 4

- Can play what-if games
- Changes are in a different color
- Up to three screens can be shown

44

- The investor can change some or all of her Linguistic Alternatives or Linguistic weights.
- The former might be done with the counseling of a financial advisor, because the investor may not be knowledgeable about some or all of the investment alternatives.
- The latter may also be done with the counseling of a financial advisor, because the investor may not be knowledgeable about the meanings of some of the investment criteria.
- Side-by-side comparisons are displayed to the individual.
- Up-to three side-by-side comparisons can be displayed to the individual.
- The individual may make changes up to 10 times, but only the last three changes will be displayed side-by side.
- Finally, no FOUs are shown to the individual—such FOUs may be of interest to a FL researcher, but are only a means to an end for the individual who interacts with this Per-C.

Conclusions

- Focus has been on obstacles (challenges) to implementing the Per-C and how they have been overcome
- Perceptual computing methodology has evolved
- Perceptual computing is very different from function approximation applications of FL
- This took 10 years

45

• This brings me to the end of my story.

• It has taken my students and I (more than) 10 years to implement Zadeh's CWW paradigm just for the applications of aiding people in making subjective judgments.

• FL has been very widely used in the past, but for most of its applications, the outputs from a fuzzy system have been numbers (i.e., defuzzified fuzzy sets); however, as I hope is now clear to you, this is not the case when FL is used for perceptual computing, where the output is a recommendation.

• Perceptual Computing and CWW are fruitful and very rich areas for much more research.

• I hope that this talk has stimulated you to begin such research.

Acknowledgements

- Nilesh Karnik
- Feilong Liu
- Dongrui Wu
- Jhiin Joo

46

• These are my Ph. D. students who have contributed to the Perceptual Computer, either directly or indirectly.

• Nilesh Karnik developed (among many other things) the widely-used KM algorithms, which are the backbone calculations for many computations used by the Per-C.

• Feilong Lu developed the IA.

• Dongrui Wu developed the LWA, Perceptual Reasoning, Jaccard Similarity measure for IT2 FSs, Ranking of IT2 FSs, and a multitude of other things. He also worked on the following three applications for the Per-C: Social Judgment Advisor, Procurement Judgment Advisor, and Journal Publication Judgment Advisor.

• Jhiin Joo worked on the Investment Judgment Advisor and developed the interactive IJA demo.

Major Reference

- The major reference for this talk is:
  Perceptual Computing: Aiding People in Making Subjective Judgments, J. M. Mendel and D. Wu, IEEE Press and John Wiley, 2010.

47

• This research monograph will be available around the end of March 2010.

• It is totally self-contained.

References

N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, pp. 195-220, 2001.

F. Liu and J. M. Mendel, "Encoding words into interval type-2 fuzzy sets using an *Interval Approach*," *IEEE Trans. on Fuzzy Systems*, vol. 16, pp 1503-1521, December 2008.

J. M. Mendel, "Fuzzy sets for words: a new beginning," *Proc. FUZZ-IEEE 2003*, St. Louis, MO, pp. 37-42, 2003.

J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Information Sciences*, vol. 177, pp. 998-1006, 2007a.

J. M. Mendel, "Type-2 fuzzy sets and systems: an overview,"*IEEE Computational Intelligence Magazine*, vol. 2, pp. 20-29, Feb. 2007b.

J. M. Mendel, "Computing with words: Zadeh, Turing, Popper and Occam," *IEEE Computational Intelligence Magazine*, vol. 2, pp. 10-17, November 2007c.

J. M. Mendel and D. Wu, "Perceptual reasoning for perceptual computing," *IEEE Trans. on Fuzzy Systems*, vol. 16, no. 6, pp. 1550-1564, December 2008.

J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*," John Wiley and IEEE Press, 2010.

J. T. Rickard, J. Aisbett and G. Gibbon, "Fuzzy subsethood for fuzzy sets of type-2 and generalized type-*n*," *IEEE Trans. on Fuzzy Systems*, vol. 17, no. 1, pp. 50-60, 2009.

R. M. Tong and P. P. Bonissone, "A linguistic approach to decision making with fuzzy sets," *IEEE Trans. on Systems, Man, Cybernetics*, vol. 10, pp. 716-723, 1980.

I. Vlachos and G. Sergiadis, "Subsethood, entropy, and cardinality for interval-valued fuzzy sets–an algebraic derivation," *Fuzzy Sets and Systems*, vol. 158, pp. 1384-1396, 2007.

D. Wu and J. M. Mendel, "A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets, *Information Sciences*, vol. 179, no. 8, pp. 1169-1192, 2008.

L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 103-111, 1996.

L. A. Zadeh, "From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions," *IEEE Trans. on Circuits and Systems–1, Fundamental Theory and Applications*, vol. 4, pp. 105-119, 1999.

- This list of reference are only the ones that are called out in the notes.
- See Mendel and Wu (2010) for many more related references.