METHODS & DESIGNS

Computerized fuzzy ratings: The concept of a fuzzy class

TIM HESKETH and BERYL HESKETH

University of New South Wales, Kensington, Australia

In this paper, we outline the concept of a fuzzy class used in programs for eliciting computerized fuzzy ratings. Fuzzy ratings allow respondents to provide symmetrical or asymmetrical latitudes of acceptance around a preferred point. These ratings have been used in research testing career theories and person-environment fit models. The variables defining the fuzzy class and the various functions that can be performed by it are described in the paper. The idea of a fuzzy class may be of interest to those involved in expert systems, knowledge engineering, or in fuzzy classification and measurement in general. A program, FUZRATE, written in object-oriented C++ code that uses the concept of a fuzzy class, is available on request. Both source code and a binary (executable) file are available.

The Fuzzy Rating Scale is a modification of traditional approaches to measurement that allows respondents to indicate both a preferred point on a scale and latitudes of acceptance on either side. (T. Hesketh, Pryor, & B. Hesketh, 1988). Figure 1 provides examples of two fuzzy ratings. In each case, the preferred point, indicated by a V pointer, is extended by horizontal lines that may be symmetrical or asymmetrical. Fuzzy ratings avoid the precision error of traditional measurement (Polkinghorne, 1984), in which the ratings obtained may be more precise than the experience being rated.

The idea underlying the placement of latitudes of acceptance around attitude ratings, and hence fuzzy measurement, was foreshadowed by Thurstone (1922) and Sherif (1976). The application of fuzzy set theory (Zadeh, 1973) and the use of a computerized procedure to elicit ratings has made the implementation of these ideas possible.

In the present paper, we refer to previous uses of the Fuzzy Rating Scale and then describe fuzzy sets and fuzzy variables. The rest of the paper outlines a key programming concept, the fuzzy class, which provides the ability to manipulate and display fuzzy variables. Finally, we present an example of a program, FUZRATE, used in previous research, and other applications of fuzzy ratings and the concept of fuzzy class are suggested.

Previous Applications of Computerized Fuzzy Ratings

Fuzzy ratings have been used in testing Gottfredson's (1981) theory of career compromise, a theory that requires a measure of zones of preference on sex type, prestige, and interest dimensions (B. Hesketh, Elmslie, & Kaldor, 1990). The ratings have also been used in extending ideas of person-environment fit to incorporate the concept of flexibility (B. Hesketh & Gardner, 1993; B. Hesketh, McLachlan, & Gardner, 1992). Research on fit requires that one assess the degree of "match" between two fuzzy ratings, such as those relating to work preferences and job perceptions (see Figure 1).

Fuzzy Sets and Variables

A fuzzy set is a set in which the members may have grades of belonging. Conventionally, members of the set are assigned a number in the range [0,1] to represent the degree of "belonging." For example, various occupations and individuals may have grades of membership of the "realistic" set (Holland, 1985).

A fuzzy variable represents the grades of membership assigned to particular values or levels of an attribute (the fuzzy support). It comprises a set of pairs of numbers. The first number in each pair is a possible value or level for fuzzy support (typically [0,99] in the current applications), and the second number indicates the grade of membership associated with that level or value of the attribute on the fuzzy support. Graphically, the fuzzy variable is drawn as a continuous function with the fuzzy support on the horizontal axis and the membership function on the vertical axis. Membership grades take values in the range [0,1]; 0 indicates no possibility of set membership, and 1 indicates definite membership of the set.

This research was funded by the Australian Research Council; however, the ideas expressed are those of the authors alone. Address correspondence to T. Hesketh, School of Electrical Engineering, The University of New South Wales, P.O. Box 1, Kensington, NSW 2033, Australia (Tel: 612-697-4068; e-mail: t.hesketh@unsw.edu.au).

—Accepted by previous editor, N. John Castellan, Jr.

Figure 1. Fuzzy ratings for work preference and companion job perception.

Figure 2. Representations of a triangular membership function.

The Fuzzy Rating Scale (see Figure 2) is a stylized representation of a fuzzy variable. In the case of the fuzzy rating, the maximum membership function of 1 is associated with the V pointer, with membership tapering uniformly to zero at the end of the extensions. The rationale for this procedure is described in T. Hesketh et al. (1988).

A FUZZY CLASS

A fuzzy class is a data structure that may be included within any application program using fuzzy variables, such as the program FUZRATE (described later). The class declaration defines the variables within the data structure, prototypes of functions that manipulate those variables, and provides an overall outline of the capabilities of the software supporting fuzzy operations.

```
class fuzzy {
private:
    float left, middle, middler, right, L, R;
    float EV,EVt;
    int ev,evt;
    float v[100];
    int number;
    float threshold;
    int x1, x2, y;
    int blips;
    int mode; // 0 = normal, 1 = vptr, 2 = box-whisker

public:
    fuzzy();
```

```
fuzzy(float l, float m, float r);
fuzzy(float a[]);
fuzzy(int a[]);
fuzzy(const fuzzy &);
friend float E(fuzzy&);
float E();

void norm(void);
void fuzzify(void);
void smooth(int);
void set_threshold(float t) { threshold = t; };
void get(int *);

fuzzy& operator =(const fuzzy&);
fuzzy& operator =(float *a);
fuzzy& operator =(int *a);
friend fuzzy operator -(const fuzzy&, const fuzzy&);
friend int operator >(const fuzzy&, const fuzzy&);
friend int operator <(const fuzzy&, const fuzzy&);
friend int operator ==(const fuzzy&, const fuzzy&);
friend fuzzy operator &(const fuzzy&, const fuzzy&);
friend fuzzy operator |(const fuzzy&, const fuzzy&);

void setxy(int, int, int);
void draw(void);
void erase(void);
void no_blips(void) { blips = 0; };
void set_mode(int m) { mode = m; };
int get_mode(void) { return mode; };

int ID() {return number;};
};
```

It is informative to associate the fuzzy class with the various graphic representations of a fuzzy variable, before examining the fuzzy operations defined within the class. Figures 2 and 3 provide two possible representations of membership functions. Variables embodied in a fuzzy class, and used in the figures, are described below.

Fuzzy Support

The fuzzy support comprises 100 evenly spaced points on the x-axis. For purposes of scaling, which relate each of the 100 points to a real measurable value outside the fuzzy construct, the leftmost point is associated with a value *L*, and the rightmost point with a value *R*.

Fuzzy Membership Function

The fuzzy membership function is a set of y-axis values, each in the range [0,1], defined in the array *v*[].

Figure 3. Representations of a four-sided membership function.

Simplified Representations

Simplified representations of a fuzzy variable may be given by specifying either three points on the fuzzy support (*left*, *middle*, and *right*; see Figure 2), or four points (*left*, *middle*, *middler*, and *right*; see Figure 3). The two middle values define the points on the support between which the membership function has the value 1.0. The membership function tapers uniformly from this peak to 0.0 at *left* and *right*. If *middle* = *middler*, the single point so defined is sometimes referred to as the V pointer.

A fuzzy variable specified using the simplified representations may still be represented conventionally. All fuzzy operations are defined for the conventional representation. The simplified representations provide a graphic means for presenting fuzzy variables for purposes of measurement.

Unique Identification

Fuzzy variables are usually allocated space in the free area of memory. When a fuzzy variable is created, it is given a unique identifier (*number*).

Expected Value

The expected value is a single-value estimate of a fuzzy variable. Two expected values may be calculated. *EV* is the conventional expected value, defined below, whereas *EVt* is the expected value calculated after values of the membership function less than *threshold* have been set to 0.

$$EV = \sum_{k=0}^{99} v[k] \cdot w$$

$$w = L + \frac{k}{99}(R - L)$$

Two flags, *ev* and *evt*, indicate whether the calculation of the expected value is "up to date." The flags are cleared after any operation on the fuzzy variable that may change the expected value. If either flag is set, the corresponding expected value is "up to date," and may be used without recalculation.

FUNCTIONS PERFORMED ON THE FUZZY CLASS

The fuzzy class permits the construction of a fuzzy variable, the calculation of an expected value (best single number representing the fuzzy variable), normalization, fuzzification, and smoothing of variables. Fuzzy operators and fuzzy graphics are also part of the functions performed by the fuzzy class.

In the following text, a consistent format is used to describe the various functions that can be performed on a fuzzy variable. First the syntax is given, then an example of its use is provided and, finally, the operations it performs are outlined.

Constructors

The constructor functions are called to initialize a fuzzy variable any time it is declared.

Syntax

```
fuzzy();
fuzzy(float,float,float);
fuzzy(float *);
fuzzy(int *);
fuzzy(const fuzzy &);
```

Examples of Use

```
fuzzy A; // definition 1
fuzzy B(30,40,40,70); // definition 2
float a[] = { 100 values }; fuzzy C(a); // definition 3 (4 similar)
fuzzy D(A); // definition 5
fuzzy *P = new fuzzy(A); // memory allocation
```

Operation. A fuzzy variable may be defined with values initialized to suitable default values (Definition 1), or defined by a simplified (*left*, *middle*, *middler*, *right*) specification (Definition 2), specified by a fuzzy membership function containing 100 values (Definitions 3 and 4), or using a copy constructor, when the fuzzy variable is initialized to be the same as some previously defined fuzzy variable (Definition 5).

The function *get*(*int* *) allows retrieval of the *left*, *middle*, *middler*, and *right* values once a fuzzy variable has been initialized.

Calculation of Expected Value

Syntax

```
float E(fuzzy &);
float E();
```

Examples of Use

```
fuzzy A(30,40,40,70);
expected_value = A.E(); // true expected value
expected_value_threshold = E(A); // expected value with threshold
```

Operation. The expected value of the fuzzy variable (defined previously) is calculated. Two forms of function are provided. The first calculates the true expected value, taking into account all values along the entire support, and the second sets all membership function values below the threshold to 0, before calculating the expected value.

Calculation of the expected value only takes place if the flags *ev* and *evt* are not set. If the flags are set, then precalculated values of the expected value are returned. The flags are set when the expected values are calculated, and cleared after any operation that could result in a change to the expected value. For this function, the return value is the expected value.

Normalization

Syntax

```
void norm(void);
```

Example of Use

```
a[] = { 100 values }; fuzzy B(a);
B.norm();
```

Operation. The minimum value of the membership function is set to 0.0, and the maximum to 1.0. Other values of the membership function are scaled to lie between these two extremes.

Fuzzification

Syntax

```
void fuzzify(void);
```

Example of Use

```
fuzzy A(30,40,40,70);
A.fuzzify();
```

Operation. This forces calculation of the fuzzy membership function, defined in the private array $v[]$. This function is seldom required, as it is invoked automatically by the relevant fuzzy constructors.

Smoothing the Membership Function

Syntax

```
void smooth(int);
```

Example of Use

```
a[] = { 100 values }; fuzzy A(a);
A.smooth(7);
```

Operation. A rank-order filtering operation is invoked to smooth the membership function. The integer argument is the size of the filter window.

Fuzzy Operators

Syntax

```
fuzzy& operator =(const fuzzy&);
fuzzy& operator =(float *a);
fuzzy& operator =(int *a);
friend fuzzy operator -(const fuzzy&, const fuzzy&);
friend int operator >(const fuzzy&, const fuzzy&);
friend int operator <(const fuzzy&, const fuzzy&);
friend int operator ==(const fuzzy&, const fuzzy&);
friend fuzzy operator &(const fuzzy&, const fuzzy&);
friend fuzzy operator |(const fuzzy&, const fuzzy&);
```

Examples of Use

```
float a[] = { 100 values };
int b[] = {30,40,40,70};
fuzzy A,B,C;
A = a;
B = b;
C = A;
C = A - B;
if (A > B) { ... };
if (A == B) { ... };
C = A & B;
C = A | B;
```

Operation. The operators “overload” the normal C operators of the same notation. Overloading means that the newly defined operators will be used in place of the

original operators for arguments of the appropriate type. The operators are defined according to normal C syntax: replacement (=), subtraction (-), greater than (>), less than (<), equal to (==), fuzzy AND (&), and fuzzy OR (|).

Various operations are defined as follows. If A and B share the same support, so that

$$A = \{X(i), A(i) | i = 0:99\};$$

$$B = \{X(i), B(i) | i = 0:99\};$$

then

$$C = A \& B = \{X(i), \text{MIN}(A(i), B(i)) | i = 0:99\};$$

$$C = A | B = \{X(i), \text{MAX}(A(i), B(i)) | i = 0:99\};$$

$$C = A \pm B = \{X(i) \pm X(j), \text{MIN}(A(i), B(j)) | i = 0:99; j = 0:99\};$$

$$(A > B) = ((A - B) > 0)?1:0;$$

$$(A < B) = ((A - B) < 0)?1:0;$$

$$(A == B) = ((A - B) == 0)?1:0;$$

Fuzzy Graphics

Syntax

```
void setxy(int, int, int);
void draw(void);
void erase(void);
void no_blips(void) { blips = 0; };
void set_mode(int m) { mode = m; };
int get_mode(void) { return mode; };
```

Examples of Use

```
fuzzy A(30,40,40,70);
A.setxy(100,540,100);
A.set_mode(0);
A.draw();
A.erase();
```

Operation. Before drawing a fuzzy variable, it is necessary to define its position on the screen (defined by the private variables $x1$, $x2$, and y). The three arguments to $setxy()$ are screen coordinates for placement of the left and right extremes of the support, and the y -coordinate of the x -axis. The vertical size is dependent on the size of the x -axis representation. Suitable lines are drawn for the axes. The function $draw()$ will produce a representation of a fuzzy variable on the screen, in rectangular x, y coordinates, with the support along the x -axis and the membership function as a function $y = f(x)$.

A full graphic representation of the fuzzy variable is drawn if Mode 0 is set, whereas Modes 1 and 2 produce the simplified representations shown in Figures 2 and 3.

The function $erase()$ will completely erase the representation of the fuzzy variable from the screen, including the representation of the axes, if Mode 0 is set. For Modes 1 and 2, the axes are not erased. This allows the fuzzy representation to be dynamically changed on the screen during the interactive definition of fuzzy variables.

By default, the horizontal axis is divided into seven equally spaced regions (which relate to the use of the fuzzy class for eliciting measures based on semantic differen-

Table 3 (Continued)

Rhyme Pair	Total Set Size			First Response Set Size		
	Index of Commonality	No. of Common Pairs	<i>r</i>	Index of Commonality	No. of Common Pairs	<i>r</i>
King Ring	.97	17	.94	.62	5	.78
Lake Snake	.82	18	.82	.78	7	.72
Length Strength	1.00	1		1.00	1	
Lice Spice	.85	14	.80	.57	4	.55
Life Wife	1.00	5	.86	.89	4	.12
Madness Sadness	.67	2	-1.00	.67	2	-1.00
Middle Riddle	.80	6	.95	.77	5	.96
Plank Tank	1.00	20	.80	.75	9	.02
Rubble Trouble	.89	4	.86	.86	3	.80
Saloon Typhoon	.71	18	.24	.42	4	.06
Spree Tree	.88	27	.80	.58	7	.18
Thicket Ticket	1.00	5	.67	.89	4	.00

*No rhyme words were generated in response to *contract*.

REFERENCES

- ASTHANA, B. B., & GOEL, R. (1987). Retention under matched input and output context. *Psycho-Lingua*, **17**, 55-70.
- BADDELEY, A. D. (1964). Semantic and acoustic similarity in STM. *Nature*, **204**, 1116-1117.
- BADDELEY, A. D., & SALAMÉ, P. (1986). The unattended speech effect: Perception or memory? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **12**, 525-529.
- BESNER, D., & DAVELAAR, E. (1982). Basic processes in reading: Two phonological codes. *Canadian Journal of Psychology*, **36**, 701-711.
- BISHOP, D. V. M., & ROBSON, J. (1989). Unimpaired short-term memory and rhyme judgement in congenitally speechless individuals: Implications for the notion of "articulatory coding." *Quarterly Journal of Experimental Psychology*, **41A**, 123-140.
- BOUSFIELD, W. A., & WICKLUND, D. A. (1969). Rhyme as a determinant of clustering. *Psychonomic Science*, **16**, 183-184.
- CAMDEN, C., & MOTLEY, M. (1982). Synonymous rhymes: Stimuli for psycholinguistic research. (From *JSAS Catalog of Selected Documents in Psychology*, **12**, 5, Abstract No. 2412)
- CROWDER, R. G. (1971). The sounds of vowels and consonants in immediate memory. *Journal of Verbal Learning & Verbal Behavior*, **10**, 587-596.
- DICKEL, M. J. (1979). *Arousal and the verbal discrimination learning of acoustically and semantically related lists*. Unpublished master's thesis, Central Michigan University, Mt. Pleasant, MI.
- DORNIC, S., & FERNACUS, S. E. (1987). *Type of processing in high-load tasks: The differential effect of noise*. (Report No. 576). Stockholm, Sweden: University of Stockholm, Department of Psychology.
- DUNLAP, G. L., & DUNLAP, L. L. (1979). Manipulating the word frequency effect in free recall. *Memory & Cognition*, **7**, 420-425.
- ELING, P. (1983). Consistency of ear advantage: An improvement due to increase in presentation rate. *Neuropsychologia*, **21**, 419-423.
- EYSENCK, M. W. (1981). Learning, memory, and personality. In H. J. Eysenck (Ed.), *A model for personality* (pp. 169-209). New York: Springer-Verlag.
- FISHER, R. P., & CRAIK, F. I. M. (1977). Interaction between encoding and retrieval operations in cued recall. *Journal of Experimental Psychology: Human Learning & Memory*, **3**, 701-711.
- FLAVEL, J. H., DRAGUNS, J., FEINBERG, L. D., & BUDIN, W. (1958). A microgenetic approach to word association. *Journal of Abnormal & Social Psychology*, **57**, 1-7.
- FURST, B. (1948). *Stop forgetting*. Garden City, NY: Garden City Books.
- HAGAN, M. L. (1979). *Encoding and retrieval: Type and strength of relationship*. Unpublished master's thesis, Illinois Institute of Technology, Chicago.
- HORTON, K. D. (1983). Phonemic and semantic generation norms. *Behavior Research Methods & Instrumentation*, **15**, 109-110.
- HUNT, R. R., & MITCHELL, D. B. (1978). Specificity in non-semantic orienting tasks and distinctive memory traces. *Journal of Experimental Psychology: Human Learning & Memory*, **4**, 121-135.
- JOHNSON, W. L., & CHAMBERLIN, C. M. (1970). Semantic and acoustic similarity and their effects on short-term retention. *Psychonomic Science*, **19**, 217-218.
- JUNG, C. G. (1918). *Studies in word association*. London: William Heinemann.
- KING, D. J., & FORRESTER, W. E. (1970). The influence of semantic and acoustic similarity on retention of short sentences. *Psychonomic Science*, **20**, 222-223.
- KOZLOWSKI, L. T. (1977). Effects of distorted auditory and of rhyming cues on retrieval of tip-of-the-tongue words by poets and nonpoets. *Memory & Cognition*, **5**, 477-481.
- KRAMER, A. F., & DONCHIN, E. (1987). Brain potentials as indices of orthographic and phonological interaction during word matching. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **13**, 76-86.
- KUČERA, H., & FRANCIS, W. N. (1967). *Computational analysis of present-day American English*. Providence, RI: Brown University Press.
- LISIECKI, P. M. (1978). Evidence for search in lexical memory listing a forced-choice recognition task. *Dissertation Abstracts International*, **39**, 6162-B. (University Microfilms No. 79-13, 103)

- LUPKER, S. J. (1982). The role of phonetic and orthographic similarity in picture-word interference. *Canadian Journal of Psychology*, **36**, 349-367.
- MUELLER, J. H. (1978). The effects of individual differences in test anxiety and type of orienting tasks on levels of organization in free recall. *Journal of Research in Personality*, **12**, 100-116.
- NELSON, D. L. (1979). [Rhyming responses to 1,001 words]. Unpublished norms.
- NELSON, D. L. (1981). Many are called but few are chosen: The influence of context on the effects of category size. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 15, pp. 129-162). New York: Academic Press.
- NELSON, D. L., & BROOKS, D. H. (1974). Retroactive inhibition of rhyme categories in free recall: Inaccessibility and unavailability of information. *Journal of Experimental Psychology*, **102**, 277-283.
- NELSON, D. L., & McEVOY, C. L. (1979). Encoding context and set size. *Journal of Experimental Psychology: Human Learning & Memory*, **5**, 292-314.
- NELSON, D. L., McEVOY, C. L., & FRIEDRICH, M. A. (1982). Extralist cuing and retrieval inhibition. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **8**, 89-105.
- NELSON, D. L., REED, V. S., & McEVOY, C. L. (1977). Encoding strategy and sensory and semantic interference. *Memory & Cognition*, **5**, 462-467.
- NELSON, D. L., WHEELER, J. W., BORDEN, R. C., & BROOKS, D. H. (1974). Levels of processing and cuing: Sensory versus meaning features. *Journal of Experimental Psychology*, **103**, 971-977.
- NOBLE, C. E. (1952). An analysis of meaning. *Psychological Review*, **59**, 421-430.
- PAIVIO, A., YUILLE, J. C., & MADIGAN, S. A. (1968). Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of Experimental Psychology Monographs*, **76** (1, Pt. 2).
- PARKIN, A. J. (1981). Determinants of cued recall. *Current Psychological Research*, **1**, 291-300.
- PAYNE, D. G., NEELY, J. H., & BURNS, D. J. (1986). The generation effect: Further tests of the lexical activation hypothesis. *Memory & Cognition*, **14**, 246-252.
- RABINOWITZ, J. C., & CRAIK, F. I. [M.] (1986). Specific enhancement effects associated with word generation. *Journal of Memory & Language*, **25**, 226-237.
- RUBIN, D. C. (1990). Directed graphs as memory representations: The case of rhyme. In R. W. Schvaneveldt (Ed.), *Pathfinder associative networks: Studies in knowledge organization* (pp. 121-133). Norwood, NJ: Ablex.
- SHAW, R. J., & CRAIK, F. I. [M.] (1989). Age differences in predictions and performance on a cued recall task. *Psychology & Aging*, **4**, 131-135.
- SPRING, E., & PERRY, L. (1983). Naming speed and serial recall in poor and adequate readers. *Contemporary Educational Psychology*, **8**, 141-145.
- THORNDIKE, E. L., & LORGE, I. (1944). *The teachers word book of 30,000 words*. New York: Teachers College Press, Columbia University.
- UNDERWOOD, B. J., & SCHULTZ, R. W. (1960). *Meaningfulness and verbal learning*. Philadelphia: Lippincott.
- WIEGEL-CRUMP, C. A., & DENNIS, M. (1986). Development of word finding. *Brain & Language*, **27**, 1-23.
- WOOD, C. (1943). *Wood's unabridged rhyming dictionary*. Cleveland: World Publishing.

APPENDIX A
Rhyme (R), Imagery (I), Concreteness (C),
Meaningfulness (m), Frequency (F^a), and
Overlap (O) Values for 477 Target Words

Target Words	R		I	C	m	F ^a	O
	M	SD					
Abess	.00	.00	2.97	4.78	2.48	1	
Abode	1.54	1.52	5.07	6.31	6.64	16	.10
Air	6.64	1.96	4.17	5.67	6.36	AA	.21
Alligator	.48	.73	6.87	6.93	7.72	6	.04
Amour	.88	1.65	5.50	2.65	5.08	1	.04
Anger	.17	.38	4.87	1.70	5.83	A	.17
Angle	3.18	1.31	5.50	5.19	6.24	30	.40

APPENDIX A (Continued)

Target Words	R		I	C	m	F ^a	O
	M	SD					
Ankle	.12	.33	6.77	7.00	6.82	21	.12
Annual	1.08	.70				44	.27
Answer	2.31	1.12	2.77	4.49	6.04	AA	.46
Apple	.68	.77	6.73	7.00	7.67	A	.23
Arm	2.40	1.35	6.53	6.96	6.92	AA	.60
Army	.00	.00	6.53	6.55	6.68	AA	
Arrow	1.93	1.11	6.57	7.00	6.80	37	.39
Artist	.21	.53	5.93	6.14	6.65	A	.10
Attic	.45	.69				12	.09
Baby	.82	.39	6.70	6.90	7.04	AA	.82
Bag	6.46	1.70				AA	.43
Bald	1.94	1.31				9	.16
Band	3.36	1.86				A	.19
Bandit	.00	.00	5.83	6.28	6.92	5	
Banker	1.04	.64	6.03	6.38	6.56	27	.26
Banner	1.74	1.03	5.93	6.58	6.20	23	.29
Barrel	1.16	.94	6.57	6.94	6.16	32	.23
Beast	2.98	1.48	5.90	6.51	6.08	A	.20
Beat	5.61	1.70				AA	.17
Beaver	1.36	1.19	6.50	6.96	8.19	23	.23
Beer	6.56	1.82				23	.23
Beggar	.02	.15	6.40	6.25	6.50	29	.02
Belief	2.20	1.03	2.73	1.55	5.24	39	.22
Beverage	.67	.48	5.87	5.96	6.58	8	.67
Bird	2.52	1.33	6.67	6.96	7.88	AA	.18
Blisters	1.91	.73	6.53	6.67	7.13	6	.34
Blood	2.21	1.12	6.70	6.82	6.56	AA	.32
Bloom	3.82	1.75	5.63	5.82	5.12	35	.21
Blossom	.53	.50	6.67	6.62	7.60	43	.53
Board	4.10	1.83	6.07	6.87	6.92	AA	.18
Body	.51	.77	6.40	6.58	5.61	AA	.13
Book	4.34	1.28	6.43	6.96	7.68	AA	.40
Boredom	.00	.00	3.83	1.94	4.63	AA	
Boss	3.22	1.30	5.73	6.35	5.81	23	.40
Bottle	.40	.55	6.57	6.94	7.24	A	.10
Boulder	3.32	1.01	6.13	6.96	5.88	5	.47
Bouquet	.92	.80	6.77	6.55	5.76	8	.46
Bowl	3.11	1.54	6.30	6.90	7.24	A	.14
Boy	3.41	1.24	6.57	6.93	6.96	AA	.22
Brain	5.21	1.81	5.93	6.63	6.80	A	.11
Breast	3.71	1.87	6.77	6.80	5.81	A	.18
Breeze	4.07	2.07	5.87	5.94	7.33	29	.18
Bronze	.20	.45	6.17	6.59	6.19	19	.06
Brute	4.06	1.78	5.17	5.44	6.08	14	.18
Butter	3.04	1.47	6.57	6.96	6.91	AA	.28
Cabin	.00	.00	6.47	6.96	7.20	A	
Calm	1.78	.95				A	.30
Camp	3.47	1.61	6.57	6.56	7.17	AA	.27
Candy	3.66	1.21	6.63	6.56	6.39	32	.52
Cane	4.80	1.79	6.43	6.93	7.48	19	.13
Car	4.66	1.73	6.87	7.00	6.38	AA	.33
Card	2.73	1.30				A	.21
Carrot	1.00	.92				9	.33
Cart	4.77	1.51				27	.43
Cash	6.20	1.16	6.17	6.28	7.21	46	.31
Cast	3.10	1.31				A	.28
Castle	1.55	1.24				A	.52
Cat	6.55	1.68	6.80	7.00	6.76	AA	.38
Cattle	2.85	1.10	6.40	6.79	7.56	A	.36
Cell	4.87	1.89	6.47	6.63	7.52	A	.35
Cellar	1.45	1.14	6.27	6.83	6.79	32	.21
Ceremony	.00	.00	5.37	4.58	5.17	25	
Chair	5.98	2.15	6.63	7.00	7.20	AA	.19
Chance	3.12	1.35	2.50	1.51	5.61	AA	.21
Chant	2.51	1.32				16	.25
Charm	2.55	.98	4.70	2.17	6.13	A	.51
Chart	4.50	1.32				15	.41
Charter	3.16	1.50	3.83	5.05	4.89	20	.29

tials). The x -axis markers will not be drawn if *no-blips()* has been called.

FUZRATE

FUZRATE is a PC program that elicits fuzzy ratings interactively and includes the fuzzy class described above. It makes use of the functions defined within the class to support fuzzy graphics and manipulation of fuzzy variables. FUZRATE provides fundamental support for any research program using fuzzy ratings.

The program begins with initial dialogue that asks for the specification of disk file names. The presentation of the prompts for the ratings is governed by an ASCII text file, which provides the text prompts that are displayed above, below, and to each side of each fuzzy variable displayed on the screen, and incidentally defines the number of ratings in a session. The file may be created by using any standard editor; simple rules of syntax must be adhered to in its creation. The results of the rating will be written to another disk file at the end of the rating session, together with a subject identifier, for subsequent analysis.

Each rating screen provides the following information: an instructional header describing how the rating should be carried out; a statement that is the subject of the rating; prompts that appear at the left, right, and below the fuzzy variable, which describe the meaning of possible ratings; and a menu box enabling selection of the V pointer, left or right extension, and the previous or next rating screen. Selection from the menu, or alteration of the rating itself, may be carried out by using the normal "arrow" keys, or with the aid of a mouse.

Companion programs are available for comparative presentation of fuzzy variables from previous rating sessions for feedback or counseling purposes, for presentation of the fuzzy variables in alternative graphic forms, and for preparation of postscript files for graphic output to printers. Programs have been written for specific analyses of data for particular experiments using the capabilities of the fuzzy class for manipulating fuzzy variables.

Potential Uses of Fuzzy Ratings and the Concept of Fuzzy Class

Fuzzy ratings have potential in areas such as marital counseling, in which both partners can complete a set of scales and the fuzzy ratings provided by the two partners can be redisplayed on the computer screen and explored for potential areas of overlap. Fuzzy ratings can be used for job analysis, rating of selection interviews, and for performance ratings. With respect to performance ratings, the value of the fuzzy ratings lies in allowing a supervisor to indicate best and worst performance as well as typical or modal performance. This permits a more honest and accurate rating, which may ease the process of giving feedback to those rated poorly because good performance is also acknowledged.

Fuzzy ratings may also prove valuable in evaluating training courses (in which changes in consistency may be

as important as increases in modal performance), in knowledge engineering for the development of expert systems, and in the recording of pain or other subjective states. Fuzzy measurement fosters a new way of thinking about traditional measurement. For example, fuzzy measurement underlies the development of alternative scales on the Strong Interest Inventory (B. Hesketh, T. Hesketh, Hansen, & Gorenson, 1993).

The fuzzy class described above serves as a starting point for software for a variety of purposes. Fuzzy logic is used for implementing control systems (Kienitz, 1993), in which the definition of membership functions that govern control actions provides the means for "tuning" the controller. Other applications for which fuzzy concepts have been found useful include approximate reasoning in artificial intelligence (Hirota & Pedrycz, 1993) and fuzzy decision making (e.g., Pedrycz, 1993).

Availability of Code

The software described in this paper (executable program and code) is available from T. Hesketh, School of Electrical Engineering, The University of New South Wales, P.O. Box 1, Kensington, NSW 2033, Australia. Requests should be accompanied by a blank, formatted diskette (either 5.25 or 3.5 in.). E-mail requests (t.hesketh@unsw.edu.au) would be preferred.

The programs have been compiled on PCs using a Zortech compiler. Only the graphics code (confined within a single graphics class) should need to be changed to enable another compiler to be used. FUZRATE comprises a single executable file of approximately 150K. A PC with graphics capability supported by the compiler is all that is required.

REFERENCES

- GOTTFREDSON, L. S. (1981). Circumscription and compromise: A developmental theory of occupational aspirations. *Journal of Counseling Psychology*, *28*, 545-579.
- HESKETH, B., ELSLIE, S., & KALDOR, W. (1990). Career compromise: An alternative account to Gottfredson's theory. *Journal of Counseling Psychology*, *37*, 49-56.
- HESKETH, B., & GARDNER, D. (1993). Person-environment fit models: A reconceptualization and empirical test. *Journal of Vocational Behavior*, *42*, 315-332.
- HESKETH, B., HESKETH, T., HANSEN, J. I., & GORENSEN, D. (1993). *The use of fuzzy variables in developing new scales from the Strong Interest Inventory*. Manuscript submitted for publication.
- HESKETH, B., McLACHLAN, K., & GARDNER, D. (1992). Work adjustment theory: An empirical test using a fuzzy rating scale. *Journal of Vocational Behavior*, *40*, 318-337.
- HESKETH, T., PRYOR, R., & HESKETH, B. (1988). An application of a computerized fuzzy graphic rating scale to the psychological measurement of individual differences. *International Journal of Man-Machine Studies*, *29*, 21-35.
- HIROTA, K., & PEDRYCZ, W. (1993). Matching fuzzy quantities. *IEEE Transactions on Systems, Man, & Cybernetics*, *21*, 1580-1586.
- HOLLAND, J. L. (1985). *Making vocational choices: A theory of vocational personalities and work environments*. Englewood Cliffs, NJ: Prentice-Hall.
- KIENITZ, K. H. (1993). Controller design using fuzzy logic: A case study. *Automatica*, *29*, 549-555.
- PEDRYCZ, W. (1993). A referential scheme of fuzzy decision making and its neural network structure. *IEEE Transactions on Systems, Man, & Cybernetics*, *21*, 1593-1604.

- POLKINGHORNE, D. E. (1984). Further extensions of methodological diversity for counseling psychology. *Journal of Counseling Psychology*, **31**, 416-429.
- SHERIF, C. W. (1976). *Orientation in social psychology*. NY: Harper & Row.
- THURSTONE, L. L. (1922). Attitudes can be measured. *American Journal of Sociology*, **33**, 529-554.
- ZADEH, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, & Cybernetics*, **SMC-3**, 28-44.

(Manuscript received May 27, 1993;
revision accepted for publication October 9, 1993.)