Convenient Interactive Computing for Coherent Imprecise Prevision Assessments

JAMES DICKEY
University of Minnesota, USA

Abstract

A generalization of deFinetti's Fundamental Theorem of Probability facilitates coherent assessment, by iterated natural extension, of imprecise probabilities or expectations, conditional and unconditional. Point values are generalized to assessed bounds, accepted under weak coherence, that is, allowing the input of redundant loose bounds. The method is realized in a convenient interactive computer program, which is demonstrated here, and made available as open source code. This work suggests that a consulting expert's fees should not be paid unless his/her assessed probabilities cohere.

Keywords

assessment, imprecise probabilities, previsions, coherence, natural extension, interactive computing

1 Introduction

We consider previsions of random quantities, loosely, expectations of random variables, a probability being the prevision of an event, or 0-1 random quantity. Prevision assessments can either be intended as estimates of frequencies, more generally averages, or they can be intended as mere quantitative expressions of human uncertainty. In either case, they should be coherent, that is, extendible to at least one full probability distribution. For estimates of frequencies or averages to be taken seriously, this says that their values must not be impossible when interpreted together as limiting frequencies or limiting averages in an experiment. They can describe a conceivable, possibly infinite, population. For previsions intended as expressions of uncertainty, coherence is a kind of rationality, a direct generalization of non-contradiction for statements of fact, a self-consistency in the sense that, if taken as a person's betting prices, the person could not be made a sure-loser merely by combining a finite number of bets at such prices.

2 Coherent Assessment by Iterated Natural Extension

It is becoming more widely known that deFinetti's Fundamental Theorem of Probability [12, 13] provides a dynamic for interactive computational assessment of coherent previsions. For a sequence of mathematically related random quantities (including logically related events), if coherent prevision values are given for an initial segment of the sequence, the available cohering values for the prevision of the next quantity comprise an interval whose endpoints can be computed by linear programming (first noted by Boole [2], Hailperin [14], and Bruno and Gilio [6]). Walley [22] calls this interval the "natural extension" of the given coherent previsions.

The linear-programming variables are interpretable as the probabilities of the "constituent" events, the events of the joint-range points of the random quantities. Coherence restricts the prevision vector of the quantities to the convex hull of the joint- range set, that is, the prevision point must be some weighted average of the join-range points. The assessed previsions impose additional linear constraints.

In textbook-type problems, where a probability is determined by given probabilities, the extension interval reduces to a single value. If the given values, themselves, are not coherent, the linear programming calculation will so indicate by reporting that there are "no feasible solutions," which implies an empty extension interval. Coherent previsions are always capable of being extended coherently with the value for any further random quantity assignable in an extend-assess cycle. If supplementary calculations are made of the extension interval for a random quantity of special interest, the interval will be seen to shrink to a subinterval whenever a further coherent prevision is assessed.

The method generalizes to include conditional previsions, as inputs and/or outputs. In addition, since prevision is a linear operator, a linear combination of previsions can be assessed directly as the prevision of a linear combination of random quantities. For example, if the assessor defines the difference of two events as a random quantity, then the difference of their probabilities can be assessed as a prevision, and so included in the analysis. The convenience of the method suggests that any consulting expert should not be paid unless her/his probability assessments cohere.

3 Coherence for Imprecise Assessments

An interval, or even a single bound, generalizes a point value, and experts may only be willing to report such imprecise previsions. So how do the coherence concept and the iterated extend-assess algorithm generalize to handle imprecise previsions?

If mere bounds are input, instead of precise values, the output extension interval consists of all the available values for the further prevision for which there exists at least one mutually coherent list of precise values satisfying the input bounds. And, of course, for each such precise list, the corresponding cohering values for the further random quantity would form a subinterval of the output interval. This was defined as the problem of probability logic by Hailperin [15] (following Boole [2]), included as "natural extension" by Walley [22], and presented in a generalization of deFinetti's Fundamental Theorem by Lad, Dickey, and Rahman [18, 19]. The latter two papers are the basis for the algorithm coded in the present program. A prototype program written in Mathematica in 1991 has had limited distribution.

So, what assessed further bounds should one say "cohere" with the output extension interval?

Definition 1 *(Weak Coherence) Assessed bounds that do not contradict the output bounds will be said to cohere weakly with the given input bounds. An assessed lower (upper) bound must not lie above (below) the output upper (lower) bound, that is, the assessed interval must overlap the extension interval. Also, of course, an assessed lower (upper) bound must not be higher (lower) than the corresponding assessed upper (lower) bound. Weak coherence is directly equivalent to the prevention of sure-loss combined bets.*

Definition 2 *(Strong Coherence) Assessed bounds that neither contradict, in the weak-coherence sense, nor relax the output bounds will be said to cohere strongly with the given input bounds. So, in addition, an assessed lower (upper) bound must not lie below (above) the output lower (upper) bound, that is, the assessed interval must be a subinterval of the extension interval.*

P. Walley [22] uses the term "coherence" to refer to strong coherence, with the interpretation that an assessed lower (upper) value is asserted as the highest (lowest) agreeable relative purchase (selling) price for the random quantity scaled in monetary units, an interpretation under which dynamic refinement of assessed previsions would seem less than natural. Whereas, weakly coherent buying (selling) prices can be interpreted as conservative purchase offers (offers to sell) that can be refined upward (downward). The weak version of coherence was termed "g-coherence" by Biazzo and Gilio [3]. Weak coherence is relevant to our program, for if a user chooses a bound that is a relaxation of the latest extension interval, it has no effect on any subsequent computed interval. Being subject to later refinement, it need not be the tightest bound, now.

In a trivial mathematical sense, the order in which assessments are made does not matter. If an expert asserts the same coherent bounds in a different order, then the same coherent joint bounds will result. (The tightest implied bounds prevail, of course.) In a practical sense, however, ones psychological reaction to encountering different computed intervals for a different order can make a substantial difference in ones assessed values or bounds.

Relevant further references on coherence and coherence methods for imprecise unconditional and conditional previsions, as suggested by referees, include [1], [7], [8], [9], [10], [11], [16].

4 Implementation

This is to introduce an interactive computer program for coherent assessment of imprecise previsions by iterated coherent extension, in which the user communicates with the program through a combined input-output text file. The interaction proceeds as a series of steps, each in the form of an extend-assess cycle:

1. Based on all the prevision bounds assessed so far, the program computes natural extensions, the implied extension interval(s), for the previsions of one or more user-selected quantities.

2. The user assesses a lower and/or upper bound (or a point value) for a prevision, cohering with its computed extension interval.

4.1 Algorithm

To calculate the extension interval for the unspecified prevision of a quantity, say $p_n = P(X_n)$, the program must determine the convex hull of the joint range set of the considered quantities, and then impose the linear constraints of the assessed prevision values and bounds. Denote by $\mathbf{X}\,(n \times 1)$ the vector of n quantities, $R\,(n \times N)$ the matrix of N joint-range points, and $\mathbf{C}\,(N \times 1)$ the vector of N "constituent" events (joint point-value events). Then $\mathbf{C}$ is a partition, and $\mathbf{X} = R\mathbf{C}$. The convex hull of the set of columns of R is the set of all convex combinations,

$$\mathbf{p} = R\mathbf{q}, \tag{1}$$

where $\mathbf{q} \geq \mathbf{0}$ and $\mathbf{1}^{\mathbf{T}}\mathbf{q} = \mathbf{1}$. Now, suppose our assessments impose the further constraints,

$$A\mathbf{p} \leq \mathbf{b},$$

some of the inequalities of which may be equalities. The prevision variable to be optimized is $p_n = \mathbf{r}_{\mathbf{n}}^{\mathbf{T}}\mathbf{q}$, from Eq. (1). This fully defines the relevant linear-programming calculations.

The steps to achieve this construction and calculation are:

1. Define the product quantities needed for any conditional previsions considered.

2. Define subroutines to reject the potential columns of R that do not satisfy the logical and mathematical constraints on $\mathbf{X}$.

3. Border R for any new random quantities, or start over to reconstruct R from scratch if any old quantities are omitted or redefined.

4. For each prevision to be optimized, form a linear-programming input file and run the routine lp-solve. (Perform a change-of-variables if a conditional prevision is to be optimized.)

4.2 Zero Probabilties

A coherent prevision conditional on an event of zero probability is not determined by the usual unconditional previsions: if $P(A) = 0$, then $P(XA) = 0$ and $P(X|A) = 0/0$, which is indeterminate. Nor can such a conditional prevision have any coherent effect on unconditional prvisions: if $P(A) = 0$, then $P(X) = P(X|A)P(A) + P(X|nA)P(nA) = P(X|nA)$. So, although the program can accept, as input, prevision assessments that are conditional on an event of probability zero, as presently coded, it will not respond to a request to calculate extension bounds on such a prevision. The practical reason for this is that the program solves the fractional-programming problem for a bound on conditional prevision by a change-of-variable that divides by $P(A)$. Improvements in this aspect of the program are contemplated.

4.3 Input/Output

The combined input/output file is organized as a sequence of records, or lines, separated by carriage returns. The following two types of records represent utterances about previsions.

1. Assessed lower and/or upper bound(s) (or point value) on the prevision of a quantity. (Input.)

2. A computed extension interval for the prevision of a quantity. (Output.)

In each type of utterance about a prevision, the case of equal lower and upper bounds, a single point value, is handled by special notation. (A pair of equal bounds are optional on input.)

In order to keep track of what assessed bounds are assumed as the bases for computed intervals, and to promote the stepwise coherence-preserving use of the method, a step number is assigned to a new assessment the first time it is imposed in the calculation of an extension interval. That step number is also assigned to all extension intervals that are subsequently calculated before any further assessments are introduced.

It should be noted that a computed interval will only guarantee coherence of an assessment one new quantity at a time. If more than one quantity's new assessment is uttered in the same step, the linear programming routine could find that they are not coherent, even though each new assessment would be coherent if added singly. The program will issue a warning, yet it will not prevent the user from introducing multiple new assessments in a single step. The user may happen to know that coherence will be preserved, or may just wish to take a chance.

A third type of record provides the framework for prevision utterances:

3. A definition of a random quantity, stated with identifying name, description, range set, and relation(s) (if any) to preceding random quantities. An event is a quantity with the range set $\{0, 1\}$.

The records that define random quantities are spaced out in the file in the order they are introduced, and each is immediately followed by its corresponding prevision utterances, with step numbers. This format seems an important contribution, lending great convenience to the use of the program. The program actually allows the prevision utterances to be placed arbitrarily, but arranging them by quantity seems helpful. What the program requires for quantities is that they be defined and listed in a logical order that facilitates the computation of the joint range set.

4.4 Relations and the Joint Range

Hailperin [14, 15] seems not to have noticed that logical and other mathematical relations among random quantities can substantially reduce the size of their joint range set and, hence, diminish computing costs. It is not necessary, first, to define a full product space and then discard all the points made impossible by the relations. The program brings in only the possible points during the formation of the joint range set. Each definition of a quantity, imposing constraints relating it to previously defined quantities, enables the program to construct only those points that are possible as each quantity is introduced to the joint range. Any reference to a quantity that has not yet been introduced will raise an exception. Of course, the user can wait until the very last quantity defined in the file to impose all the relations, but this can be very inefficient, hence even nonfeasible.

Consider, for example, a partition, $A_1,\ldots,A_n$. The relation $A_1+\ldots+A_n=1$, meaning mutually exclusive and exhaustive (for $0-1$ quantities), can be more efficiently imposed piecemeal, as $A_1+\ldots+A_k\leq 1$ at each definition of A_k, $k=1,\ldots,n-1$, and then $=1$ at $k=n$. However, a more convenient approach, also efficient, is to define the A_k's as the value events of an artificial random quantity X with the arbitrary range $\{1,\ldots,n\}$. After first defining X with that range, let A_k : $X=k$, for $k=1,\ldots,n$. Then $A_1,\ldots,A_n$ will automatically comprise a partition.

4.5 Availability

The program, a moderately large Perl script wrapper on a publicly available open-source linear programming routine, lp-solve, currently runs under unix/linux. User control is through a program command line and the vi editor. The menu for the

MENU			
FILE:		ACTION:	
n	New	a	Assess
o	Open	au	Undo Assessment
s	Save	e	Extend
p	Print	eu	Undo Extension
q	Quit	t	Option

Figure 1: Program menu.

command line is shown in Fig. 1. To obtain the program via e-mail or ftp transfer, contact the author at dickey@stat.umn.edu. A tutorial file is also available.

5 Example: A Medical Screening Test

We demonstrate the use of the program with a simplified example of medical diagnosis. The assessed probability values here will help introduce the program, but they are not necessarily appropriate to the real problem, nor is the problem claimed to be a typical use of the program. Interaction with the program in the example will be described by showing the progressive states of the input/output file.

Suppose a person from the general population receives a positive test result, event S, in a screening skin test for tuberculosis. What is the conditional probability of the event T that she/he has tuberculosis, $P(T\,|\,S)$? This is a classic Bayes' Theorem problem, but the program does not see it as such, treating it more directly as a problem of implied bounds on conditional probability.

Assuming, first, the bounds on the prior probability, $5 \times 10^{-5} \leq P(T) \leq 10^{-4}$, and the test-performance probabilities, $P(S|T) = 1$, $1/20 \leq P(S|nT) \leq 1/10$, we will obtain the implied bounds on the marginal symptom probability, $.05005 \leq P(S) \leq .10001$, and the posterior-probability bounds, $.0004998 \leq P(T|S) \leq .001996$. This posterior probability, following a positive symptom, is small; but of course, it lies between about ten and twenty times the prior probability.

We will then use the computed extension interval of the marginal probability, $0.05005 \leq P(S) \leq 0.1001$, as a coherent guide for a further, precise assessment, $P(S) = 0.07$. It could be known, for example, that the relevant empirical frequency of positive test readings is equal to this value. The program then outputs the corresponding step-2 extension intervals. The interval for the conditional false-positive probability will shrink almost to a single point, $.06991 \leq P(S\,|\,nT) \leq .06995$, and the posterior probability of having the disease will be confined to the subinterval, $0.0007143 \leq P(T\,|\,S) \leq 0.001429$. Again, this is small; but it's about 15 times the prior probability.

```
* TITLE/DESCR:        -> SCREENING TEST FOR TB <-
* Separate fields by "; " (semicolon space(s)). Records (lines) by <Enter>.
* EVENT/QUANT DEFN FIELDS: xname; descr; rangeSet(or"fun"); relation(or expr)
* PROB/EXPEC FIELDS: (Indent)P(xname); bdL; bdU; "a"(assess) or "e"(extend)stepN

T;        Patient has TB;          (0, 1);       none
    P(T);     5.00e-05;   1.00e-04;   a

nT;       Doesn't have TB;         fun;          not $T

S;        Pos skin test;           (0, 1);       none
    P(S|T);     eq;        1;        a
    P(S|nT);    1/20;      1/10;     a
    P(S);       ;          ;         e
```

Figure 2: Initial input file.

6 Using the Program

The initial input file is given in Fig. 2. Four automatic header lines, each starting with a star “*”, consist of: title/description of the problem (as entered by user), a line giving formats, and two lines defining the fields of the quantity-definition lines and the fields of the prevision-utterance lines. (We drop these header lines in the subsequent figures.) The user-input lines of the three types follow.

The three left-justified lines here define the events, T, nT (for not$T$), and S. The ranges of T and S are given as the Perl list “$(0,1)$”, followed by “none” (for no relation). The event nT is defined as the function (“fun”), *not* T, of the 0-1 event quantity T. (The dollar sign in the expression “*not* \$$T$” signifies a variable in Perl.) Alternatively, nT could be defined as an event subject to a relation, with the fields, “$(0,1)$; $(\$T \ or \ \$nT) \ == \ 1$”.

The remaining indented lines are prevision utterances. The second and third fields are for lower and upper bounds, respectively, or for a point value when “*eq*” is entered in the second field followed in the third field by a single number. In the fourth field of a prevision utterance, the user indicates whether an assessment is being asserted (“*a*”) or an extension requested (“*e*”). Perpetual calculation, at each step, of the current extension interval for a quantity is the default action triggered by “*e*”. (To prevent the automatic later extensions, enter “*e*!”.) For a check on the effectiveness of assessed bounds, “*a*” also, by default, triggers the perpetual calculation of extension intervals. (Use “*a*!” to prevent it.) After an interval is calculated, the current step number is automatically appended to the fourth field.

Fig. 3 shows the file updated to report the calculation of three extension inter-

T;	Patient has TB;		(0, 1);	none
P(T);	5.00e-05;	1.00e-04;	a1	
;	[5e-05;	0.0001];	e1	
nT;	Doesn’t have TB;		fun;	not $T
S;	Pos skin test;		(0, 1);	none
S_T;	S and T;		fun;	$S and $T
S_nT;	S and nT;		fun;	$S and $nT
P(S|T);	eq;	1;	a1	
P(S|nT);	1/20;	1/10;	a1	
;	[0.05;	0.1];	e1	
P(S);	[0.05005;	0.1001];	e1	

Figure 3: First output

P(T|S);	[0.0004998;	0.001996];	e1

Figure 4: Second output (fragment)

vals: for $P(T)$; for $P(S\,|\,nT)$, both as checks; and for $P(S)$, the marginal probability of a positive skin test result. Note the new events, $(S\,and\,T)$ and $(S\,and\,nT)$, automatically defined by the program as needed to work with bounds on the conditional probabilities, $P(S\,|\,T) = P(S\,and\,T)\,/\,P(T)$ and $P(S\,|\,nT) = P(S\,and\,nT)\,/\,P(nT)$.

After including a new request for the conditional (“posterior”) probability $P(T\,|\,S)$, we obtain the output as given in Fig. 4, differing only in this one line from the output in Fig. 3. Note that, because no additional assessments were input, the assigned step number remains at 1.

Finally, we use the computed extension interval of the marginal probability $0.05005 \leq P(S) \leq 0.1001$, as a coherent guide for a further, precise assessment, $P(S) = 0.07$. Then the program outputs the corresponding step-2 extension intervals as given in Fig. 5.

7 Relevance of the Method

It seems to the author that these interactive methods could potentially be employed to advantage by real-time decision makers, such as physicians or military commanders. In personal discussion, Glen Meeden has suggested simultaneous cooperative use by a group of experts as an aid to achieving a jointly agreeable coherent assessment.

T;	Patient has TB;		(0, 1);	none
P(T);	5.00e-05;	1.00e-04;	a1	
;	[5e-05;	0.0001];	e1	
;	[5e-05;	0.0001];	e2	
P(T|S);	[0.0004998;	0.001996];	e1	
;	[0.0007143;	0.001429];	e2	
nT;	Doesn't have TB;		fun;	not $T
S;	Pos skin test;		(0, 1);	none
S_T;	S and T;		fun;	$S and $T
S_nT;	S and nT;		fun;	$S and $nT
P(S|T);	eq;	1;	a1	
P(S|nT);	1/20;	1/10;	a1	
;	[0.05;	0.1];	e1	
;	[0.06991;	0.06995];	e2	
P(S);	[0.05005;	0.1001];	e1	
;	eq;	0.07;	a2	
;	EQ;	[0.07];	e2	

Figure 5: Third output

Because of the convenience of this coherent assessment algorithm and its interactive implementation, and the flexibility afforded by imprecise assessments, the method would seem destined for heavy use. However, the need for and advantage of such a method hinges on the recognition of logical and other mathematical relations among the quantities whose previsions are subject to assessment. It seems still an open question whether such relations are rare or common in practice. Early Wittgenstein, in what has been called his Logical Independence Thesis, might be interpreted as claiming that such relations tend not to be basic in an analyses. Quoting from the Tractatus [23]:

> The world divides into facts [Prop. 1.2]
>
> Each can be the case or not the case, while the others remain the same [Prop. 1.21]
>
> (See also [Props. 2.061, 2.062].)

In the opinion of a referee, this is no longer an open question, "We simply have applications where logical independence holds and other cases where the random quantities are not logically independent."

8 Further Developments

1. Events or quantities having special properties are amenable to special coding:

   (a) Exchangeable events. Logical independence is usually assumed. Direct definition of variables representing the common invariant joint probabilities seems preferable to imposing the equality constraints for exchangeability on probability variables for a large number of events: $n+1$ variables with 1 constraint, versus 2^n variables with $2^n - n$ constraints.

   (b) Interval events on a random quantity. These can usefully accommodate envelope and other statements regarding the c.d.f.

2. Various upper and lower probability systems (C.A.B. Smith, Dempster-Shafer, etc.) can be incorporated as special program modes. Comparisons can be made in such applications as the use of multiple messages with specifiable reliabilities.

3. Reconciliation of incoherent previsions, by minimum distance under weighted least squares, or other, metric. See, for example, Nau [20, 21].

4. A graphical user interface (Perl/Tk) is being put onto the current functionality, for unix/linux and win32.

5. Charles Geyer has suggestied integration of the program into the emacs editor environment with separate simultaneous displays for the menu and input/output file.

6. Charles Geyer suggested that the program be recast as an ad hoc computing language, for possible inclusion in rweb, or other general system.

9 A Plea

The author would like to hear from conference participants and others interested in using or improving the program. Advice is welcome on what to do or how to do it better, and collaborative and coding help is especially welcome.

Acknowledgement

The author is grateful to anonymous referees for valuable suggestions that improved the paper.

References

[1] M.Baioletti, A.Capotorti, S.Tulipani, B.Vantaggi. Simplification rules for the coherent probability assessment problem, Annals of Mathematics and Artificial Intelligence, 35 (2002), 11-28.

[2] G. Boole. An Investigation of the Laws of Thought. Walton and Maberly, London, 1854.

[3] V. Biazzo, A. Gilio. "A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments". International Journal of Approximate Reasoning 24, 251-272, 2000.

[4] V. Biazzo, A. Gilio. "On the linear structure of betting criterion and the checking of coherence", Annals of Mathematics and Artificial Intelligence 35: 83-106, 2002.

[5] V. Biazzo, A. Gilio, G. Sanfilippo. "Coherence checking and propagation of lower probability bounds", Soft Computing 7 (2003), 310-320.

[6] G. Bruno and A. Gilio. Applicazione del metodo del simplesso al teorema fondamentale per le probabilita nella concezione soggettivistica. Statistica, Vol. 40, 1980, No. 3, pp 337-344.

[7] A.Capotorti, B.Vantaggi. A simplified algorithm for inference by lower conditional probabilities, Proc. ?2nd ISIPTA? (2001) Ithaca (New York), 68-76;

[8] A.Capotorti, B.Vantaggi. Locally strong coherence in inferential processes, Annals of Mathematics and Artificial Intelligence, 35 (2002), 125-149.

[9] G.Coletti. Coherent numerical and ordinal probabilistic assessments, IEEE Trans. on Systems, Man, and Cybernetics, 24(12) (1994), 1747-1754.

[10] G.Coletti, R.Scozzafava. The Role of Coherence in Eliciting and Handling Imprecise Probabilities and its Application to Medical Diagnosis. Information Science, 130 (2000), 41-65.

[11] G.Coletti, R.Scozzafava. Probabilistic logic in a coherent setting. Trends in logic n.15, Kluwer, Dordrecht/Boston/London 2002.

[12] B. deFinetti. La prévision, ses lois logiques, ses sources subjectives. Annales de L'Institute Henri Poincaré, Vol. 7, 1937, pp 1-68. H. Kyburg (tr.) Foresight, its logical laws, its subjective sources, in Kyburg, H., and Smokler, H. (eds.) Studies in Subjective Probability, Wiley, New York, 1964; 2nd ed., Krieger, NewYork, 1980.

[13] B. deFinetti. Theory of Probability, Vol. 1. English translation of Teoria delle Probabilità (1970). Wiley, London, 1974.

[14] T. Hailperin. Best possible inequalities for the probability of a logical function of events. American Mathematical Monthly, Vol. 72, 1965, pp. 343-359.

[15] T. Hailperin. Sentential Probability Logic. Lehigh University Press, Bethlehem, PA, 1996.

[16] B. Jaumard; P. Hansen; M. Poggi de Aragao. Column generation methods for probabilistic logic, ORSA J. Comput. 3 (1991) 135-148.

[17] F. Lad. Operational Subjective Statistical Methods. Wiley, New York, 1996.

[18] F. Lad; J. Dickey; M. Rahman. The fundamental theorem of prevision. Statistica, Vol. 50, 1990, No. 1, pp 15-38.

[19] F. Lad; J. Dickey; M. Rahman. Numerical application of the fundamental theorem of prevision. J. of Statistical Computation and Simulation, Vol 40, 1992, pp 135-151.

[20] R. Nau. Decision Analysis with Indeterminate or Incoherent Probabilities. Annals of Operations Research, 1989.

[21] R. Nau. Indeterminate Probabilities on Finite Sets. Annals of Statistics, 1992.

[22] P. Walley. Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London, 1991.

[23] L. Wittgenstein. Tractatus Logico-Philosophicus. English translation by D. F. Pears and B. F. McGuiness from the German, in Annalen der Naturpholosophie (1921). Routledge & Kegan Paul Ltd, London, 1961.

James Dickey is with the School of Statistics of the University of Minnesota.