Averaging and Eliciting Expert Opinion*

Peter M. Williams

p n o Co p n
n s o ss
B on ss B

A s
s

Abstract

The paper considers the problem of averaging expert opinion when opinions are expressed quantitatively by belief functions in the sense of Glenn Shafer. Practical experience shows that experts usually differ in their exact quantitative assessments and some method of averaging is necessary. A natural requirement of consistency demands that the operations of averaging and combination, in the sense of Dempster's rule, should commute. Experience also shows that symmetric belief functions are difficult to distinguish in practice. By forming a quotient of the monoid of belief functions modulo the ideal of symmetric belief functions, we are left with an Abelian group with a natural scalar multiplication making it a real vector space. The elements of this quotient space correspond to what we call "regular" belief functions. This solves the averaging problem with arbitrary weights. The existence of additive inverses for regular belief functions means that contrary evidence can be treated without assuming the existence of complements. Opinions expressed by conditional judgements can be incorporated by lifting suitable measures from a quotient space to a numerator. The appendix describes a computer program for implementing these ideas in practice.

---

*Preparation of this paper was supported by SERC grant GR/E 05360. The ADRIAN project was sponsored by ICI Pharmaceuticals. Thanks are due to both organisations.

Contents

1.3 Contrary Evidence

o p C

2.1 Distributive Lattices

A **partially ordered** s s s **A** n on $\leq$ s s n

$\mathbf{a} \leq \mathbf{a}$

$\mathbf{a} \leq \mathbf{b}$ n $\mathbf{b} \leq \mathbf{a}$ p $\mathbf{a}$ $\mathbf{b}$

$\mathbf{a} \leq \mathbf{b}$ n $\mathbf{b} \leq \mathbf{c}$ p $\mathbf{a} \leq \mathbf{c}$

o $\mathbf{a, b, c}$ n **A** A s s **S** o p o s **A** s s o n **upper set** A

2.2 Probability Measures on Distributive Lattices

Proposition 1 **Every probability measure on a distributive lattice D has a unique extension to a probability measure on the Boolean algebra freely generated by D.**

$$p(a \vee b \vee c)$$
$$p(a) + p(b) + p(c) - p(a \wedge b) - p(a \wedge c) - p(b \wedge c) + p(a \wedge b \wedge c).$$

n n **S** n n o **D** n **p** p o on **D** n

$$p(\bigvee S) = \sum_{R} (-1)^{|R|} p(\bigwedge R)$$

$|R|$ n o n n **R** n on o p n n p o o o n o o n n p o o o n o n

2.3 Semilattices

o o n n n o n on n o on o n C o o o o o o po n o A **meet semilattice** p o n n A **join semilattice** p o n n o n A n n op n o n n o o n A o p o p p n o p o o n p p n o n p n p o o o p o

n o ono o o p
n n po n o
on n o n on

$$\mathbf{a} \leq \mathbf{b} \quad \mathbf{a} \vee \mathbf{b} \quad \mathbf{b}.$$

A on ∧ p ∨ n p
n o on o n oppo n

$$\mathbf{a} \leq \mathbf{b} \quad \mathbf{a} \wedge \mathbf{b} \quad \mathbf{a}.$$

o po n o no n n
n on o no o pon n o
n on n n n p o p n
on on n oo on o o n o o n
A o **complete**
n o p on o p
on o o o p on n o p
p n o o n o n n
o **suplattices** n **inflattices** p
n p n p o o o p o oppo
o o

A p o p on n **A** n
on o n **S**
on o o on

$$\bigwedge \mathbf{S} \quad \bigvee\{\mathbf{a} \in \mathbf{A} | \mathbf{S} \subseteq \uparrow \mathbf{a}\}.$$

n n on[3] n n **A**
p p o on
oppo p o $\mathbf{A}^o$ o p on o $\mathbf{A}^o$
o **A** oppo o n n n n o
A n **B** p n $\mathbf{f}: \mathbf{A} \to \mathbf{B}$ o p o p
o **f** p on n n o p o
p o $\mathbf{f}: \mathbf{B} \to \mathbf{A}$ n on on

$$\mathbf{f}(\mathbf{a}) \leq \mathbf{b} \quad \mathbf{a} \leq \mathbf{f}(\mathbf{b})$$

o $\mathbf{a} \in \mathbf{A}$ n $\mathbf{b} \in \mathbf{B}$ p **f** n

$$\mathbf{f}(\mathbf{b}) \quad \bigvee\{\mathbf{a} \in \mathbf{A} | \mathbf{f}(\mathbf{a}) \leq \mathbf{b}\}.$$

f p n **A** on n **f** p
n n $\mathbf{f}: \mathbf{A} \to \mathbf{B}$ n $\mathbf{g}: \mathbf{B} \to \mathbf{A}$ o p n p n
p o n

$$\mathbf{f}(\mathbf{a}) \leq \mathbf{b} \quad \mathbf{a} \leq \mathbf{g}(\mathbf{b})$$

[3] Thus, as an object, a complete semilattice or either sort is in fact a complete lattice. However, since a morphism of suplattices need not preserve meets, nor a morphism of

4848-1274252((()-0i)0.742562837(n)1.91,22283(2056]TJ 83(20131(,)-444.923(s)-3.0.26064(r)-0.649399(y)-396.625(j)-1.94207(o)-2.2626

$a \in A$ $b \in B$ f **left adjoint** g g **right adjoint** f

$$f(a) = \bigwedge\{b \in B \mid a \leq g(b)\}$$

$$g(b) = \bigvee\{a \in A \mid f(a) \leq b\}.$$

A A^o f $f^o : B^o \to A^o$ A B B^o A^o

2.4 Probability Measures on Inflattices

Definition 1 A … on a finite inflattice A is a real unit-interval valued function $p : A \to$, satisfying

$p(b)$ … p

a o **A** o ↓ **a** **DA**

for all $a \in A$. Moreover this function, called the n **of p, is unique when it exists.**

p o **p** n **p^o** n **m** n o **p** n **p^o** n

$$p^o(a) = \sum\{m(b) \mid a \leq b\}.$$

A n **B** n no **$A \oplus B$** o p **(a, b)** **$a \in A$** n **$b \in B$** n **$A \oplus B$** n n n oo n p o n p o o **A** n **B** n o o n n o o n on n p o on

Proposition 6 **If p and q are probability measures on the finite inflattices A and B respectively, then the function $p \times q$ defined for all $a \in A$ and $b \in B$ by**

$$(p \times q)(a, b) = p(a)\, q(b)$$

is a probability measure on $A \oplus B$.

Proof n **m** o **$p \times q$** po n p o o n o **p** n **q** **$m(a, b) = m_p(a)\, m_q(b)$** **$m_p$** n o **$p$** n **$m_q$** n o **$q$** □

Corollary 7 **If p and q are probability measures on an inflattice A then the function $p \cdot q$ defined for all $a \in A$ by**

$$(p \cdot q)(a) = p(a)\, q(a)$$

is also a probability measure on A.

Proof **$A \to A \oplus A$** on o p n n **a** o **(a, a)** n **$p \cdot q$** **$(p \times q)$** o p o on **A** □

Co o o n o n op on on p o on n n o p o o n on n o o n on **$\mathrm{Pr}(A)$** no o p o on n n **A** n n op on n

$$p \,\square\, q = (p^o \cdot q^o)^o$$

o **$p, q \in \mathrm{Pr}(A)$**

Proposition 8 **$\mathrm{Pr}(A)$ is a commutative monoid under □.**

Proof A … $\square$

… $\mathbf{v}$ … $\mathbf{Pr}(\mathbf{A})$ … **vacuous** … on $\mathbf{A}$ …

$$\mathbf{v}(\mathbf{a}) = \begin{cases} 1 & \mathbf{a} = 1 \\ 0 & \text{otherwise} \end{cases}$$

… $\mathbf{p}$ … $\mathbf{m}_p$ … $\mathbf{q}$ … $\mathbf{m}_q$ … $\mathbf{p} \oplus \mathbf{q}$ …

$$\mathbf{m}(\mathbf{a}) = \sum\{\mathbf{m}_p(\mathbf{b})\,\mathbf{m}_q(\mathbf{c}) \mid \mathbf{a} = \mathbf{b} \wedge \mathbf{c}\}.$$

… $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ … $\mathbf{f}^o$ … $\mathbf{B}^o$ … $\mathbf{A}^o$ … $\mathbf{Pr}(\mathbf{f}) : \mathbf{Pr}(\mathbf{A}) \to \mathbf{Pr}(\mathbf{B})$

$$\mathbf{Pr}(\mathbf{f})(\mathbf{p}) = (\mathbf{p}^o \circ \mathbf{f}^o)^o$$

… $\mathbf{p} \in \mathbf{Pr}(\mathbf{A})$

Proposition 9 **Pr is a (covariant) functor from the category of finite inflattices to the category of commutative monoids.**

Proof … $\mathbf{p} \in \mathbf{Pr}(\mathbf{A})$ … $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ … $\mathbf{f}^o : \mathbf{B}^o \to \mathbf{A}^o$ … $\mathbf{p}^o \circ \mathbf{f}^o \in \mathbf{Pr}(\mathbf{B}^o)$ … $\mathbf{p}^o \in \mathbf{Pr}(\mathbf{A}^o)$ … $(\mathbf{p}^o \circ \mathbf{f}^o)^o = \mathbf{Pr}(\mathbf{f})(\mathbf{p}) \in \mathbf{Pr}(\mathbf{B})$ … $\mathbf{p}, \mathbf{q} \in \mathbf{Pr}(\mathbf{A})$ …

$$\begin{aligned}
\mathbf{Pr}(\mathbf{f})(\mathbf{p} \oplus \mathbf{q}) &= ((\mathbf{p} \oplus \mathbf{q})^o \circ \mathbf{f}^o)^o \\
&= ((\mathbf{p}^o \cdot \mathbf{q}^o) \circ \mathbf{f}^o)^o \\
&= ((\mathbf{p}^o \circ \mathbf{f}^o) \cdot (\mathbf{q}^o \circ \mathbf{f}^o))^o \\
&= (\mathbf{p}^o \circ \mathbf{f}^o)^o \oplus (\mathbf{q}^o \circ \mathbf{f}^o)^o \\
&= \mathbf{Pr}(\mathbf{f})(\mathbf{p}) \oplus \mathbf{Pr}(\mathbf{f})(\mathbf{q})
\end{aligned}$$

… $\mathbf{Pr}(\mathbf{f})$ … $\mathbf{Pr}(\mathbf{A})$ … $\mathbf{Pr}(\mathbf{f})$ … $\mathbf{Pr}(\mathbf{g} \circ \mathbf{f}) = \mathbf{Pr}(\mathbf{g}) \circ \mathbf{Pr}(\mathbf{f})$ … $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ … $\mathbf{g} : \mathbf{B} \to \mathbf{C}$ … $(\mathbf{g} \circ \mathbf{f})^o = \mathbf{f}^o \circ \mathbf{g}^o$ … $\mathbf{Pr}$ … $\square$

… $\mathbf{Pr}$ … $\mathbf{p}$ … on $\mathbf{A}$ … $\mathbf{m}$ … $\mathbf{f} : \mathbf{A} \to \mathbf{B}$ … $\mathbf{m}_f$ … $\mathbf{Pr}(\mathbf{f})(\mathbf{p})$ … $\mathbf{b} \in \mathbf{B}$

$$\mathbf{m}_f(\mathbf{b}) = \sum\{\mathbf{m}(\mathbf{a}) \mid \mathbf{f}(\mathbf{a}) = \mathbf{b}\}.$$

on n s s o o n n n o n s n

s no po n s o n n s o op o o o n s

, , . . .

o n n o $p \in Pr(A)$ $q \in Pr(A)$ $p \neq q$ o n o o $p \wedge q$ n o n $(p \wedge q)^o$ $p^o \cdot q^o$ n o q^o o n p opo on o p o o o p^o n o o on p o n p n o o p^o o n o o n on o p n o

Lemma 11 **Let f be any real-valued function on a finite inflattice A with $n \neq$ ranks. Then there exists a proper probability measure p on A and a sequence of positive real number $K_0, \dots, K_n$ such that for each i $, \dots, n$**

$$p^o(a) \quad K_i \quad \text{p}\, f(a)$$

whenever n (a) **i.**

Proof A $f(y)$ n n o n on $(m_i)_{i \le n}$ n on m_i n on $\{a \in A \mid \text{n}(a) \le i\}$ o

$$m_0(y) \quad \text{p}\, f(y)$$

$$m_i(a) \quad \begin{cases} k_i m_{i-1}(a) & \text{n}\,(a) < i \\ \text{p}\, f(a) - k_i g_i(a) & \text{n}\,(a) \quad i \end{cases}$$

$$g_i(a) \quad \sum\{m_{i-1}(b) \mid a < b\}$$

$$k_i \quad \text{n}\, \frac{\text{p}\, f(a)}{g_i(a)}$$

n n o $\{a \in A \mid \text{n}(a) \quad i\}$ o n on on n k_i n n po n n on m_i non n n p m_n non n m m_n n o n on

$$(y) \quad \sum\{m_i(b) \mid a \le b\} \quad \text{p}\, f(a)$$

n n (a) i n $a < b$ n n (a) i p n $(b) < i$ $\sum_{a \in A} m(a)$ $\sum$ i $f($

o s p s on n s m s p op p o s o o o o s o n n n a i

$$p^o(a) \quad \sum\{m(b) \mid a \le b\} \quad \sum\{K_i m_i(b) \mid a \le b\} \quad K_i \quad p\, f(a)$$

s o n ss on f( ) / pp o on s on o n on $f'(a)$ $f(a) - f( )$ n o p $- f( )$ n o o o s K_i □

Definition 3 **If f is any real-valued function on a finite inflattice A we denote by f the proper probability measure defined by the above construction.**

Proposition 12 **Pr(A) /Un(A) is an Abelian group.**

Proof ppo s **p** on s o **Pr(A)** **q** $-$ **p**^{o} n **q**^{o} p opo on o p $-$ **p**^{o} /**p**^{o} o ss n s n **p** **q**

Proof $p \equiv q$ … u … v … $p \cdot u$ … $q \cdot v$ … $p^o -$ … q^o … $v^o -$ … u^o … $N(A)$ … □

Proposition 14

$Pr(A)/Un(A)$ is isomorphic to the additive group of $L(A)/N(A)$.

Proof … $\varphi : Pr(A)/Un(A) \to L(A)/N(A)$

$$\varphi(p) \ \ldots \ p^o$$

… $\psi : L(A)/N(A) \to Pr(A)/Un(A)$

$$\psi(f) \ \ldots \ f.$$

… $\varphi(p)$ … p …

$$\begin{aligned}\varphi(p \ldots q) &= \varphi(p \cdot q)\\ &= \ldots (p \cdot q)^o\\ &= \ldots p^o \ldots q^o\\ &= \ldots p^o \ldots q^o\\ &= \varphi(p) \ldots \varphi(q).\end{aligned}$$

… φ … f … $L(A)$ …

$$\varphi(\psi(f)) \quad \varphi(\ldots f) \quad \ldots f^o \quad f$$

… $\varphi \circ \psi$ … $L(A)/N(A)$ … φ … $\psi \circ \varphi$ … $Pr(A)/Un(A)$ … φ … ψ … □

3.2 Regular Measures

…

Definition 4 **Let $\rho : Pr(A) \to Pr(A)$ be defined by**

$$\rho(p) \ \ldots \ p^o).$$

We say that a proper measure p is … if and only if $\rho(p)$ … p and we denote by $Reg(A)$ the set of regular measures on a finite inflattice A.

no o n o $\mathbf{Pr}(A)/\mathbf{Un}(A)$ on n on

Lemma 15 **ρ is idempotent: $\rho \circ \rho$ ρ. Hence $\rho(p)$ is regular for all $p \in$ $\mathbf{Pr}(A)$.**

Proof f p^o n o n pp □

Proposition 16 **Each element of $\mathbf{Pr}(A)/\mathbf{Un}(A)$ contains one and only one regular measure.**

Proof ppo p on o $\mathbf{Pr}(A)/\mathbf{Un}(A)$ n $\varphi(\psi(p))$ p p p^o p o n o o $\rho(p)$ p B $\rho(p)$ n n o $\mathbf{Pr}(A)/\mathbf{Un}(A)$ on n on o ppo $p \equiv q$ n $\rho(p)$ $\rho(q)$ n o o p n q o p $\rho(p)$ $\rho(q)$ q n n $\mathbf{Pr}(A)/\mathbf{Un}(A)$ on n o on □

n $\mathbf{Reg}(A)$

o $a \in A$ n n op on on $\mathrm{Blf}(A)$ p
o o n on n

$$p \quad q \quad \beta(p \quad q)$$

n $\beta: \mathrm{Pr}(A) \to \mathrm{Blf}(A)$ o p o o ono
o o o p o on o n
$\rho: \mathrm{Pr}(A) \to \mathrm{Reg}(A)$ n

$$\rho(p) \quad o \quad p^{o} .$$

n op on on $\mathrm{Reg}(A)$ n

$$p \quad q \quad \rho(p \quad q)$$

n $\rho: \mathrm{Pr}(A) \to \mathrm{Reg}(A)$

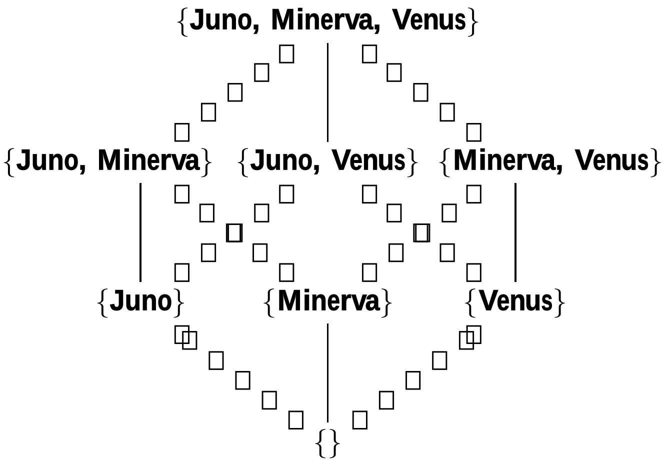
n on on o o n o n o n **B**
n no o n n o **A**
n o
o n **b** o n o n n **A** n
B $\downarrow$ **b** n o n on o pon n o p
o o n p $\wedge$ **b** **A** $\rightarrow$ **B** n n n **a** $\in$ **A** o
a $\wedge$ **b** $\in$ **B** po n o o $\downarrow$ **b** n no n n
o n o o **A** n o **rILf** n n on o $\downarrow$ **b** n **A**
n no p o n B po o oppo
o no n n o n
n **A** n n **A** **tree-like** $\uparrow$ **b**
n o o **b** $\in$ **A** p po **b** n n on o
$\downarrow$ **b** n **A** p o n
p n o o on on on n Boo n
p o o on p n $\downarrow$ **b** n
o n o o **A** App n **Reg** o o p $\wedge$ **b** **A** $\rightarrow$ **B**
n n o on o **Reg** **A** o **Reg** **B** o pon n o po
on on on o n o o o
on **A**

Reg … $\mathbf{M}(A)$ … $\mathbf{Reg}(A)$ … $\mathbf{M}(A)$ … $\mathbf{L}(A)$ … A … $f: A \to B$ … $\mathbf{L}(f): \mathbf{L}(A) \to \mathbf{L}(B)$

$\mathbf{L}(f)$ … let

o A n n o α

Pr f p ono op on n o
n o po n o po on **f** o no o
n on on o n p $\wedge$ **b** n **b** / Co po n
n o **Pr** $\downarrow$ **b** no o n o **Pr A** o
on n on **b** n o n n p op on C
n o o p po n on n no n n on o
ppo o p o no n o
o no on on o B o
n o A B on p
p n on o n o on
o n o n o



The free suplattice generated by the set { no n n }

n on n X on o n n po
o X n o n n on po
o oppo n on n po o o p o
oppo o p n

⊤

{**subject drug**} {**something else**}

⊥

A simple alternative.

necessary … **su□cient**

Proposition 21 Every probability measure on a finite suplattice A has a

n op on o n

v a { a
o

n on o pon n o o o
n o on o n n o n on n n p opo on n
p p o o on n n p opo
on n n n p o o opo on n
p o nno n n p o
o n p o n o Pr no n o o
no on o n on n
o n n A n p Pr A no o

Example 1 A n p n $\bigvee$ $PA \to A$ A o n o p n p o p n **rSLf** o n o n n pp n

Example 2 S n o o n p A n B $S \cup \{\ \}$ n on o S op n o A n $f: A \to B$ n

$$f(a) \quad \begin{cases} a & a \in S \\ & \text{o} \end{cases}$$

o n p n **rSLf** o n n on

Example 3 ppo A PX n po p o n on n Y o o X B $\{S \subseteq X \mid Y \subseteq S\}$ o on o p o Y B o n n on n o A n o o pon o o n p o n p $\cup Y: A \to B$ n $S \subseteq X$ o n on Y o n n n on o n p o n n o p o S n n o

6.3 Contravariant Transformations

8 Elicitation

('Juno or Minerva or Venus', 1)
('Juno or Minerva', 1)
('Juno or Venus', 1)
('Minerva or Venus', 1)
('Juno', 1)
('Minerva', 1)
('Venus', 0.6)
('', 0)

Evidence against

('Juno or Minerva or Venus', 1)
('Juno or Minerva', 0.84)
('Juno or Venus', 1)
('Minerva or Venus', 1)
('Juno', 0.6)
('Minerva', 0.6)
('Venus', 1)
('', 0)

s no oo s n n n o n o o p opo s s n s n

s p s on o n s p o o n n

('Diana or Juno or Minerva or Venus', 1)
('Diana or Juno or Minerva', 1)
('Diana or Juno or Venus', 0.8741)
('Diana or Minerva or Venus', 0.8741)
('Juno or Minerva or Venus', 0.8659)
('Diana or Juno', 0.7796)
('Diana or Minerva', 0.7796)
('Diana or Venus', 0.6537)
('Juno or Minerva', 0.8659)
('Juno or Venus', 0.6455)
('Minerva or Venus', 0.6455)
('Diana', 0.4647)
('Juno', 0.5510)
('Minerva', 0.5510)
('Venus', 0.3306)
('', 0)

p op p o **A** o **B** n op p o
on n o n o o p

Hom A

Declarations

`val`

```
val x =  ;
```

`int` `fun`

```
fun successor x = x + 1;
```

`successor` `int -> int`

```
fun mult(x,y) = x * y: int;
```

`(int * int) -> int` [14]

```
fun add x y = x + y: int;
```

`int -> (int -> int)` `add` `add`

```
val successor = add 1;
```

```
val successor = fn x => x + 1;
```

```
val add = fn x => fn y => x + y: int;
```

Lists

A ... [, ,] ... int list ... [] ... nil ...

```
[ , , ] =  :: [ , ]
        =  :: (  :: [ ])
        =  :: (  :: (  :: nil))).
```

... nil ... a::l ...

```
fun sum nil = 0
  | sum (a::l) = a + sum l;
```

... int list -> int ... iter ...

```
fun iter f u nil = u
  | iter f u (a::l) = f a (iter f u l);
```

... f ... add ... u ... 0 ...

```
val sum = iter add 0;
```

... iter ... foldr ... reduce ...

... map ... filter ... l ... $a_1, \ldots, a_n$ ... 'a ... f ... f ... 'a -> 'b ... map f l ... $f\ a_1, \ldots, f\ a_n$ ... 'b ... 'a ... 'b ... map ... ('a -> 'b) -> ('a list -> 'b list) ... 'a ... filter p l ... filter ... ('a -> bool) -> ('a list -> 'a list)

... g o f ... ('b -> 'c) * ('a -> 'b) -> 'a -> 'c ...

The Code

```
(*******************************************************
 *       Title:      Moebius                           *
 *       LastEdit:   1 June 1                          *
 *       Author:     Peter M Williams                  *
 *                   University of Sussex              *
 *******************************************************)


datatype SENSE = Inf | Sup;

type LATTICE  = bool list list list;

type DATUM =
     (bool list * (bool list list * bool list list)) * real;


exception hd;
fun hd nil = raise hd
  | hd (a::l) = a;

fun cons a l = a::l;

fun iter f u nil = u
  | iter f u (a::l) = f a (iter f u l);

fun append l m = iter cons m l;

val flat = iter append nil;

fun map f = iter (cons o f) nil;

fun filter p =
    iter (fn a => fn l => if p a then a::l else l) nil;

val sum'r = iter (fn x => fn y => x + y) 0.0;

val inf'r =
    iter (fn x => fn y => if x < y then x else y) (1.0/0.0);
```

The Code

```
infix C;
```

```
  | mean l = sum'r l/length'r l;

fun center nil = nil
  | center l =
    let val m = mean(map (fn(a,x) => x) l)
    in map (fn(a,x) => (a,x - m)) l end;

fun lookup (a: bool list) nil = 0.0
  | lookup a ((b,x)::l) = if a = b then x else lookup a l;

fun combine f (a::l) (b::m) = f a b ::  combine f l m
  | combine f _ _ = nil;

val zero = (map o map) (fn a => (a,0.0));

val add =
    (combine o combine) (fn(a,x) => fn(_,y) => (a,x+y: real));

fun mult k = (map o map) (fn(a,x) => (a,k*x: real));

fun profile sense lattice =
let fun insert (datum as ((b,(pos,neg)),s)) =
    let val x = sgn(s) *  (ln(1.0 - abs s))
        val w = if sense = Sup then  x else x
        val (S,T) =
        if sense = Sup then (neg,pos) else (pos,neg)
        val unit  = (hd o hd o rev) lattice
        val c = union unit S
        val l = map (filter (fn a => (c C a))) lattice
        val m =
        iter (fn t => map (filter (fn a => not(t C a)))) l T
        val n =
        (map o map)(fn a => if b C a then (a,w) else (a,0.0)) m
        val q = (flat o map center) n
        fun f(a) = let val ac = a U c in (a,lookup ac q) end
    in (map o map) f lattice end
in
iter (add o insert) (zero lattice)
end;
```

The Code

```
abstype MEASURE = Measure of SENSE *
      ((bool list * real) list list * (bool list * real) list)
with
local

fun construct sense (lattice: LATTICE) (data: DATUM list) =
let val profile = profile sense lattice data
    val measure = regularise sense profile
in Measure(sense,(profile,measure)) end

in

val infcon = construct Inf
val supcon = construct Sup

exception sense
infix ++
fun (Measure(s1,(q1,p1))) ++ (Measure(s ,(q ,p ))) =
if s1 <> s  then raise sense else
let val s = s1
    val q = add q1 q
in Measure(s,(q, regularise s q)) end

infix **
fun (Measure(s,(q,p))) ** k =
let val kq = mult k q
in Measure(s,(kq, regularise s kq)) end

fun find(Measure(s,(q,p))) = p

end
end;

(**********************************************************
The exported functions have types:
```

REFERENCES

A o on o p o **Annals of Probability 7**

B n on n po n C B **The Analysis of Fuzzy Information 1** C C

o on B n o n o p **Journal of Combinatorial Theory 2**