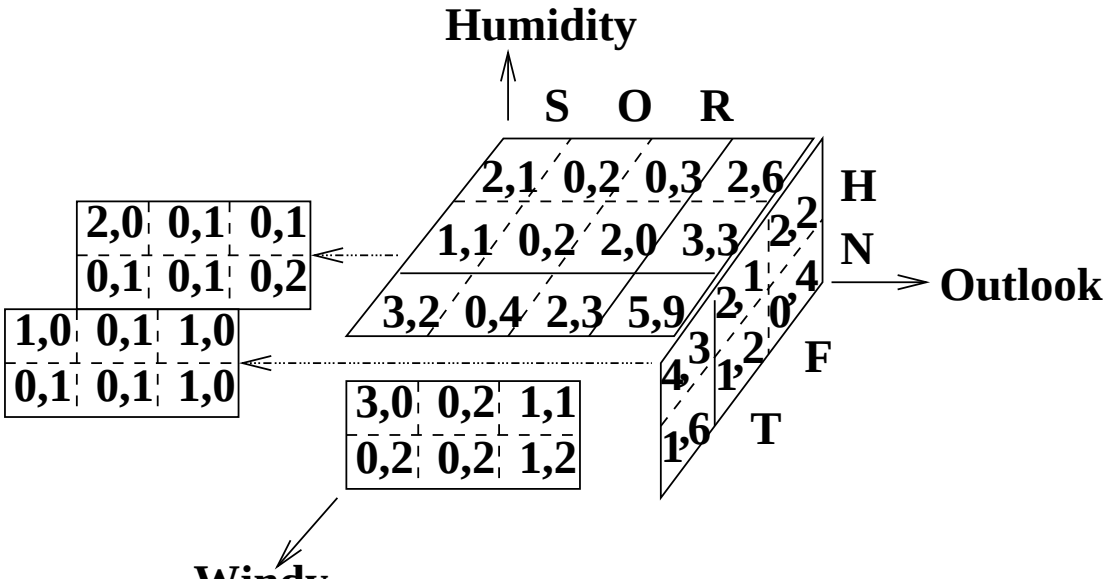
Copyright ©2001 Timothy Howard Merrett
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation in a prominent place. Copyright for components of this work owned by others than T. H. Merrett must be honoured. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to republish from: T. H. Merrett, School of Computer Science, McGill University, fax 514 398 3883.

The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.

Attribute Metadata for Relational OLAP and Data Mining

T. H. Merrett, McGill University

I Building a DataCube with the Domain Algebra



Trning	(*Ok*	*Tp*	*Hm*	*Wn*	*Cl*)
1	sn	ht	hi	f	N
2	sn	ht	hi	t	N
3	ov	ht	hi	f	P
4	rn	ml	hi	f	P
5	rn	cl	nm	f	P
6	rn	cl	nm	t	N
7	ov	cl	nm	t	P
8	sn	ml	hi	f	N
9	sn	cl	nm	f	P
10	rn	ml	nm	f	P
11	sn	ml	nm	t	P
12	ov	ml	hi	t	P
13	ov	ht	nm	f	P
14	rn	ml	hi	t	N

1. Count, Remove *Temperature* (*Tp*)

```
let N be equiv + of
  if Cl="N" then 1 else 0;                    N01
  by Ok, Hm, Wn;
let P be equiv + of
  if Cl="P" then 1 else 0;                    P01
  by Ok, Hm, Wn;
Trning <- [Ok, Hm, Wn, N, P] in Trning;
```

Trning	(*Ok*	*Hm*	*Wn*	N	P)
1,8	sn	hi	f	2	0
2	sn	hi	t	1	0
9	sn	nm	f	0	1
11	sn	nm	t	0	1
3	ov	hi	f	0	1
12	ov	hi	t	0	1
13	ov	nm	f	0	1
7	ov	nm	t	0	1
4	rn	hi	f	0	1
14	rn	hi	t	1	0
5,10	rn	nm	f	0	2
6	rn	nm	t	1	0

Intermediate Details:

let *N01* **be if** *Cl*="N" **then** 1 **else** 0;
let N **be equiv** + **of** *N01* **by** *Ok*, *Hm*, *Wn*;
let *P01* **be if** *Cl*="P" **then** 1 **else** 0;
let P **be equiv** + **of** *P01* **by** *Ok*, *Hm*, *Wn*;

Trning	(*Ok*	*Hm*	*Wn*	*Tp*	*Cl*)	*N01*	N	*P01*	P
1	sn	hi	f	ht	N	1	2	0	0
8	sn	hi	f	ml	N	1	2	0	0
2	sn	hi	t	ht	N	1	1	0	0
9	sn	nm	f	cl	P	0	0	1	1
11	sn	nm	t	ml	P	0	0	1	1
3	ov	hi	f	ht	P	0	0	1	1
12	ov	hi	t	ml	P	0	0	1	1
13	ov	nm	f	ht	P	0	0	1	1
7	ov	nm	t	cl	P	0	0	1	1
4	rn	hi	f	ml	P	0	0	1	1
14	rn	hi	t	ml	N	1	1	0	0
5	rn	nm	f	cl	P	0	0	1	2
10	rn	nm	f	ml	P	0	0	1	2
6	rn	nm	t	cl	N	1	1	0	0

2a. Sum Counts in *Windy* (*Wn*) Direction

let N **be** *totN*;
let P **be** *totP*;
let *Wn* **be** `"ANY"`;
let *totN* **be equiv** + **of** N **by** *Ok, Hm*;
let *totP* **be equiv** + **of** P **by** *Ok, Hm*;
update *Trning* **add**
[*Ok, Hm, Wn, N, P*] **in**
[*Ok, Hm, totN, totP*] **in** *Trning*;

The following are added to *Trning*

Trning	(*Ok*	*Hm*	*Wn*	N	P)
1,2,8	`sn`	`hi`	`ANY`	3	0
9,11	`sn`	`nm`	`ANY`	0	2
3,12	`ov`	`hi`	`ANY`	0	2
7,13	`ov`	`nm`	`ANY`	0	2
4,14	`rn`	`hi`	`ANY`	1	1
5,6,10	`rn`	`nm`	`ANY`	1	2

Intermediate Details:

let *totN* **be equiv** + **of** N **by** *Ok*, *Hm*;

let *totP* **be equiv** + **of** P **by** *Ok*, *Hm*;

Trning	(*Ok*	*Hm*	*Wn*	N	P)	*totN*	*totP*
1,8	sn	hi	f	2	0	3	0
2	sn	hi	t	1	0	3	0
9	sn	nm	f	0	1	0	2
11	sn	nm	t	0	1	0	2
3	ov	hi	f	0	1	0	2
12	ov	hi	t	0	1	0	2
13	ov	nm	f	0	1	0	2
7	ov	nm	t	0	1	0	2
4	rn	hi	f	0	1	1	1
14	rn	hi	t	1	0	1	1
5,10	rn	nm	f	0	2	1	2
6	rn	nm	t	1	0	1	2

let N **be** *totN*;

let P **be** *totP*;

let *Wn* **be** "ANY";

2b. Sum Counts in *Outlook* (*Ok*) Direction

let N **be** *totN*;
let P **be** *totP*;
let *Ok* **be** `"ANY"`; ←
let *totN* **be equiv** + **of** N **by** *Wn*, *Hm*; ←
let *totP* **be equiv** + **of** P **by** *Wn*, *Hm*; ←
update *Trning* **add**
[*Ok*, *Hm*, *Wn*, *N*, *P*] **in**
[*Wn*, *Hm*, *totN*, *totP*] **in** *Trning*; ←

The following are added to *Trning*

Trning	(*Ok*	*Hm*	*Wn*	N	P)
1,3,4,8	ANY	hi	f	2	2
2,12,14	ANY	hi	t	2	1
5,9.10,13	ANY	nm	f	0	4
6,7,11	ANY	nm	t	1	2
	ANY	hi	ANY	4	3
	ANY	nm	ANY	1	6

2c. Sum Counts in *Humidity* (*Hm*) Direction

let N **be** *totN*;
let P **be** *totP*;
let *Hm* **be** "ANY"; ←
let *totN* **be equiv** + **of** N **by** *Ok, Wn*; ←
let *totP* **be equiv** + **of** P **by** *Ok, Wn*; ←
update *Trning* **add**
[*Ok, Hm, Wn, N, P*] **in**
[*Ok, Wn, totN, totP*] **in** *Trning*; ←

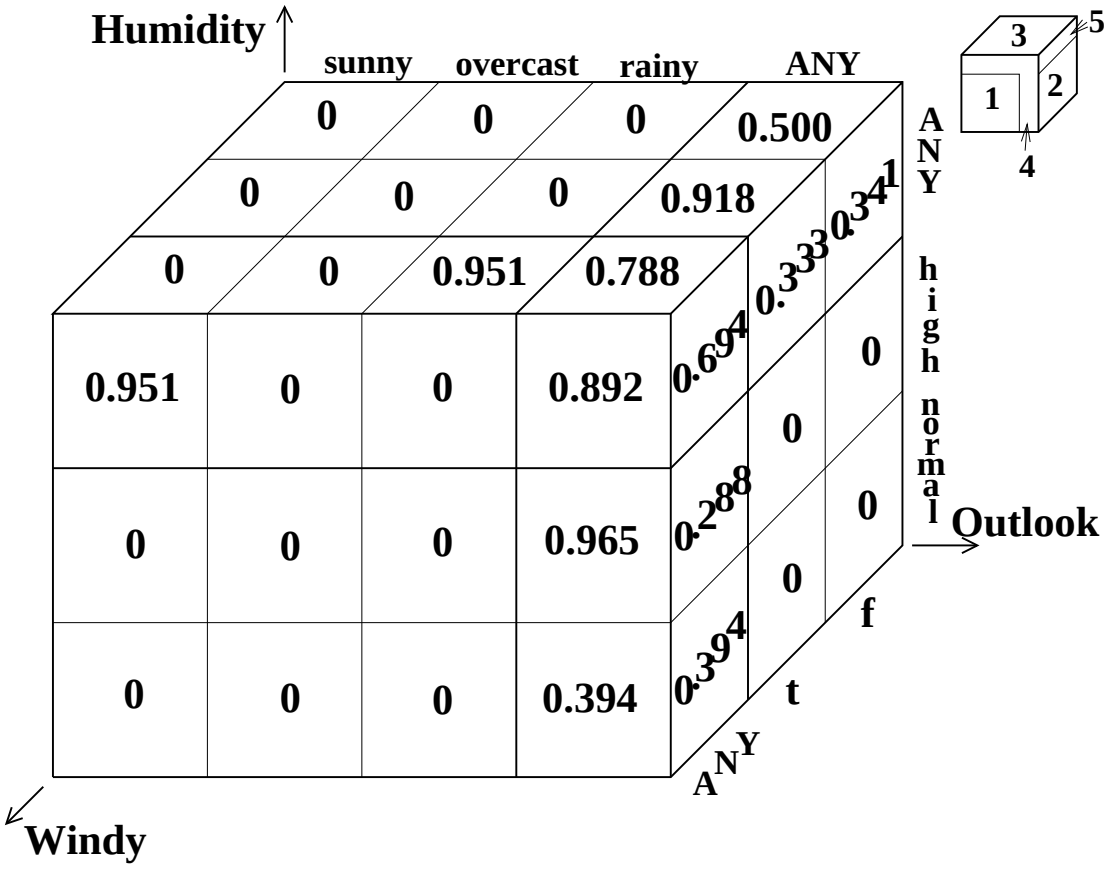
The following are added to *Trning*

Trning	(*Ok*	*Hm*	*Wn*	N	P)
1,8,9	sn	ANY	f	2	1
2,11	sn	ANY	t	1	1
	sn	ANY	ANY	3	2
3,13	ov	ANY	f	0	2
7,12	ov	ANY	t	0	2
	ov	ANY	ANY	0	4
4,14	rn	ANY	f	0	3
5,6,10	rn	ANY	t	2	0
	rn	ANY	ANY	2	3
	ANY	ANY	f	2	6
	ANY	ANY	t	3	3
	ANY	ANY	ANY	5	9

II 2a,b,c General Code, Using Attribute Metadata

```
    let N be totN;
    let P be totP;
    TotAttribs<-{totN, totP}                              ←
    PropAttribs<-{Wn, Ok, Hm}                             ←
    for Attribs in PropAttribs {
        let eval Attrib be "ANY";                         ←
        let totN be equiv + of N
        by (PropAttribs diff Attrib);                     ←
        let totP be equiv + of P
            by (PropAttribs diff Attrib);                 ←
        update Trning add [AllAttribs] in                 ←
            [PropAttribs diff Attrib                      ←
            union TotAttribs]                             ←
            in Trning;
    }
```

III Information DataCube for Decision Tree (loop $2d - 1 = 5$ times)



IV Nested Relations:
(New Syntax only to Add a Level)

ShoppingBasket

(*item*	*xact*)
`milk`	2
`bread`	1
`bread`	2
`bread`	3

let *xactset* **be relation**(*xact*);

(*item*	*xactset*) (*xact*)
`milk`	2
`bread`	1
`bread`	2
`bread`	3

let *xacts* **be equiv union of** *xactset* **by** *item*.

SBsets

(*item*	*xacts*) (*xact*)
`milk`	2
`bread`	1
	2
	3

let *count* **be** [**red** + **of** 1] **in** *xacts*;

SBsetsAgg

(*item*	*xacts* (*xact*)	*count*)
`milk`	2	1
`bread`	1	3
	2	
	3	

Intermediate Details: Raising by One Level

SBsets (*item*	*xacts*) (*xact*)	**red** + **of** 1	*count*
`milk`	2	1	1
`bread`	1	3	3
	2	3	
	3	3	

V The **transpose** Operator

Trning	(*Ok*	*Hm*	*Wn*	N	P)
	`sn`	`hi`	`f`	`2`	`0`
	`sn`	`hi`	`t`	`1`	`0`
	:	:	:	:	:

AllAttribs <– [**red ujoin of**
[*attr*] **transpose** *Ok*, *Hm*, *Wn*, *N*, *P*]
in *Trning*;

Trning	(*Ok*	*Hm*	*Wn*	N	P)	<**transpose**>	
						attr	*val*
	`sn`	`hi`	`f`	`2`	`0`	*Ok*	`sn`
						Hm	`hi`
						Wn	`f`
						N	`2`
						P	`0`
	`sn`	`hi`	`t`	`1`	`0`	*Ok*	`sn`
						Hm	`hi`
						Wn	`t`
						N	`1`
						P	`0`
	:	:	:	:	:	:	: