OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ... The binary setting is simple: Class 0 represents non-dangerous material, and class 1 represents dangerous threat material. One might know how often a class-0 and a class-1 material is occurring, statistically, in a "large stream" of N data. For example, a class-0 material might arise n_0 times, and a class-1 material might arise n_1 times. Since $n_0 + n_1 = N$, the relative occurrence equation $n_0/N + n_1/N = 1$ holds. One can also interpret $u_0 = n_0/N$ and $u_1 = n_1/N$, where $u_0 + u_1 = 1$, as normalized, quasi-barycentric, relative occurrence probabilities of class 0 and class 1. AS THE VALUES OF u_0 AND u_1 ARE "HISTORICAL," I.E., BASED ON A PAST, PREVIOUSLY SEEN DATA STREAM(S), ONE MUST USE THIS INFORMATION CAREFULLY WHEN CLASSIFYING NEW, FUTURE DATA STREAMS.

We consider the eight ($=2^3$) possible streams of three data:

...

	stream		
0	0	0	0
1	0	0	1
2	0	1	0
3	1	0	0
4	0	1	1
5	1	0	1
6	1	1	0
7	1	1	1

All 8 3-data streams.

	0	0	0
C	0, TN	0, TN	0, TN
L	0, TN	0, TN	1, FP
A	0, TN	1, FP	0, TN
S	1, FP	0, TN	0, TN
S	0, TN	1, FP	1, FP
I	1, FP	0, TN	1, FP
F	1, FP	1, FP	0, TN
.	1, FP	1, FP	1, FP

All 8 possible classification results for 0,0,0.

...

	0	1	1
C	0, TN	0, FN	0, FN
L	0, TN	0, FN	1, TP
A	0, TN	1, TP	0, FN
S	1, FP	0, FN	0, FN
S	0, TN	1, TP	1, TP
I	1, FP	0, FN	1, TP
F	1, FP	1, TP	0, FN
.	1, FP	1, TP	1, TP

All 8 possible classification results for 0,1,1.

...

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... Each of the $8 = 2^3$ "3-bit" binary streams induces $8 = 2^3$ possible "3-bit" binary classification streams. The previous page provides the classifications that are combinatorially possible for the two given (true) streams 0,0,0 and 0,1,1.

Considering all available and known "historical" data streams, one might have observed that class-1 (threat) material has a relative occurrence probability of, for example, 1/3. One might potentially want to consider and use this probability in the definition of the classification performance function – WHEN APPROPRIATE. One must keep in mind that such a probability value is subject to change, dynamically increasing or decreasing over time. A performance function would therefore also have to adapt dynamically over time.

We discuss the specific example of "3-bit" binary classification streams in more detail. First, the term "binary" can be mis-leading. This term and its use only refer to the fact that two classes of material are considered, classes 0 and 1. Second, the classification "result type" of these two material classes can be a type of the 4-element set {TN, TP, FP, FN}.

...

Bernd Hamann

Stratovan

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Therefore, in this example, the "3-letter classification result words," having the 4 possible "result types" as "letters" ('TN', 'TP', 'FP' or 'FN') define $2^6 = 64$ combinatorially distinct "words." In order to simplify the notation further one can use integers for "result types": 0 ≙ 'FN', 1 ≙ 'FP', 2 ≙ 'TP' and 3 ≙ 'TN'. The "word" columns included here (left) are merely serving an illustrative purpose. For example, if one were to associate a unique minimal weight with the type 0, then the "word" 000 would represent the worst-possible classification result; this result is high-lighted via '↯'. If one were to associate the same maximal weight with the types 2 and 3, then the eight possible "words" consisting only of these two types would all represent the best-possible classification results; these results are high-lighted via '!'. We can now use this simple example to explore the definition of desirable weight and performance function values.

↯	000	200	
	001	201	
	002	202	
	003	203	
	010	210	
	011	211	
	012	212	
	013	213	
	020	220	
	021	221	
	022	**222**	!
	023	**223**	!
	030	230	
	031	231	
	032	**232**	!
	033	**233**	!
	100	300	
	101	301	
	102	302	
	103	303	
	110	310	
	111	311	
	112	312	
	113	313	
	120	320	
	121	321	
	122	**322**	!
	123	**323**	!
	130	330	
	131	331	
	132	**332**	!
	133	**333**	!

$64 = 4^3$ possible "result words".

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... To clearly explain and better understand the combinatorial facts governing the statistical aspects, we explore the result occurrence patterns for the above example:

word	C	word	C	word	C	word	C
000	●	100	a	200	a	300	a
001	a	101	b	201	A	301	B
002	a	102	A	202	c	302	C
003	a	103	B	203	C	303	d
010	a	110	b	210	A	310	B
011	b	111	●	211	b	311	b
012	A	112	b	212	c	312	D
013	B	113	b	213	D	313	d
020	a	120	A	220	c	320	C
021	A	121	b	221	c	321	D
022	c	122	c	222	●	322	c
023	C	123	D	223	c	323	d
030	a	130	B	230	C	330	d
031	B	131	b	231	D	331	d
032	C	132	D	232	c	332	d
033	d	133	d	233	d	333	●

"3-letter classification results" table with associated case (C) label.

C (case)	#3	#2	#1	#0	P			
●	0	0	0	3	1	4		64
	0	0	3	0	1			
	0	3	0	0	1			
	3	0	0	0	1			
a	0	0	1	2	3	9	36	
	0	1	0	2	3			
	1	0	0	2	3			
b	0	0	2	1	3	9		
	0	1	2	0	3			
	1	0	2	0	3			
c	0	2	0	1	3	9		
	0	2	1	0	3			
	1	2	0	0	3			
d	2	0	0	1	3	9		
	2	0	1	0	3			
	2	1	0	0	3			
A	0	1	1	1	6	24		
B	1	0	1	1	6			
C	1	1	0	1	6			
D	1	1	1	0	6			

Cases (C) of classifications.

The left table organizes the $2^6 = 4^3 = 64$ "3-letter classification result words." These "words" have associated case (C) labels, summarized in the right table:

● $(3 = 3+0+0)$ Four "words" belong to this case, i.e., the "words" that consist of only one letter, repeated **3** times.

a, b, c, d $(3 = 2+1+0)$ 36 "words" belong to these (sub-) cases, i.e., the "words" consisting of two letters, one letter used **2** times.

A, B, C, D $(3 = 1+1+1)$ 24 "words" belong to these (sub-) cases, i.e., the "words" consisting of three different letters, each letter used **1** time.

•••

■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

• Laplacian eigenfunctions and neural networks: ... The right table on the previous page list the numbers of combinatorially possible cases and sub-cases when constructing 3-letter "words" using the "alphabet" {'0', '1', '2', '3'}. The numbers of '0', '1', '2' and '3' in a "word" are called #0, #1, #2 and #3 in the table (columns). The numbers of cases, sub-cases and case/sub-case occurrence probability (P) — not normalized to the unit interval — are combinatorially important, and they might be of interest for the definition of the performance function.

In combinatorics and number theory, the terms COMPOSITION and PARTITION are used to refer to the representation of an integer via sums of integers (usually considering only non-negative integers). The table provides examples:

N	composition tuples of N (k-part compositions)	c#	p#
1	(1)	1	1
2	(2) (1,1)	2	2
3	(3) (2,1) (1,2) (1,1,1)	4	3
4	(4) (3,1) (1,3) (2,2) (2,1,1) (1,2,1) (1,1,2) (1,1,1,1)	8	5
5	(5) (4,1) (1,4) (3,2) (2,3) (3,1,1) (1,3,1) (1,1,3) (2,2,1) (2,1,2) (1,2,2) (2,1,1,1) (1,2,1,1) (1,1,2,1) (1,1,1,2) (1,1,1,1,1)	16	7
⋮		⋮	⋮

e.g., $4 = 1+2+1$

number of compositions: $c\# = \sum_{k=1}^{N} \binom{N-1}{k-1} = 2^{N-1}$

This table lists all possible COMPOSITIONS and PARTITIONS (underlined) of $N = 1, 2, 3, 4, 5$. The number of tuple components is the number of parts (k) to add to obtain N: tuple $\# = (t_1, \dots, t_k) \Rightarrow N = t_1 + \dots + t_k$. PARTITIONS do not consider the order of tuple components. ...