Expert Data Mining

Introduction

This program calculates all possible Boolean vectors for a given dimension and determines the class of each vector by asking a user, reducing the number of questions by using the properties of monotonicity and Hansel Chains. A question in the context of this subject refers to asking the class of a particular vector or datapoint. The executable and code for this program is attached. Please note: the program has minimal error handling for user input, so if the program species a specific input, anything else will crash the program (e.g. entering an "a" when the input should be "1" or "0").

Pilot Questions

There are several different orderings of Hansel Chains that the user can choose: shortest Hansel Chain first, longest Hansel Chain first, and default order (the order output by the algorithm). Moreover, there are several different options that aim to reduce the number of questions further. These options are chain jumping, true attributes, and majority vectors, and the definitions of those lie below.

Definitions

Pilot Questions

The following definitions refer to the options that can be chosen to attempt to reduce the number of questions. All options can be chosen together or separately, except when the option is an ordering (e.g., shortest Hansel Chain first cannot be paired with another order).

Chain jumping — this refers to skipping a Hansel Chain if the first or last vector yields an unsatisfactory answer. Provided that the chains are in monotonically increasing order, if the first vector has a class of 0 (false), no subsequent vectors in that Hansel Chain are expanded, so that Hansel Chain is skipped. Starting at the last vector is also possible: if the last vector has a class of 1 (true), then no preceding vectors in that Hansel Chain are expanded, so that Hansel

Chain is skipped. The idea is that by going to the next Hansel Chain, a prior, skipped Hansel Chain may have its vectors expanded by the next Hansel Chain. Moreover, the class of each vector in that subsequent Hansel Chain may be determined by only one question, thus reducing the number of questions. When chain jumping is chosen as an option, the program will ask whether the user wants to start at the top of the Hansel Chain (1 for yes, 0 for no). Starting at the top is equivalent to starting at the end or last vector of the Hansel Chain. Skipped Hansel Chains are returned to after all Hansel Chains have been visited.

True attributes — a true attribute is an attribute which must be true. For example, in a 5-dimension dataset, if the 5^{th} attribute, *x5,* must be true for that vector to be true, then *x5* is a true attribute; therefore, if a particular vector's 5^{th} attribute is false, then that vector is false. The user can define a true attribute before any questions are asked. Note: only a single true attribute has been tested, but theoretically it should work with multiple.

Majority vectors/majority flag — a majority vector is a vector where half or the ceiling of half of the vector's attributes are true. For a 5-D or 6-D dataset, a majority vector will have 3 attributes which are true. For a 7-D dataset, a majority vector will have 4 attributes which are true. If the user decides to use the majority flag, then all majority vectors are asked first, and once those are done, the normal sequence of questions is asked. If the user does not know roughly how many majority vectors are true, then all majority vectors are asked first. If the user specifies that a number of these majority vectors are true, then we ask at random, and if half of those majority vectors are true, the normal sequence of questions is asked. The idea is that it is likely that for a given dimension, it will take half or slightly over half of the attributes to be true for the vector to be true, but that is not guaranteed. The majority flag can be paired with chain jumping. In this case, after the majority questions are asked, the Hansel Chains which have majority vectors that are true are asked before returning to the normal sequence of Hansel Chains.

Expansions

The following definitions refer to the types of expansions as output by the program.

Expandable 1-1 — expandable 1-1 refers to expansions which occurred in the positive (true) direction. Hence, a vector with a class of 1 expanded another vector.

Expandable 0-0 — expandable 0-0 refers to expansions which occurred in the negative (false) direction. Hence, a vector with a class of 0 expanded another vector.

Unexpandable 1-1 — unexpandable 1-1 refers to expansions which could have but did not occur in the positive (true) direction. Hence, a vector with a class of 1 *could have* expanded this other vector, but it doesn't because that vector was already assigned a class (already asked or already expanded).

Unexpandable 0-0 — unexpandable 0-0 refers to expansions which could have but did not occur in the negative (false) direction. Hence, a vector with a class of 0 *could have* expanded this other vector, but it doesn't because that vector was already assigned a class (already asked or already expanded).

Prior 1-1 and prior 0-0 — these are a special case of unexpandable 1-1 or unexpandable 0-0, respectively, where a vector is not expanded because it was already asked. For example, a vector in Hansel Chain 2 cannot expand a vector in Hansel Chain 1 because Hansel Chain 1 was asked first. Some vectors will not have prior expansions due to semantics: if the majority flag is used, majority vectors are asked first and can expand previous elements, whereas they normally would not be able to. When majority vectors are paired with chain jumping, then it is possible that no vector will have a prior expansion.

Results

Some of the results of the program are simply output to the console, but a more complete version of the data is output to a CSV file, titled "results.csv," which can simply be viewed in Microsoft Excel or some other software. The file is output to the same directory that the code is in. Some of the results that have been attached were at different iterations of the program so there is slight variation, but at the top of the file are the pilot questions and their answers. For example,

	A	B
	Pilot Questions:	
	Default Order	
	Majority Flag not used	
	Static	

The above output means that the default order of Hansel Chains was chose, the majority flag was not used, and it is static, which means there is no chain jumping. If true attributes are used, then it will say so as well as which attribute is true. Furthermore, in the results is the Monotone Boolean Function of the dataset in both a simplified and non-simplified version, the vectors, the ordering of questions that were asked as well as the answers for those, the total number of questions, as well as every expandable, unexpandable, and prior expansion for each vector. Each vector also has a reference number, which is given by the Hansel Chain number and the order in that Hansel Chain (e.g., the first vector in the first Hansel Chain for a specific ordering of Hansel Chains is vector 1.1). Moreover, the class for each vector is given, as well as the majority flag: 1 if it is a majority vector, and 0 if it is not. Lastly, there are 3 columns that are dedicated to the ordering of questions: planned order, updated order, and final order. The planned order is simply the order that was determined from the pilot questions when expansions are not considered. The updated order can be the same as the planned order, but if chain jumping is used it can change. The final order is simply the actual order in which the questions were asked.

A folder of results can be found in the 5-D subdirectory, where a standardized dataset was used. The monotone Boolean function for each results file is *x4x5 v x2x3x5 v x1x3x5.* Moreover, if a file has MajorityFlag0 in the name, it means that the majority flag was used and a number of majority vectors was not given. If it is MajorityFlag5_*n*, then 5 is the number of majority vectors given, and *n* is the iteration of that test (since the order of majority vectors in that case are random). Below there are several figures which show some results.

Total Questions: 15												
Reference Number	Vector	Planned Query Order	Updated Query Order	Final Query Order	Class	Majority Flag	Expanded 1-1	Expanded 0-0	Unexpandable 1-1	Unexpandable 0-0	Prior 1-1	Prior 0-0
1.1	0;0;0;0;0	1	1	1	0	0			7.1;4.1;3.1;2.1;1.2;			
1.2	0;0;0;0;1	2	2	2	0	0			7.2;4.2;3.2;1.3;			1.1;
1.3	0;0;0;1;1	3	3	3	1	0	7.3;4.3;1.4;			2.1;		1.2;
1.4	0;0;1;1;1	4	4		1	1	7.4;1.5;			3.2;2.2;		1.3;
1.5	0;1;1;1;1	5	5		1	0	1.6;			4.3;3.3;2.3;		1.4;
1.6	1;1;1;1;1	6	6		1	0				7.4;4.4;3.4;2.4;		1.5;
2.1	0;0;0;1;0	7	7	4	0	0			8.1;5.1;2.2;		1.3;	1.1;
2.2	0;0;1;1;0	8	8	5	0	0		3.1;	8.2;2.3;		1.4;	2.1;
2.3	0;1;1;1;0	9	9	6	0	1		5.1;6.1;	2.4;		1.5;	2.2;
2.4	1;1;1;1;0	10	10	7	0	0		8.2;5.2;6.2;			1.6;	2.3;
3.1	0;0;1;0;0	11	11		0	0			9.1;6.1;3.2;		2.2;	1.1;
3.2	0;0;1;0;1	12	12	8	0	0			9.2;3.3;		1.4;	1.2;3.1;
3.3	0;1;1;0;1	13	13	9	1	1	3.4;			4.2;6.1;	1.5;	3.2;
3.4	1;1;1;0;1	14	14		1	0				9.2;10.2;6.2;	1.6;	3.3;
4.1	0;1;0;0;0	15	15	10	0	0			10.1;6.1;5.1;4.2;			1.1;
4.2	0;1;0;0;1	16	16	11	0	0			10.2;4.3;		3.3;	1.2;4.1;
4.3	0;1;0;1;1	17	17		1	1	4.4;			5.1;	1.5;	1.3;4.2;
4.4	1;1;0;1;1	18	18		1	0				7.3;10.2;5.2;	1.6;	4.3;
5.1	0;1;0;1;0	19	19		0	0			5.2;		2.3;4.3;	2.1;4.1;
5.2	1;1;0;1;0	20	20		0	1		8.1;10.1;			2.4;4.4;	5.1;
6.1	0;1;1;0;0	21	21		0	0			6.2;		2.3;3.3;	3.1;4.1;
6.2	1;1;1;0;0	22	22		0	1		9.1;		10.1;	2.4;3.4;	6.1;
7.1	1;0;0;0;0	23	23	12	0	0			10.1;9.1;8.1;7.2;			1.1;
7.2	1;0;0;0;1	24	24	13	0	0			10.2;9.2;7.3;			1.2;7.1;
7.3	1;0;0;1;1	25	25		1	1			7.4;	8.1;	4.4;	1.3;7.2;
7.4	1;0;1;1;1	26	26		1	0				9.2;8.2;	1.6;	1.4;7.3;
8.1	1;0;0;1;0	27	27		0	0			8.2;		5.2;7.3;	2.1;7.1;
8.2	1;0;1;1;0	28	28		0	1				9.1;	2.4;7.4;	2.2;8.1;
9.1	1;0;1;0;0	29	29		0	0			9.2;		6.2;8.2;	3.1;7.1;
9.2	1;0;1;0;1	30	30	14	1	1					3.4;7.4;	3.2;7.2;9.1;
10.1	1;1;0;0;0	31	31		0	0			10.2;		6.2;5.2;	4.1;7.1;
10.2	1;1;0;0;1	32	32	15	0	1					3.4;4.4;	4.2;7.2;10.1;

Figure 1: Default order, majority flag not used, and static.

Total Questions: 8												
Reference Number	Vector	Planned Query Order	Updated Query Order	Final Query Order	Class	Majority Flag	Expanded 1-1	Expanded 0-0	Unexpandable 1-1	Unexpandable 0-0	Prior 1-1	Prior 0-0
1.1	0;0;0;0;0	1	1		0	0			5.1;4.1;3.1;2.1;1.2;			
1.2	0;0;0;0;1	2	2	1	0	0			5.2;4.2;3.2;1.3;			1.1;
1.3	0;0;0;1;1	3	3	2	1	0	5.3;4.3;1.4;			2.1;		1.2;
1.4	0;0;1;1;1	4	4		1	1	5.4;1.5;			3.2;2.2;		1.3;
1.5	0;1;1;1;1	5	5		1	0	1.6;			4.3;3.3;2.3;		1.4;
1.6	1;1;1;1;1	6	6		1	0				5.4;4.4;3.4;2.4;		1.5;
2.1	0;0;0;1;0	7	7		0	0			8.1;6.1;2.2;		1.3;	1.1;
2.2	0;0;1;1;0	8	8		0	0		3.1;	8.2;2.3;		1.4;	2.1;
2.3	0;1;1;1;0	9	9		0	1		6.1;7.1;	2.4;		1.5;	2.2;
2.4	1;1;1;1;0	10	10		0	0		8.2;6.2;7.2;			1.6;	2.3;
3.1	0;0;1;0;0	11	11		0	0			9.1;7.1;3.2;		2.2;	1.1;
3.2	0;0;1;0;1	12	12	3	0	0			9.2;3.3;		1.4;	1.2;3.1;
3.3	0;1;1;0;1	13	13	4	1	1	3.4;			4.2;7.1;	1.5;	3.2;
3.4	1;1;1;0;1	14	14		1	0				9.2;10.2;7.2;	1.6;	3.3;
4.1	0;1;0;0;0	15	15		0	0			10.1;7.1;6.1;4.2;			1.1;
4.2	0;1;0;0;1	16	16	5	0	0			10.2;4.3;		3.3;	1.2;4.1;
4.3	0;1;0;1;1	17	17		1	1	4.4;			6.1;	1.5;	1.3;4.2;
4.4	1;1;0;1;1	18	18		1	0				5.3;10.2;6.2;	1.6;	4.3;
5.1	1;0;0;0;0	19	19		0	0			10.1;9.1;8.1;5.2;			1.1;
5.2	1;0;0;0;1	20	20	6	0	0			10.2;9.2;5.3;			1.2;5.1;
5.3	1;0;0;1;1	21	21		1	1			5.4;	8.1;	4.4;	1.3;5.2;
5.4	1;0;1;1;1	22	22		1	0				9.2;8.2;	1.6;	1.4;5.3;
6.1	0;1;0;1;0	23	23		0	0			6.2;		2.3;4.3;	2.1;4.1;
6.2	1;1;0;1;0	24	24		0	1		8.1;10.1;			2.4;4.4;	6.1;
7.1	0;1;1;0;0	25	25		0	0			7.2;		2.3;3.3;	3.1;4.1;
7.2	1;1;1;0;0	26	26		0	1		9.1;		10.1;	2.4;3.4;	7.1;
8.1	1;0;0;1;0	27	27		0	0			8.2;		6.2;5.3;	2.1;5.1;
8.2	1;0;1;1;0	28	28		0	1				9.1;	2.4;5.4;	2.2;8.1;
9.1	1;0;1;0;0	29	29		0	0			9.2;		7.2;8.2;	3.1;5.1;
9.2	1;0;1;0;1	30	30	7	1	1					3.4;5.4;	3.2;5.2;9.1;
10.1	1;1;0;0;0	31	31		0	0			10.2;		7.2;6.2;	4.1;5.1;
10.2	1;1;0;0;1	32	32	8	0	1					3.4;4.4;	4.2;5.2;10.1;

Figure 2: Longest Hansel Chain first, True attributes: x5, majority flag not used, and static.

Total Questions: 15												
Reference Number	Vector	Planned Query Order	Updated Query Order	Final Query Order	Class	Majority Flag	Expanded 1-1	Expanded 0-0	Unexpandable 1-1	Unexpandable 0-0	Prior 1-1	Prior 0-0
1.1	0;0;0;0;0	11	15		0	0			5.1;4.1;3.1;2.1;1.2;			
1.2	0;0;0;0;1	12	14	13	0	0		1.1;	5.2;4.2;3.2;1.3;			
1.3	0;0;0;1;1	13	13	12	1	0			5.3;4.3;1.4;	1.2;2.1;		
1.4	0;0;1;1;1	1	1	1	1	1	5.4;1.5;			1.3;3.2;2.2;		
1.5	0;1;1;1;1	14	12		1	0			1.6;	1.4;4.3;3.3;2.3;		
1.6	1;1;1;1;1	15	11	11	1	0				1.5;5.4;4.4;3.4;2.4;		
2.1	0;0;0;1;0	16	27		0	0			8.1;6.1;2.2;1.3;	1.1;		
2.2	0;0;1;1;0	17	26		0	0		2.1;	8.2;2.3;1.4;	3.1;		
2.3	0;1;1;1;0	2	2	2	0	1		2.2;6.1;7.1;	2.4;1.5;			
2.4	1;1;1;1;0	18	25	15	0	0			1.6;	2.3;8.2;6.2;7.2;		
3.1	0;0;1;0;0	19	18		0	0			9.1;7.1;2.2;3.2;	1.1;		
3.2	0;0;1;0;1	20	17	14	0	0		3.1;	9.2;3.3;1.4;	1.2;		
3.3	0;1;1;0;1	3	3	3	1	1	3.4;		1.5;	3.2;4.2;7.1;		
3.4	1;1;1;0;1	21	16		1	0			1.6;	3.3;9.2;10.2;7.2;		
4.1	0;1;0;0;0	22	21		0	0			10.1;7.1;6.1;4.2;	1.1;		
4.2	0;1;0;0;1	23	20		0	0		4.1;	10.2;3.3;4.3;	1.2;		
4.3	0;1;0;1;1	4	4	4	1	1	4.4;		1.5;	1.3;4.2;6.1;		
4.4	1;1;0;1;1	24	19		1	0			1.6;	4.3;5.3;10.2;6.2;		
5.1	1;0;0;0;0	25	24		0	0			10.1;9.1;8.1;5.2;	1.1;		
5.2	1;0;0;0;1	26	23		0	0		5.1;	10.2;9.2;5.3;	1.2;		
5.3	1;0;0;1;1	5	5	5	1	1			4.4;5.4;	1.3;5.2;8.1;		
5.4	1;0;1;1;1	27	22		1	0			1.6;	1.4;5.3;9.2;8.2;		
6.1	0;1;0;1;0	28	28		0	0			6.2;2.3;4.3;	2.1;4.1;		
6.2	1;1;0;1;0	6	6	6	0	1		8.1;10.1;	2.4;4.4;	6.1;		
7.1	0;1;1;0;0	29	29		0	0			7.2;2.3;3.3;	3.1;4.1;		
7.2	1;1;1;0;0	7	7	7	0	1		9.1;	2.4;3.4;	7.1;10.1;		
8.1	1;0;0;1;0	30	30		0	0			6.2;8.2;5.3;	2.1;5.1;		
8.2	1;0;1;1;0	8	8	8	0	1			2.4;5.4;	2.2;8.1;9.1;		
9.1	1;0;1;0;0	31	31		0	0			7.2;8.2;9.2;	3.1;5.1;		
9.2	1;0;1;0;1	9	9	9	1	1			3.4;5.4;	3.2;5.2;9.1;		
10.1	1;1;0;0;0	32	32		0	0			7.2;6.2;10.2;	4.1;5.1;		
10.2	1;1;0;0;1	10	10	10	0	1		4.2;5.2;	3.4;4.4;	10.1;		

Figure 3: Longest Hansel Chain First, majority flag used (0), and chain jump starting from the top of the Hansel Chain.