Paper 118-2008

New Strategies for Identifying Customer Use Patterns

Barry de Ville, SAS Institute Inc., Cary, NC

ABSTRACT

Hardware devices that are used for communications possess a wide variety of programmable hardware components, called slots. You can configure slots in multiple ways to deliver a variety of communications services such as file transfer, remote computer access, internet page access, and voice-over-internet capabilities.

It is extremely difficult to create high level, summary descriptions of a collection of hardware devices. Any given hardware device could have one or more programmable hardware slots, and each slot can be any one of multiple family types.

Because of the many possible configurations, it is hard to identify common combinations that could be particularly useful in a given application area. Alternatively, it is difficult to identify common configurations that generate an inordinate number of faults, for example. This places a premium on deriving common configurations as an incentive to the dissemination of best practices on the one hand, and as a vehicle for superior quality control and trouble-shooting on the other.

A simple but effective summarization technique that accomplishes the objective of summarization is presented here. We show how over 6,000 different combinations of hardware configurations can be effectively summarized in only five clusters.

PROBLEM DESCRIPTION

Table 1 illustrates the various family types that you can use in combination to characterize one particular type of communications device. These family types are fabricated, but an example label has been included as an aid to understanding.

Table 1: Distribution of Slot Type on Communications Hardware (Example)

HW Slot	Label	Frequency	Percent
SEC	Security	450	9.79
IP	Internet	26	0.57
VPN	Virtual Network	2235	48.62
WIRE	Hard Wire	83	1.81
MX	Multiplex	434	9.44

HW Slot	Label	Frequency	Percent
AIR	Wireless	19	0.41
AAIR	Advanced Wireless	1350	29.37

To grasp the magnitude of the problem, assume that a device can have as many as five slots and that each slot can be configured with any one of these seven hardware types. Many of these potential combinations exist in deployments of these devices internationally, but there are also common patterns that characterize the majority of the devices.

The problem is to discover these common patterns. Compressing or representing a multiplicity of possibilities into a smaller set of categories is a typical problem of dimensional reduction. Dimensional reduction problems are often solved through the application of numerical techniques such as cluster analysis, factor analysis, or principal components.

An alternative approach is to use Market Basket or Association Rules to identify the most frequently occurring combinations (or baskets) among the various 1-, 2-, 3-, 4-, or 5-slot devices. Association rules, however, do not identify common baskets according to the specific configuration of the hardware as is required here.

The alternative solution presented here relies on a technique that identifies common combinations in two ways:

- as one number that consists of a concatenation of all possible categories
- as a cluster formed through the application of a decision tree

Because many slot types are collapsed into one, concatenated code, there is an immediate benefit of flattening multiple dimensions into one, single dimension. This is equivalent to producing the full range of potential market baskets.

Secondly, a decision tree analysis is undertaken that uses the hardware configuration as the target or dependent variable. Clustering common combinations within various hardware configurations is then relatively simple.

SOLUTION APPROACH

CREATE THE HYPERCODE. The solution begins with the creation of a single field, which is called a hypercode. The hypercode captures all of the the unique, multiple hardware configurations that are possible, given specific combinations of slot types within hardware devices. This field is created by first sorting the characteristic slots that exist on any one hardware device in the collection, then creating a multi-dimensional representation of the hardware slots that is based on a concatenation of all codes together.

Thus, if a particular hardware device contained three VPN slots, an MX slot, and an AAIR slot, then the hypercode would be as follows:

VPN+VPN+VPN+MX+AAIR

The table below provides an example in which we see that the Hardware ID is 120. There are five slots on this hardware device. Three slots are VPN types. Slots 4 and 5 are MX and AAIR, respectively.

Table 2: Example Source Record for a Hardware Device in the Analysis

Slot	1	2	3	4	5
Hw ID	120	120	120	120	120
Slot Type	VPN	VPN	VPN	MX	AAIR

To create a hypercode for this device, first sort the Slot Types . A discrete record is produced for each slot within the Hardware ID. Then a concatenated hypercode is built up across the individual records, and the final hypercode is written out.

Table 3: Example Hardware Device Hypercode Derivation

	Hw ID	Slot Type	hyper	final
1	120	VPN	VPN	VPN+VPN+VPN+MX+AAIR
2	120	VPN	VPN+VPN	VPN+VPN+VPN+MX+AAIR
3	120	VPN	VPN+VPN+VPN	VPN+VPN+VPN+MX+AAIR
4	120	MX	VPN+VPN+VPN+MX	VPN+VPN+VPN+MX+AAIR
5	120	AAIR	VPN+VPN+VPN+MX+AAIR	VPN+VPN+VPN+MX+AAIR

The final record representation is shown in Table 4.

Table 4: Example Device with Hypercode

Slot	1	2	3	4	5	Hypercode
Hw ID	120	120	120	120	120	120
Slot Type	VPN	VPN	VPN	MX	AAIR	VPN+VPN+VPN+MX+AAIR

The code that produces this follows.

```
proc sort data=indata;
   by HWid SlotType;
run;

data indata_hyper;
   set indata;
length hyper $40;
   retain hyper;
   by HWid;

if first.HWid then hyper = trim(slotType);
else hyper =  trim(hyper) || "+" || trim(slotType);
run;
```

This creates a set of representative hypercodes that are based on the example slot families shown in Table 1.

This process of turning multiple hardware slots into a single vector representation takes a hardware entity with a multi-dimensional attribute (various hardware slots) and reduces the attribute to a single-dimensional form (the hypercode).

CLUSTER SIMILAR CODES TOGETHER. The next step in the solution is to create a clustering of these hypercodes such that hardware devices with similar configurations are grouped together. Ideally, when such a cluster or grouping is formed, different configurations that are not a part of this initial cluster are separated into other groups, which, in turn, also represent devices with similar characteristics. The resulting clusters are as similar as possible, but they retain differences between the clusters.

The widely used technique of decision trees creates clusters based on a succession of variable inputs that are used as partitioning rules in the creation of branches and descendent nodes (or leaves). The normal approach begins with a root node and works through descendent nodes recursively until a tree-growth stopping rule is encountered (1)

A typical form of a decision tree is shown in Figure 1 (2).

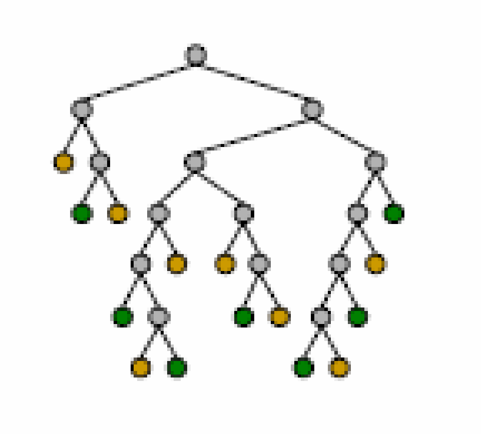

Figure 1: Top-Down Cluster Development in Trees

Here we see the root node at the top of the diagram followed by as many as six levels of binary partitions. Only the terminal nodes (shown in color) are presented in this particular illustration of a decision tree. Each of these terminal nodes represents a cluster. You can find the attributes that define a cluster by tracing up the respective branches from the terminal node to identify the input fields that have been used as partitioning variables.

As shown in Figure 2 (3), various forms of trees and therefore various forms of clusters can be produced depending on whether binary splitting is used to identify the branches of the tree, or whether some k-way split is used. In the figure, splits are shown for two-, three-, four-, five-, and six-way splits. In the example below we use a five-way split. This results in a very simple decision rule and correspondingly simple description of the hardware clusters.

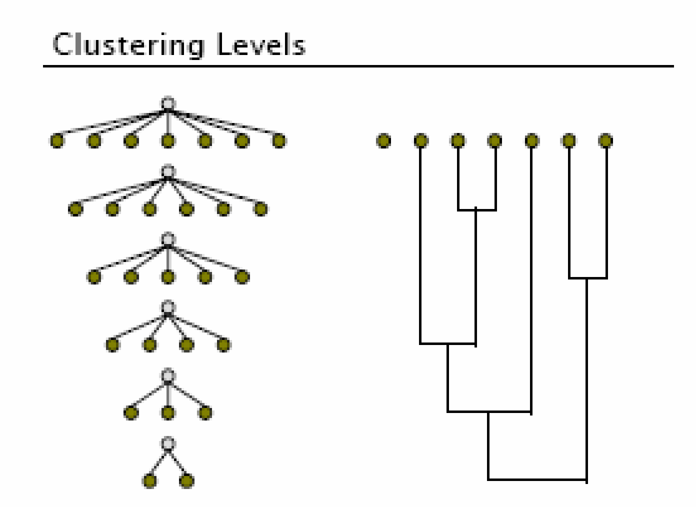

Figure 2: Two-way and Multi-way Branches Illustrated

The final process combines concatenated hypercodes and decision trees, as shown in Figure 3 below.

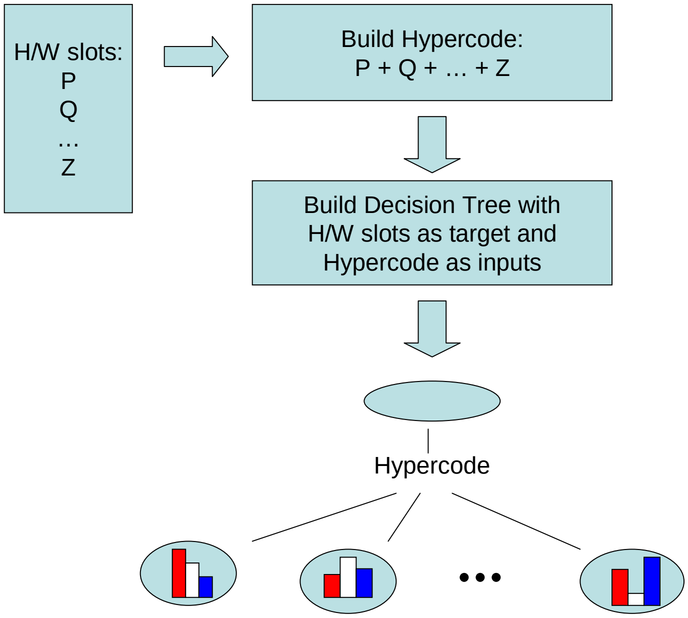

Figure 3: A Schematic of the Complete Cluster-Hypercode Process

In the example analysis presented here, the unclustered hardware slots, which are shown in Table 1 above, are used as the target value in the analysis. The associated hypercodes are used as inputs. Because we know what the individual slots for a given site are, we can determine what combinations of slots (hypercodes) tend to be associated with a given configuration of slots.

The overall results are displayed in Figure 4 below. Because we use a five-way branch partition on the first input, we need to partition the root node of the decision tree only once: we derive five clusters (or five terminal nodes) with one, single partition on the hypercode input variable.

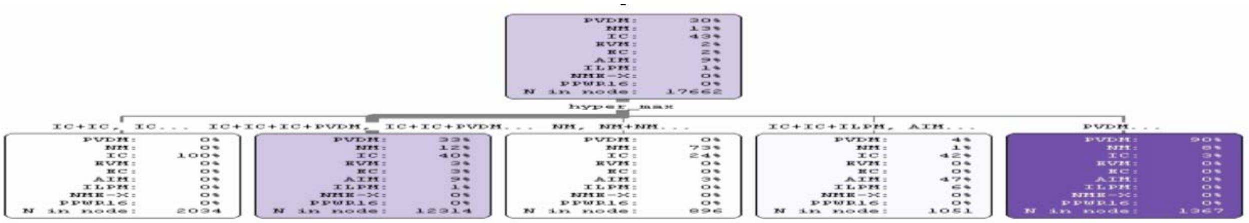

Figure 4: Top Level of the Decision Tree Showing Five Baseline Clusters

The attributes of the five descendent nodes are shown in the following table. Terminal node contents or clusters are described moving from left to right.

Table 5: Detailed Contents of the Clusters (Terminal Nodes)

Node (Cluster)	Associated HyperSlots	Modal Categories
1	VPN+VPN, VPN, VPN+VPN+VPN, VPN+VPN+VPN+VPN	VPN (100%)
2	VPN+VPN+VPN+AAIR, VPN+VPN+AAIR, VPN+VPN+VPN+VPN+AAIR+AAIR, VPN+AAIR+AAIR, VPN+VPN+VPN+VPN+AAIR, VPN+VPN+VPN+MX, VPN+VPN+AAIR+AAIR+AAIR+AAIR, MX+MX+AAIR+AAIR+AAIR, VPN+VPN+VPN+VPN+VPN+VPN+MX+MX, VPN+VPN+VPN+AAIR+AAIR+AAIR, VPN+AAIR+AAIR+AAIR, EVM+VPN+VPN+MX+AAIR+AAIR, MX+MX+AAIR, SIP+AAIR+AAIR, VPN+VPN+VPN+VPN+MX, VPN+VPN+VPN+VPN+VPN+VPN+VPN+VPN+VPN+MX+MX, VPN+MX+AAIR+AAIR+AAIR+AAIR+AAIR, VPN+VPN+VPN+VPN+VPN+VPN+VPN+VPN+VPN+VPN+MX, SIP+MX+MX+MX, MX+MX+AAIR+AAIR, VPN+VPN+AAIR, VPN+VPN+VPN+MX+AAIR+AAIR+AAIR+AAIR, VPN+VPN+VPN+VPN+MX+AAIR+AAIR, VPN+VPN+AAIR+AAIR+AAIR, IP+IP+EVM+AAIR+AAIR, SIP+SIP+AAIR, SIP+VPN+MX+MX, IP+IP+IP+IP+EVM+EVM+AAIR, EVM+VPN+AAIR, IP+IP+VPN+VPN+MX+AAIR, SIP+SIP+EVM+AAIR	AAIR (33%) VPN (12%) MX (12%) SEC (9%)
3	MX, MX+MX, VPN+MX, VPN+VPN+MX, MX+MX+MX, SIP+MX+MX, SIP+MX+MX+AAIR, VPN+MX+MX+MX, VPN+MX+MX, MX+MX+MX+MX, VPN+VPN+VPN+VPN+MX+MX+MX	MX (73%) VPN (24%)
4	VPN+VPN+WIRE, SIP, SIP+VPN+VPN, SIP+VPN+VPN+VPN, SIP+SIP+VPN+MX+AAIR, VPN+VPN+VPN+WIRE+AAIR, VPN+WIRE, WIRE, SIP+VPN+VPN+VPN+VPN, SIP+SIP+AAIR+AAIR, SIP+VPN+VPN+VPN+VPN+AAIR+AAIR, VPN+VPN+VPN+WIRE, SIP+SIP+VPN, SIP+SIP+MX+AIR+AAIR	SIP (47%) MX (42%)

5	AAIR, MX+MX+AAIR+AAIR+AAIR+AAIR, AAIR+AAIR, VPN+MX+AAIR+AAIR, MX+AAIR+AAIR+AAIR+AAIR+AAIR+AAIR, VPN+MX+MX+AAIR+AAIR+AAIR+AAIR+ AIR, AAIR+AAIR+AAIR, MX+AAIR+AAIR+AAIR+AAIR, MX+AAIR+AAIR+AAIR, MX+MX+MX+MX+AAIR+AAIR+AAIR+AAIR, MX+MX+AAIR+AAIR+AAIR+AAIR+AAIR, AAIR+AAIR+AAIR+AAIR, MX+MX+MX+AAIR+AAIR+AAIR+AAIR+AIR	AAIR (90%) MX (9%)

CLUSTER DESCRIPTIONS

Cluster 1 presents a simple result: when hardware devices consist of a varying number of VPN slots, then the slot combinations themselves are a series of VPN slots. In the results we can see that 1, 2, 3, and 4-way concatenations of JD slot types are presented.

Cluster 2 is more interesting. This set of communications devices contain mostly expanded wireless AAIR (33%), VPN (12%), MX (12%), and SEC (9%) slot types (the percentages show the relative percentage of each of these slot types across the entire family of slot types available in this cluster). This means that a third of the slots in this set of hardware devices are AAIR slots. The frequencies for the other slot types indicate that VPN, MX, and SEC slots are present (approximately 10% each). One sequence of hypercodes (VPN+VPN+VPN+MX+AAIR+AAIR+AAIR+AAIR) indicates that as many as eight slot locations are present among these devices.

If we assume that we could configure each of these slot types two or three ways, in various sequences, across a total of eight possible slots, it is easy to see that over 10,000 combinations are possible. The decision tree results shown here indicate that these types of communications devices tend to have as many as 24 configurations, as shown on the table. The most frequent combination, shown first in the list, is VPN+VPN+VPN+AAIR. So we could characterize Cluster 2 as a predominantly VPN service device, potentially for expanded wireless applications.

Clusters 3, 4, and 5 resolve to 10, 14, and 13 main configurations, respectively. As with cluster 1 and 2 results, we observe a significant reduction in the total possible number of configurations.

TRACTABLE RESULTS

The results shown here indicate that extremely complex hardware configurations can be effectively summarized using a hypercode concatenation technique coupled with decision trees. In this case we are able to present thousands of combinations in five clusters. The clusters contain 1, 24, 10, 14, and 13 combinations, respectively: 62 combinations in total.

In Cluster 2 we saw that with eight possible slots and four primary hardware components (AAIR, VPN, MX and SEC), there were millions of potential combinations, but only 24 combinations were identified in the cluster. The most common combination, VPN+VPN+VPN+AAIR, could be used for targeted marketing, to suggest best practices for this type of hardware device.

REFERENCES

1. For a more extensive introduction to decision trees, see deVille, Barry. 2006. ***Decision Trees for Business Intelligence and Data Mining Using SAS Enterprise Miner,*** Cary, N.C: SAS Institute Inc.
2. The figures are extracted from the 2005 publication ***Decision Tree Modeling Course Notes***, developed by SAS Education (author: William J.E. Potts. Revised by Lorne Rothman). Cary, NC: SAS Institute Inc. See page 70.
3. As in reference 2. Page 42.

ACKNOWLEGEMENTS

I was fortunate to receive numerous constructive suggestions for improvement from three content reviewers, John Bander, Sanford Gayle, and Ken King, who struggled through early versions of the draft. In particular, Sanford Gayle suggested the similarity between this approach and market basket analysis. Many thanks to Jan Wheaton who made numerous improvements in the final edit.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Barry de Ville
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919-677-8000
Fax: 919-677-4444
Email: barry.deville@sas.com