Chapter 11: The Data Survey

Overview

Suppose that three separate families are planning a vacation. The Abbott family really enjoys lake sailing. Their ideal vacation includes an idyllic mountain lake, surrounded by trees, with plenty of wildlife and perhaps a small town or two nearby in case supplies are needed. They need only a place to park their car and boat trailer, a place to launch the boat, and they are happy.

The Bennigans are amateur archeologists. There is nothing they like better than to find an ancient encampment, or other site, and spend their time exploring for artifacts. Their four-wheel-drive cruiser can manage most terrain and haul all they need to be entirely self-sufficient for a couple of weeks exploring—and the farther from civilization, the better they like it.

The Calloways like to stay in touch with their business, even while on vacation. Their ideal is to find a luxury hotel in the sun, preferably near the beach but with nightlife. Not just any nightlife; they really enjoy cabaret, and would like to find museums to explore and other places of interest to fill their days.

These three families all have very different interests and desires for their perfect vacation. Can they all be satisfied? Of course. The locations that each family would like to find and enjoy exist in many places; their only problem is to find them and narrow down the possibilities to a final choice. The obvious starting point is with a map. Any map of the whole country indicates broad features—mountains, forests, deserts, lakes, cities, and probably roads. The Abbotts will find, perhaps, the Finger Lakes in upstate New York a place to focus their attention. The Bennigans may look at the deserts of the Southwest, while the Calloways look to Florida. Given their different interests, each family starts by narrowing down the area of search for their ideal vacation to those general areas of the country that seem likely to meet their needs and interests.

Once they have selected a general area, a more detailed map of the particular territory lets each family focus in more closely. Eventually, each family will decide on the best choice they can find and leave for their various vacations. Each family explores its own vacation site in detail. While the explorations do not seem to produce maps, they reveal small details—the very details that the vacations are aimed at. The Abbotts find particular lake coves, see particular trees, and watch specific birds and deer. The Bennigans find individual artifacts in specific places. The Calloways enjoy particular cabaret performers and see specific exhibits at particular museums. It is these detailed explorations that each family feels to be the whole purpose for their vacations.

Each family started with a general search to find places likely to be of interest. Their initial search was easy. The U.S. Geological Survey has already done the hard work for them. Other organizations, some private survey companies, have embellished maps in particular ways and for particular purposes—road maps, archeological surveys, sailing maps (called “charts”), and so on. Eventually, the level of detail that each family needed was more than a general map could provide. Then the families constructed their own maps through detailed exploration.

What does this have to do with data mining? The whole purpose of the data survey is to help the miner draw a high-level map of the territory. With this map, a data miner discovers the general shape of the data, as well as areas of danger, of limitation, and of usefulness. With a map, the Abbotts avoided having to explore Arizona to see if any lakes suitable for sailing were there. With a data survey, a miner can avoid trying to predict the stock market from meteorological data. “Everybody knows” that there are no lakes in Arizona. “Everybody knows” that the weather doesn’t predict the stock market. But these “everybodies” only know that through experience—mainly the experience of others who have been there first. Every territory needed exploring by pioneers—people who entered the territory first to find out what there was in general—blazing the trail for the detailed explorations to follow. The data survey provides a miner with a map of the territory that guides further exploration and locates the areas of particular interest, the areas suitable for mining. On the other hand, just as with looking for lakes in Arizona, if there is no value to be found, that is well to know as early as possible.

11.1 Introduction to the Data Survey

This chapter deals entirely with the data survey, a topic at least as large as data preparation. The introduction to the use, purposes, and methods of data surveying in this chapter discusses how prepared data is used during the survey. Most, if not all, of the surveying techniques can be automated. Indeed, the full suite of programs from which the data preparation demonstration code is drawn is a full data preparation and survey tool set. This chapter touches only on the main topics of data surveying. It is an introduction to the territory itself. The introduction starts with understanding the concept of “information.”

This book mentions “information” in several places. “Information is embedded in a data set.” “The purpose of data preparation is to best expose information to a mining tool.” “Information is contained in variability.” Information, information, information. Clearly, “information” is a key feature of data preparation. In fact, information—its discovery, exposure, and understanding—is what the whole preparation-survey-mining endeavor is about. A data set may represent information in a form that is not easily, or even at all, understandable by humans. When the data set is large, understanding significant and salient points becomes even more difficult. Data mining is devised as a tool to transform the impenetrable information embedded in a data set into understandable relationships or predictions.

However, it is important to keep in mind that mining is not designed to extract information. Data, or the data set, enfolds information. This information describes many and various relationships that exist enfolded in the data. When mining, the information is being mined for what it contains—an explanation or prediction based on the embedded relationships. It is almost always an explanation or prediction of specific details that solves a problem, or answers a question, within the domain of inquiry—very often a business problem. What is required as the end result is *human understanding* (enabling, if necessary, some action). Examining the nature of, and the relationships in, the information content of a data set is a part of the task of the data survey. It prepares the path for the mining that follows.

Some information is always present in the data—understandable or not. Mining finds relationships or predictions embedded in the information inherent in a data set. With luck, they are not just the obvious relationships. With more luck, they are also useful. In discovering and clarifying some novel and useful relationship embedded in data, data mining has its greatest success. Nonetheless, the information exists prior to mining. The data set enfolds it. It has a shape, a substance, a structure. In some places it is not well defined; in others it is bright and clear. It addresses some topics well; others poorly. In some places, the relationships are to be relied on; in others not. Finding the places, defining the limits, and understanding the structures is the purpose of data surveying.

The fundamental question posed by the data survey is, “Just what information is in here anyway?”

11.2 Information and Communication

Everything begins with information. The data set embeds it. The data survey surveys it. Data mining translates it. But what exactly is *information*? The *Oxford English Dictionary* begins its definition with “The act of informing, . . .” and continues in the same definition a little later, “Communication of instructive knowledge.” The act referred to is clearly one where this thing, “information,” is passed from one person to another. The latter part of the definition explicates this by saying it is “communication.” It is in this sense of communicating intelligence—transferring insight and understanding—that the term “information” is used in data mining. Data possesses information only in its latent form. Mining provides the mechanism by which any insight potentially present is explicated. Since information is so important to this discussion, it is necessary to try to clarify, and if possible quantify, the concept.

Because information enables the transferring of insight and understanding, there is a sense in which quantity of information relates to the amount of insight and understanding generated; that is, more information produces greater insight. But what is it that creates greater insight?

A good mystery novel—say, a detective story—sets up a situation. The situation described includes all of the necessary pieces to solve the mystery, but in a nonobvious way. Insight comes when, at the end of the story, some key information throws all of the established structure into a suddenly revealed, surprising new relationship. The larger and more complex the situation that the author can create, the greater the insight when the true situation is revealed. But in addition to the complexity of the situation, it seems to be true that the more surprising or unexpected the solution, the greater the insight.

The detective story illustrates the two key ingredients for insight. The first is what for a detective story is described as “the situation.” The situation comprises a number of individual components and the relationship between the components. For a detective story, these components are typically the characters, the attributes of characters, their relationship to one another, and the revealed actions taken by each during the course of the narrative. These various components, together with their relationships, form a knowledge structure. The second ingredient is the communication of a key insight that readjusts the knowledge structure, changing the relationship between the components. The amount of insight seems intuitively related to how much readjustment of the knowledge structure is needed to include the new insight, and the degree to which the new information is unexpected.

As an example, would you be surprised if you learned that to the best of modern scientific knowledge, the moon really is made of green cheese? Why? For a start, it is completely unexpected. Can you honestly say that you have ever given the remotest credence to the possibility that the moon might really be made of green cheese? If true, such a simple communication carries an enormous amount of information. It would probably require you to reconfigure a great deal of your knowledge of the world. After all, what sort of possible rational explanation could be constructed to explain the existence of such a phenomenon? In fact, it is so unlikely that it would almost certainly take much repetition of the information in different contexts (more evidence) before you would accept this as valid. (Speaking personally, it would take an enormous readjustment of my world view to accept any rational explanation that includes several trillion tons of curdled milk products hanging in the sky a quarter of a million miles distant!)

These two very fundamental points about information—how surprising the communication is, and how much existing knowledge requires revision—both indicate something about how much information is communicated. But these seem very subjective measures, and indeed they are, which is partly why defining information is so difficult to come to grips with.

Claude E. Shannon did come to grips with the problem in 1948. In what has turned out to be one of the seminal scientific papers of the twentieth century, “A Mathematical Theory of Communication,” he grappled directly with the problem. This was published the next year as a book and established a whole field of endeavor, now called “information theory.” Shannon himself referred to it as “communication theory,” but its effects and applicability have reached out into a vast number of areas, far beyond communications. In at least one sense it is only about communications, because unless information is communicated, it informs nothing and no one. Nonetheless, information theory has come to describe information as if it were an object rather than a process. A more detailed look at information will assume where needed, at least for the sake of explanation, that it exists as a thing in itself.

11.2.1 Measuring Information: Signals and Dictionaries

Information comes in two pieces: 1) an informing communication and 2) a framework in which to interpret the information. For instance, in order to understand that the moon is made of green cheese, you have to know what “green cheese” is, what “the moon” is, what “is made of” means, and so on. So the first piece of information is a signal of some sort that indicates the informing communication, and the second is a dictionary that defines the interpretation of the signaled communication. It is the dictionary that allows the signaled information to be placed into context within a framework of existing knowledge.

Paul Revere, in his famous ride, exemplified all of the basic principles with the “One if by land, two if by sea” dictionary. Implicit in this is “None if not coming.” This number of lamps shown—0, 1, or 2 in the Old North Church tower in Boston, indicating the direction of British advance—formed the dictionary for the communication system. The actual signal consisted of 0 or 1 or 2 lamps showing in the tower window.

11.2.2 Measuring Information: Signals

A *signal* is a system state that indicates a defined communication. A system can have any number of signals. The English language has many thousands—each word carrying, or signaling, a unique meaning. Paul Revere came close to using the least possible signal. The least possible signal is a system state that is either present or not present. Any light in the Old North Church signaled that the British were coming—no light, no British coming. This minimal amount of signaled information can be indicated by any two-state arrangement: on and off, 1 and 0, up and down, present and absent. It is from this two-state system of signal information that we get the now ubiquitous *bi*nary digi*t*, or *bit* of information. Modern computer systems are all built from many millions of two-state switches, each of which can represent this minimal signal.

Back to the Old North Church tower. How many bits did Paul Revere’s signal need? Well, there are three defined system states: 0 lamps = no sign of the British, 1 lamp = British coming by land, 2 lamps = British coming by sea. One bit can carry only two system states. State 0 = (say) no British coming, state 1 = (say) land advance. (Note that there is no necessary connection between the number of lamps showing and the number of bits.) There is no more room in one bit to define more than two system states. So in addition to one bit signaling two states—no advance or land advance—at least one more bit is needed to indicate a sea advance. With two bits, up to four system states can be encoded, as shown in Table 11.1.

TABLE 11.1 Only three system states are needed to carry Paul Revere's message (using two bits leaves one state undefined).

Bit 1 state	Bit 2 state	Tower lights	Meaning
0	0	None	No sign of the British
0	1	1	Land advance
1	0	2	Sea advance
1	1	Undefined	Undefined

But two bits seems to be too many as this communication system has only three states. There is an undefined system state when, in this example, both bits are in their "1" state. It looks like 1 1/2 bits is enough to carry the message—and indeed it is. Fractional bits may seem odd. It may seem that fractional bits can't exist, which is true. But the measurement here is only of how many bits are needed to signal the information, and for that about 1 1/2 bits will do the job.

When Paul Revere constructed his signaling system, he directly faced the problem that, in practice, two bits are needed. When the signals were devised, Paul used one lighted lantern to indicate the state of one bit. He needed only 1 1/2 lights, but what does it mean to show 1/2 a light? His solution introduced a redundant system state, as shown in Table 11.2

TABLE 11.2 Paul Revere's signaling system used redundancy in having two states carry the same message.

Bit 1 state	Bit 2 state	Tower lights	Meaning
0	0	None	No sign of the British
0	1	1	Land advance

1	0	1	Land advance
1	1	2	Sea advance

With this signaling system, land advance is indicated by two separate system states. Each state could have been used to carry a separate message, but instead of having an undefined system state, an identical meaning was assigned to multiple system states. Since the entire information content of the communication system could be carried by about 1/2 bits, there is roughly 1/2 a bit of redundancy in this system.

Redundancy measures duplicate information in system states. Most information-carrying systems have redundancy—the English language is estimated to be approximately 50% redundant. Tht is why yu cn undrstnd ths sntnce, evn thgh mst f th vwls are mssng! It is also what allows data set compression—squeezing out some of the redundancy.

There are many measures that can be used to measure information content, but the use of bits has gained wide currency and is one of the most common. It is also convenient because one bit carries the least possible amount of information. So the information content of a data set is conveniently measured as the number of bits of information it carries. But given a data set, how can we discover how many bits of information it does carry?

11.2.3 Measuring Information: Bits of Information

When starting out to measure the information content of a data set, what can be easily discovered within a data set is its number of system states—not (at least directly) the number of bits needed to carry the information. As an understandable example, however, imagine two data sets. The first, set A, is a two-bit data set. It comprises two variables each of which can take values of 0 or 1. The second data set, set B, comprises one one-bit variable, which can take on values of 0 or 1. If these two data sets are merged to form a joint data set, the resulting data set must carry three bits of information.

To see that this is so, consider that set A has four possible system states, as shown in Table 11.3. Set B, on the other hand, has two possible system states, as shown in Table 11.4.

TABLE 11.3 Data set A, using two bits, has four discrete states.

Set A variable 1	Set A variable 2	System state

0	0	1
0	1	2
1	0	3
1	1	4

TABLE 11.4 Data set B, using one bit, has two discrete states.

Set B variable 1	**System state**
0	1
1	2

Clearly, combining the two data sets must result in an information-carrying measurement for the combined data set of three bits of information total. However, usually the information content in bits is unknown; only the numbers of system states in each data set is known. But adding the two data sets requires multiplying the number of possible system states in the combined data set, as Table 11.5 shows.

TABLE 11.5 Combining data sets A and B results in a composite data set with 2 x 4 = 8 states, not 2 + 4 = 6 states.

Set A variable 1	**Set A variable 2**	**Set B variable 1**	**Set A system state**	**Set B system state**	**Composite system state**

0	0	0	1	1	1
0	0	1	1	2	2
0	1	0	2	1	3
0	1	1	2	2	4
1	0	0	3	1	5
1	0	1	3	2	6
1	1	0	4	1	7
1	1	1	4	2	8

There are *not* 4 + 2 = 6 combined states, but 4 x 2 = 8 combined states. Adding data sets requires multiplying the number of system states. However, although the number of system states is multiplicative, the number of bits of information is additive—two bits in data set A and one bit in data set B gives 2 + 1 = 3 bits of information. This relationship allows a very convenient measure of information that is based on the properties of logarithms.

Adding the logarithms of numbers is an easy way to multiply the numbers themselves. The logarithm of a number simply consists of whatever power some base number has to be raised to so as to yield the required number. The relationship is

$\text{base}^{\text{logarithm}} = \text{number}$

So, for instance, since 62 = 36, and 63 = 216, the logarithm of 36, to the base of 6, is 2, and the logarithm of 216, to the base of 6, is 3. But

36 x 216 = 7776.

Since $6^2 = 36$, and $6^3 = 216$, then substituting gives

62 x 63 = 7776

but

$6^{2 + 3} = 6^5 = 7776$.

So

$\log_6(36) + \log_6(216) = 36 \times 216 = 7776$.

So it is easy to see that adding the logarithms of numbers is equivalent to multiplying the numbers themselves.

When measuring information content, the bit has already been seen to be a useful unit of measurement. The bit represents the minimum amount of information possible. But a bit has only two states. When using logarithms for determining information content, because a bit has only two states, the logarithm base used is 2. The “number” in the formula “$\text{base}^{\text{logarithm}}$ = number” is the number of system states, so that the logarithm of the number of system states, to the base 2, is a measure of the information content of the data set in bits. So

$\log_2$(system states) = information content in bits

Using logarithms to the base of 2 gives the information bit measure needed to carry a specified number of system states. Returning to the example of data sets A and B, for data set A with four system states that number is $\log_2(4) = 2$ bits. For data set B with two system states the information content is $\log_2(2) = 1$ bit. So for four system states, 2 bits of information are needed, and for two system states, 1 bit is needed, which is exactly as it appears in the example. When adding the two data sets, the combined data set has eight system states, which requires $\log_2(8) = 3$ bits.

(Note, however, that since $\log_2(3) = 1.58$, Paul Revere’s system, while still needing fractional bits to carry the information, seems to need a little more than just 1 bits. More on this shortly.)

So far as this example goes, it is fine. But the example assumes that each of the system states counts equally, that is to say, each outcome is equally likely. Yet, unless a data set has a uniform distribution, the frequency with which each system state turns up is not uniform. To put it another way, unless equally distributed, not all system states are equally likely. Is this important?

11.2.4 Measuring Information: Surprise

As already discussed above, information content of a particular signal depends to some extent on how unexpected, or surprising, it is. In a fixed data set, the signals correspond to system states. Any specific data sample contains a fixed amount of information. However, the various signals, or system states, in the data set do not all carry an identical amount of information. The least surprising, or most common, signals carry less information than the most surprising, or least common, signals.

Surprise may seem like a subjective factor to measure. However, so far as signal information is concerned, surprise only measures the “unexpectedness” of a particular signal. The more likely a signal, the less surprising it is when it actually turns up. Contrarily, the less likely a signal, the more surprising it is. The degree of surprise, then, can be measured by how likely it is that a particular signal will occur. But in a data set, a signal is just a system state, and how likely it is that a particular system state occurs is measured by its joint probability, discussed in Chapter 10. Surprise can be quantified in terms of the relative frequency of a particular signal, or system state, in a representative sample. This is something that, if not always exactly easy to measure, can certainly be captured in principle. Surprisingness, then, turns out to be a measure of the probability of a particular system state in a representative sample. This, of course, is just its relative frequency in the data set.

As an example, suppose that you enter a lottery. It is a very small lottery and only 10 tickets are sold. As it happens, you bought 5 tickets. Clearly, in a fair drawing you stand a 50% chance of winning. After the drawing, you receive a signal (message) that you have won. Are you very surprised? How much information does the system contain, and how much is contained in the signal?

This system has two states as far as you are concerned, each 50% likely. Either you win or you lose. System information content: $\log_2(2) = 1$ bit of information. Each state, win or lose, has a probability of $1/2 = 0.5$, or 50%. To discover how much information each signal carries, we must find how many bits of information the individual signal carries, weighted (multiplied) by the chance of that particular signal occurring. For this lottery the chance of a win is 0.5 (or 50%) and the unweighted win information is $\log_2(0.5)$, so the weighted win information is

chance of winning x unweighted win information in bits of winning x $\log_2$(chance of winning)

50% x $\log_2(0.5)$

50% x 1 = 0.5 bits of information

This system carries half a bit of information in the win signal, and by similar reasoning, half a bit in the lose signal. But suppose you had only bought one ticket? Now the win state only occurs (for you) with a 10% probability. Obviously you have a 90% probability of losing. If you win under these circumstances, you should be more surprised than before since it is much less likely that you will win. If a win signal does arrive, it should carry more information for you since it is unexpected—thus more surprising. Also, the more highly expected lose signal carries little information, since that is the one you most expect anyway. But what does this do to the total amount of information in the system?

The total amount of information in this system is lower than when the outcomes were equally likely—adding the weighted information (shown in Table 11.6 by P $\log_2(P)$) for win signal (0.332) and lose signal (0.137) gives an information content for the system of 0.469 bits. Yet the win signal, if it occurs, carries more than 0.3 bits of information—over 70% of the information available in the system!

TABLE 11.6 Information content in the lottery system when the win/lose signals aren't equal.

	Win	Lose
Probability (P)	0.1	0.9
$\log_2(P)$	3.32	0.152
P $\log_2(P)$	0.332	0.137

11.2.5 Measuring Information: Entropy

The information measure that has been developed actually measures a feature of the information content of a data set called *entropy*. Much confusion has been provoked by the use of this term, since it is identical to one used in physics to describe the capability of a system to perform work. Increased confusion is caused by the fact that the mathematical underpinnings of both types of entropy appear almost identical, and much discussion has ensued about similarities and differences between them. Data miners can leave this confusion aside and focus on the fact that *entropy measures uncertainty in a system*. When, for instance, the lottery had a 50/50 outcome, both the information measure and the uncertainty of the outcome were at their greatest for the system (1). It was impossible to say that any one outcome was more likely than any other—thus maximum uncertainty. This is a state of *maximum entropy*. As the system states move away from maximum uncertainty, so too does entropy decrease. Low entropy, as compared to the theoretical maximum for a system, suggests that the data set is possibly biased and that some system states may not be well represented. This measure of entropy is an important tool in general, and one that is particularly important in parts of the data survey.

11.2.6 Measuring Information: Dictionaries

Thus far in measuring the information content of a data set, looking at the data set alone seems sufficient. However, recall that communicated information comes in two parts—the signal and the dictionary. The dictionary places the signal in its context within a knowledge structure. It's a fine thing to know that you have won the lottery, but winning $1 is different from winning $1,000,000—even if the signal and its chance of occurring are identical. Sometimes one bit of information can carry an enormous amount of dictionary information. (When I say "go," play Beethoven's Fifth Symphony!) Creating knowledge structures is not yet within the realm of data mining (but see Chapter 12).

Measuring total information content requires evaluating the significance of the signal in context—and that is beyond the scope of the data survey! However, it is important to the miner since the data survey measures and uses the information content of the data set in many different ways. It still remains entirely up to the miner and domain experts to decide on significance, and that requires access to the system dictionaries. That may range (typically) from difficult to impossible. What this means for the miner is that while there are some objective measures that can be applied to the data set, and while valid measurements about the signal information content of the data set can be made, no blind reliance can be placed on these measures. As ever, they require care and interpretation.

To see the effect of changing even a simple dictionary, return to the example of the 10-ticket lottery. Previously we examined it from your point of view as a potential winner. We assumed that your only concern was for your personal win outcome. But suppose that you knew the other entrants and cared about the outcomes for them. This changes your dictionary, that is, the context in which you place the messages. Suppose that, as in the original case, you bought five of the 10 tickets. You still have a 50% chance of winning. But now five of your family members (say) each bought one of the other five tickets: Anne, Bill, Catherine, David, and Liz. Obviously they each have a 10% chance of winning. Does this change the information content of the system *as far as you are concerne*d?

There are now six outcomes of concern to you, as shown in Table 11.7. Well, the information in the "you win" signal hasn't changed, but the information content of the other five signals now totals 1.66. When previously you didn't care who else won, the signals totaled 0.5. As expected, when you care about the other signals, and place them in a more meaningful framework of reference, their information content changes—for you.

TABLE 11.7 Information content in the signals when outcome meaning changes.

Winner	**You**	**Anne**	**Bill**	**Catherine**	**David**	**Liz**
Probability (P)	0.5	0.1	0.1	0.1	0.1	0.1

$\log_2(P)$	1.0	3.32	3.32	3.32	3.32	3.32
$P \log_2(P)$	0.5	0.332	0.332	0.332	0.332	0.332

You might observe that in the previous example, there were in any case five other outcomes. It was just that the example lumped them all together for convenience. Suddenly, saying that the extra outcomes have meaning is “cheating.” What changed, however, is your knowledge of what the outcomes mean. It is this change in your state of knowledge of the signals’ meaning—in your dictionary—that is the crux of the matter. The knowledge you bring can materially affect the information content of the system. Outcomes and probabilities are unaffected, but information content may be dramatically affected.

As already pointed out, access to the dictionary may be impossible. So far as surveying data is concerned, it means that signal information enfolded into a data set can be assessed, but it may—probably will—be impossible to fully evaluate the information available. Fortunately, using entropic and other information measures on the signal information can be immensely useful. It is always important for the miner to keep in mind that, as powerful a tool as information analysis is, there is no point-and-shoot method of definitively capturing the information enfolded into a data set. The dictionary, or at least large parts of it, will almost always be unavailable and its content unknown.

However, what this example does show clearly is that knowledge of meaningful outcomes can materially change the information carried in a data set. It is well worth the miner’s time and effort to try to establish what constitutes meaningful outcomes for a particular project.

For instance, consider the by now almost classic example of Barbie dolls and chocolate bars. Allegedly, a major retailer discovered in a large data set that Barbie dolls and chocolate bars sell together. Assume, for the sake of the example, that this is indeed true. What can be done with the information? What, in other words, does it mean? The answer is, Not much! Consider the possibilities. Does the information indicate that Barbie dolls made out of chocolate would sell particularly well? Does it mean that Barbie dolls should be placed on the display shelf next to chocolate bars? Does it mean that a chocolate bar should be packaged with a Barbie doll? And what implications do these possibilities have for targeting the promotion? What, in short, can usefully be done with this signal? The signal carries essentially meaningless information. There is no way to act on the insight, to actually use the information communicated in context. The signal carries measurable information, but the dictionary translates it as not meaningful. The difference between system states and meaningful system states lies in the dictionary.

11.3 Mapping Using Entropy

Entropic measurements are the foundation for evaluating and comparing information content in various aspects of a data set. Recall that entropy measures levels of certainty and uncertainty. Every data set has some theoretical maximum entropy when it is in a state of maximum uncertainty. That is when all of the meaningful outcomes are equally likely. The survey uses this measure to examine several aspects of the data set.

As an example of what entropy measurements can tell the miner about a data set, the following discussion uses two data sets similar to those first described in Chapter 6 as an example. Chapter 6 discussed, in part, relationships that were, and those that were not, describable by a function. Two data sets were illustrated (Figures 6.1 and 6.2). The graphs in these two figures show similar manifolds except that, due to their orientation, one can be described by a function and the other cannot. Almost all real-world data sets are noisy, so the example data sets used here comprise noisy data approximating the two curves. Data set F (functional) contains the data for the curve that can be described by a function, and data set NF (nonfunctional) contains data for the curve not describable by a function.

While simplified for the example, both of these data sets contain all the elements of real-world data sets and illustrate what entropy can tell the miner. The example data is both simplified and small. In practice, the mined data sets are too complex or too large to image graphically, but the entropic measures yield the same useful information for large data sets as for small ones. Given a large data set, the miner must, in part, detect any areas in the data that have problems similar to those in data set NF—and, if possible, determine the scale and limits of the problem area. There are many modeling techniques for building robust models of nonfunctionally describable relationships—but only if the miner knows to use them! Very briefly, the essential problem presented in data set NF is that of the "one-to-many problem" in which one value (signal or system state) of the input variable (*x*) is associated validly with several, or many, output (*y*) values (signals or system states) that are not caused by noise.

Figure 11.1 illustrates data set F and shows the predicted values of a function fitted to the noisy data. The function fits the data quite well. Figure 11.2 shows data set NF. This figure shows three things. First, the squares plot the noisy data. Second, the continuous line shows the modeled function, which fits the data very badly. Third, the circles show the underlying *x*-*y* relationship.

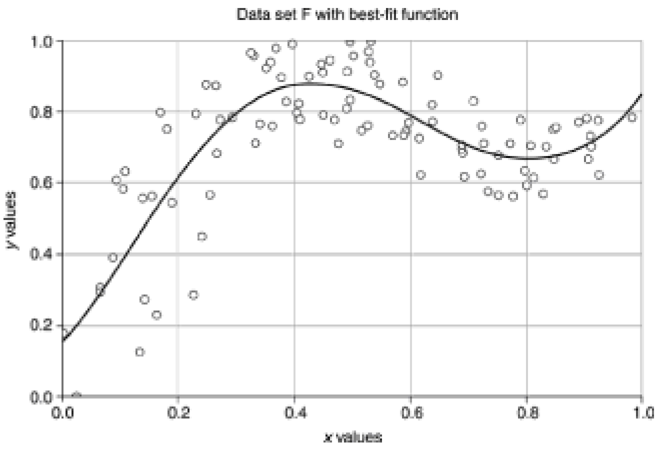


Figure 11.1 A noisy data set for which the underlying relationship can be described by a function. The *x* values are input and the *y* values are to be predicted. The continuous line shows a modeled estimate of the underlying function.

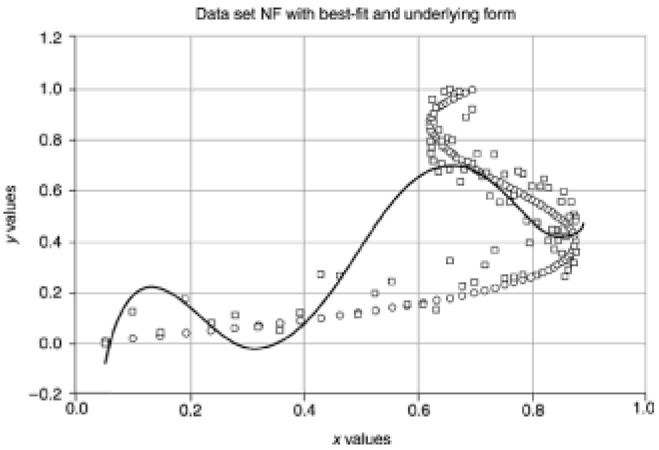


Figure 11.2 This illustrates data set NF. The squares plot the data set, and the continuous line shows the values of a modeled function to fit the data, which it does very poorly. The circles plot the actual underlying *x*-*y* relationship.

The following discussion uses six different information measures. Since the actual values vary from data set to data set, the values are normalized across the range 0–1 for convenience. The actual values for this data set, along with their normalized values, are shown in Table 11.8.

TABLE 11.8 Information measure values for the example data set NF.

Measure	Actual	Norm
Max entropy	3.459	
Entropy X	3.347	0.968
Entropy Y	3.044	0.880
Entropy (Y|X)	1.975	0.649
Entropy (X|Y)	2.278	0.681
Mutual info (X;Y)	1.069	0.351

The Supplemental Material section at the end of this chapter further explains the meaning and interpretation of these measures and their values. And because information measures are very important to the data survey, you'll also find information on deriving these measures from the data in the same section.

11.3.1 Whole Data Set Entropy

An ideal data set for modeling has an entropy that is close to the theoretical maximum. Since the modeling data set should represent all of its meaningful states, entropy will be high in such a data set. For modeling, the higher, the better. But a single data set for modeling usually comes in two pieces—the "input" and the "output." The input piece comprises all of the variables that, statistically, are described as the independent variables. These are the variables that contain the "evidence" or information that is to be mapped. The output variables contain the information to be predicted, or for which, or about which, inferences will be extracted from the input variables. Although usually collected as a single entity, both input and output data subsets should be independently examined for their entropy levels.

In the example, the input is a single variable, as is the output. Since both input and output are single variables, and have the same possible number of signals—that is, they cover the same range of 0–1—the maximum possible entropy for both variables *x* and *y* is identical. In this case, that maximum possible entropy for both variables is 3.459. This is the entropy measure if the variables' signals were completely evenly balanced.

The actual measured entropy for each variable is close to the maximum, which is a desirable characteristic for modeling. It also means that not much can be estimated about the values of either *x* or *y* before modeling begins.

11.3.2 Conditional Entropy between Inputs and Outputs

Conditional probabilities are in the form “What is the probability of B occurring, given that A has already happened?” For example, it is clear that the answer to “What is the chance of rain tomorrow?” may be very different from the answer to the question “What is the chance of rain tomorrow, given that it is pouring with rain today?” Having a knowledge of existing evidence allows a (usually) more accurate estimate of the conditional probability.

Conditional entropy is very similar. It results in a measure of mutual information. The amount of *mutual information* is given by the answer to the question “How much information is gained about the output by knowing the input?” In other words, given the best estimate of the output before any input is known, how much better is the estimated output value when a specific input *is* known? Ideally, one specific output signal would have a conditional probability of 1, given a particular input. All other outputs would have a probability of 0. That means that as a prediction, any input signal (value) implies one specific output signal (value) with complete certainty. Usually the probabilities are not so tightly focused and are spread across several outputs. Mutual information measures exactly how much information is available to determine a specific output value, given an input value.

Every different input/output value combination may have a different mutual information value. Taking the weighted sum value of all the individual mutual information values results in a general idea of how well the inputs (*x* values) predict the outputs (*y* values). The symbol used for this measure is “Entropy(Y|X)” which is read as “The entropy of the Y variable given the values of the X variable.” In the example the Entropy(Y|X) is 1.975. This means that the entropy reduces from its starting state of 3.347 to 1.975, on average, when the *x* value is specified. The level of uncertainty is reduced to 1.975/3.347 = about 59% of what it was, or by about 41%.

This 41% reduction in uncertainty is very important. Contrast it with the reduction in uncertainty in the value of *x* when *y* is known. Entropy(X|Y)/ Entropy(Y) = 75%, a reduction of only 25% in the uncertainty of *x* given the *y* value. This says

knowing the value of *x*

when predicting the value of *y*

reduces the amount of uncertainty by 41%

and

knowing the value of *y*

when predicting the value of *x*

reduces the amount of uncertainty by 25%.

It is clear that *x* predicts *y* much better than *y* predicts *x*, so these values are clearly not symmetrical. This is exactly as expected given that in the example F data set, a single value of *x* is associated with a single value of *y* plus noise. However, the reverse is not true. A single value of *y* is associated with more than one value of *x*, plus noise. The entropy measure points out this area. It also measures the amount of noise remaining in the system that is not reduced by knowing an *x* value. This is a measure of ambient noise in the system that places theoretical limits on how good any model can be.

These measures are very important to any miner. They point to potential problem areas and indicate how well any model will at best perform—before any model is attempted. They measure limits set by the nature of the data.

As well as looking at the overall mutual information, it can be worth looking at the individual signal entropy plots for the input and output data. Figure 11.3 shows the entropy plots for the range of the input variable *x* predicting variable *y* in both data sets. The upper image shows the entropy for data set F, which is functionally describable. The plot shows the level of uncertainty in the *y* value, given the *x* value, for all given *x* values. This is important information for the miner. It shows that, in general, low values of *x* have more uncertainty than higher values, although overall the entropy is fairly low. Probably a reasonably good model can be constructed so long as the accuracy required of the model does not exceed the noise in the system. (It is quite possible to specify a required minimum model accuracy, and work back to the maximum allowable noise level that the required accuracy implies, although a description of how to do this is beyond the scope of this brief introduction to the data survey.)

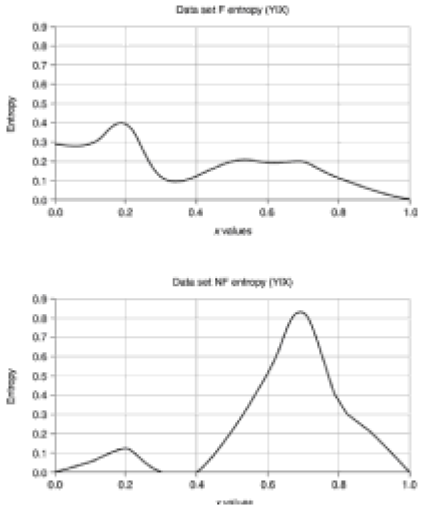


Figure 11.3 Entropy for the range of input values of *x* in the F data set (top) and in the NF data set (bottom).

A glance at the lower image tells a very different story. In the NF data set, values between about 0.6 and 0.8 are very uncertain, not well predicted at all. A look at the data shows that this is exactly the range of values where the *y* value has multiple values for a single *x* value. This may seem obvious in a small demonstration data set, but can be difficult or impossible to find in a real-world data set—unless found using mutual information.

11.3.3 Mutual Information

In an ideal data set, input and output signals all occur with equal probability—maximum entropy. In practice, such a state is very unlikely. Most of the signals will occur with some bias such that some are more likely to occur than others. Because of this bias, before modeling starts, the system is already known to have preferred states (carrying less information) and nonpreferred states (carrying more information). The purpose of any model is to use the input information to indicate the appropriate output with as high a degree of confidence as possible—or with a very low uncertainty. But the amount of uncertainty about an output depends entirely on how much information the input carries about the output. If this information can be measured, the miner can estimate the overall accuracy of any model prediction of that output—without actually building the model. In other words, the miner can estimate if any worthwhile model of specific outputs can, even theoretically, be built. Mutual information provides this estimate.

Mutual information is always balanced. The information contained in *x* and *y* is the same as that contained in *y* and *x*. For the data set F, the mutual information measure is 1.069, or has a ratio of about 0.35. The higher this number, the closer the ratio is to 1, the more information is available, so the better the model. In this case, there is a fair amount of noise in the data preventing an accurate model, as well as possible distortions that may be correctable. As it stands, the fairly low degree of information limits the maximum possible accuracy of a model based on this data set. Once again, discovering the actual limits is fairly straightforward, but beyond the scope of this introductory tour.

11.3.4 Other Survey Uses for Entropy and Information

In general, the miner greatly benefits from the power of entropic analysis when surveying data. The applications discussed so far reveal a tremendous amount about any prepared data set. Information theory can be used to look at data sets in several ways, all of which reveal useful information. Also, information theory can be used to examine data as instances, data as variables, the data set as a whole, and various parts of a data set. Entropy and mutual information are used to evaluate data in many ways. Some of these include

To evaluate the quality and problem areas of the input data set as a whole

- To evaluate the quality and problem areas of the output data set as a whole
- To evaluate the quality and problem areas of other data sets (e.g., test and verification) as a whole
- To evaluate the quality of individual variables over their range of values
- To estimate the independence of variables (measuring the entropy between inputs) both on average and in different parts of their range
- To select the input variables most independent from each other that carry maximum predictive or inferential information about the output(s)
- To estimate the maximum possible quality of a model over its range of inputs and outputs
- To identify problem areas, problem signals, and poorly defined areas of a model

11.3.5 Looking for Information

Making a comprehensive calculation of all possible variable and signal combinations is almost always impossible. The number of combinations is usually too high as the combinatorial explosion defeats a comprehensive search. In practice, data surveying searches some portion of the possibilities at a high level, looking for potential problems. This is a form of what is known as *attention processin*g—taking a high-level look at an area, and only looking more closely at any potentially interesting or difficult areas.

Attention processing simply describes a familiar method of searching for information or problems. Readers, hunting for information in, say, an encyclopedia, usually start with a high-level overview such as the index. At another level, they may skim articles for features of interest. The actual interesting features may well be scrutinized in detail. This is attention processing—more attention paid to areas in proportion to their degree of interest.

Detecting addresses on envelopes provides a practical computational example. Figure 11.4 illustrates the process. The first part of the problem is to determine a ZIP code. Rather than building a model to scrutinize the whole surface of an envelope, which would be very difficult, models with various degrees of attention are used. A top-level model determines if there is an address at all. (If the envelope is face down, for instance, there is no address.) When an address is likely to be present, another model determines if it is right side up. When right side up, the next model identifies a likely location for the ZIP code. The next model focuses only on the location of the ZIP code to extract the actual digits. Finally, the extracted digits are individually identified. Of course, then the ZIP code itself has to be identified, and so on. By using attention processing, a large task was divided into smaller tasks, each of which searched for a particular feature, allowing the next feature to receive more detailed attention.

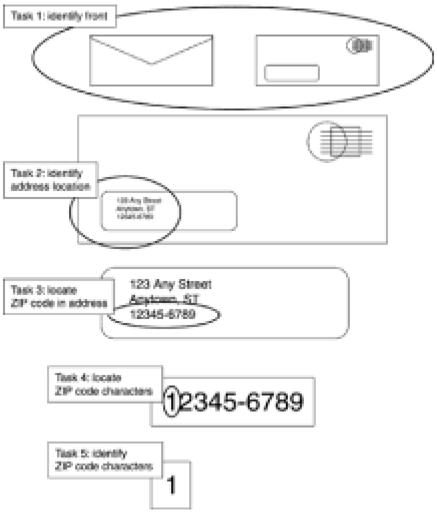

Task 1: identify front
Task 2: identify address location
Task 3: locate ZIP code in address
123 Any Street
Anytown, ST
12345-6789
Task 4: locate ZIP code characters
12345-6789
Task 5: identify ZIP code characters
1

Figure 11.4 Attention processing separates a task into parts and attends to separate parts of the task rather than trying to perform the whole task in one step.

So it is with the data survey. The survey starts with the general entropic calculations. This results in maps of particularly troublesome input/output signal interactions and establishes a level of expectation for the accuracy of the model. From here, the miner can explore either more broadly, or more deeply, into various problem areas, or areas that seem particularly promising.

The information analysis part of the survey provides a very good idea of the overall quality of the data set, and also identifies potential problem areas. However, while it identifies *where* the problems are, it says little about *what* they are.

11.4 Identifying Problems with a Data Survey

There is fundamentally one reason and three problems that can reduce or prevent mining tools from identifying a good relationship between input and output data sets. They are

Reason: The data set simply does not enfold sufficient information to define the relationship between input and output with the accuracy required.

- Problem 1: The relationship between input and output is very complex.

- Problem 2: Part(s) of the input/output relationship are not well defined by the available data.

- Problem 3: High variance or noise obscures the underlying relationship between input and output.

Turning first to the reason: The data set simply does not contain sufficient information to define the relationship to the accuracy required. This is not essentially a problem with the data sets, input and output. It may be a problem for the miner, but if sufficient data exists to form a multivariably representative sample, there is nothing that can be done to “fix” such data. If the data on hand simply does not define the relationship as needed, the only possible answer is to get other data that does. A miner always needs to keep clearly in mind that the solution to a problem lies in the problem domain, not in the data. In other words, a business may need more profit, more customers, less overhead, or some other business solution. The business does not need a better model, except as a means to an end. There is no reason to think that the answer has to be wrung from the data at hand. If the answer isn’t there, look elsewhere. The survey helps the miner produce the best possible model from the data that is on hand, and to know how good a model is possible from that data before modeling starts.

But perhaps there are problems with the data itself. Possible problems mainly stem from three sources: one, the relationship between input and output is very complex; two, data describing some part of the range of the relationship is sparse; three, variance is very high, leading to poor definition of the manifold. The information analytic part of the survey will point to parts of the multivariable manifold, to variables and/or subranges of variables where entropy (uncertainty) is high, but does not identify the exact problem in that area.

Remedying and alleviating the three basic problems has been thoroughly discussed throughout the previous chapters. For example, if sparsity of some particular system state is a problem, Chapter 10, in part, discusses ways of multiplying or enhancing particular features of a data set. But unless the miner knows that some particular area of the data set has a problem, and that the problem is sparsity, it is impossible to fix. So in addition to indicating overall information content and possible problem areas, the survey needs to suggest the nature of the problem, if possible.

The survey looks to identify problems within a specific framework of assumptions. It assumes that the miner has a multivariably representative sample of the population, to some acceptable level of confidence. It also assumes that in general the information content of the input data set is sufficient to adequately define the output. If this is not the case, get better data. The survey looks for local problem areas within a data set that overall meet the miners needs. The survey, as just described, measures the general information content of the data set, but it is specific, identified problems that the survey assesses for the possible causes. Nonetheless, in spite of these assumptions, the survey estimates the confidence level that the miner has sufficient data.

11.4.1 Confidence and Sufficient Data

A data set may be inadequate for mining purposes simply because it does not truly represent the population. If a data set doesn't represent the population from which it is drawn, no amount of other checking, surveying, and measuring will produce a valid model. Even if entropic analysis indicated that it is possible to produce a valid, robust model, that is still a mistake. Entropy measures what is present, and if what is present is not truly representative, the entropic measures cannot be relied upon either. The whole foundation of mining rests on an adequate data set. But what constitutes an adequate data set?

Chapter 5 addressed the issue of capturing a representative sample of a variable, while Chapter 10 extended the discussion to the multivariable distribution and capturing a multivariably representative sample. Of course, any data set can only be captured to some degree of confidence selected by the miner. But the miner may face the problem in two guises, both of which are addressed by the survey.

First, the miner may have a particular data set of a fixed size. The question then is, "Just how multivariably representative is this data set?" The answer determines the reliability of any model made, or inferences drawn, from the data set. Regardless of the entropic measurements, or how apparently robust the model built, if the sample data set has a very low confidence of being representative, so too must the model extracted, or inferences drawn, have a low confidence of being representative. The whole issue hinges on the fact that if the sample does not represent the population, nothing drawn from such a sample can be considered representative either.

The second situation arises when plenty of data is available, perhaps far more than can possibly be mined. The question then is, "How much data captures the multivariable variability of the population?" The data survey looks at any existing sample of data, estimates its probability of capturing the multivariable variability, and also estimates how much more data is required to capture some specified level of confidence. This seems straightforward enough. With plenty of data available, get a big enough sample to meet some degree of confidence, whatever that turns out to be, and build models. But, strange as it may seem, and for all the insistence that a representative sample is completely essential, a full multivariable representative sample may not be needed!

It is not that the sample need not be representative, but that perhaps all of the variables may not be needed. Adding variables to a data set may enormously expand the number of instances needed to capture the multivariable variability. This is particularly true if the added variable is not correlated with existing variables. It is absolutely true that to capture a representative sample with the additional variable, the miner needs the very large number of instances. But what if the additional variable is not correlated (contains little information about) the predictions or relationships of interest? If the variable carries little information of use or interest, then the size of the sample to be mined was expanded for little or no useful gain in information. So here is another very good reason for removing variables that are not of value.

Chapter 10 described a variable reduction method that is implemented in the demonstration software. It works and is reasonably fast, particularly when the miner has not specifically segregated the input and output data sets. Information theory allows a different approach to removing variables. It requires identifying the input and output data sets, but that is needed to complete the survey anyway. The miner selects the single input variable that carries most of the information about the output data set. Then the miner selects the variable carrying the next most information about the output, such that it also carries the least information in common (mutual information content) with the previously selected variable(s). This selection continues until the information content of the derived input data set sufficiently defines the model with the needed confidence. Automating this selection is possible. Whatever variable is chosen first, or whichever variables have already been chosen, can enormously affect the order in which the following variables are chosen. Variable order can be very sensitive to initial choice, and any domain knowledge contributed by the miner (or domain expert) should be used where possible.

If the miner adopts such a data reduction system, it is important to choose carefully the variables intended for removal. It may be that a particular variable carries, in general, little information about the output signals, but for some particular subrange it might be critically important. The data survey maps all of the individual variables' entropy, and these entropy maps need to be considered before making any final discard decision.

However, note that this data reduction activity is not properly part of the data survey. The survey only looks at and measures the data set presented. While it provides information about the data set, it does not manipulate the data in any way, exactly as a map makes no changes to the territory, but simply represents the relationship of the features surveyed for the map. When looking at multivariate distribution, the survey presents only two pieces of information: the estimated confidence that the multivariable variability is captured, and, if required, an estimate of how many instances are needed to capture some other selected level of confidence. The miner may thus learn, say, that the input data set captured the multivariable variability of the population with a 95% confidence level, and that an estimated 100,000 more records are needed to capture the multivariable variability to a 98% confidence level.

11.4.2 Detecting Sparsity

Overall, of course, the data points in state space (Chapter 6) vary in density from place to place. This is not necessarily any problem in itself. Indeed, it is a positive necessity as this variation in density carries much of the information in the data set! A problem only arises if the sparsity of data points in some local area falls to such a level that it no longer carries sufficient information to define the relationship to the required degree. Since each area of state space represents a particular system state, this means only that some system states are insufficiently represented.

This is the same problem discussed in several places in this book. For instance, the last chapter described a direct-mail effort’s very low response rate, which meant that a naturally representative sample had relatively very few samples of responders. The number of responses had to be artificially augmented—thus populating that particular area of state space more fully.

However, possibly there is a different problem here too. Entropy measures, in part, how well some particular input state (signal or value) defines another particular output state. If the number of states is low, entropy too may be low, since the number of states to choose from is small and there is little uncertainty about which state to choose. But the number of states to choose from may be low simply because the sample populates state space sparsely in that area. So low entropy in a sparsely populated part of the output data set may be a warning sign in itself! This may well be indicated by the forward and reverse entropy measures (Entropy(X|Y) and Entropy(Y|X)), which, you will recall, are not necessarily the same. When different in the forward and reverse directions, it may indicate the “one-to-many problem,” which could be caused by a sparsely populated area in one data set pointing to a more densely populated area in the other data set.

The survey makes a comprehensive map of state space density—both of the input data set and the output data set. This map presents generally useful information to the miner, some of which is covered later in this chapter in the discussion of clustering. Comparing density and entropy in problematic parts of state space points to possible problems if the map shows that the areas are sparse relative to the mean density.

11.4.3 Manifold Definition

Imagine the manifold as a state space representation of the underlying structure of the data, less the noise. Remember that this is an imaginary construct since, among other ideas, it supposes that there is some “underlying mechanism” responsible for producing the structure. This is a sort of causal explanation that may or may not hold up in the real world. For the purposes of the data survey, the manifold represents the configuration of estimated values that a good model would produce. In other words, the best model should fill state space with its estimated values exactly on the manifold. What is left over—the difference between the manifold and the actual data points—is referred to as *error* or *noise*. But the character of this noise can vary from place to place on the manifold, and may even leave the “correct” position of the manifold in doubt. (And go back to the discussion in Chapter 2 about how the states map to the world to realize that any idea of a “correct” position of a manifold is almost certainly a convenient fiction.) All of these factors add up to some level of uncertainty in the prediction from place to place across the manifold, and it is this uncertainty that, in part, entropy measures. However, while measuring uncertainty, entropy does not actually characterize the exact nature of the uncertainty, for which there are several possible causes. This section considers problems with variance. Although this is a very large topic, and a comprehensive discussion is far beyond the scope of this section, a brief introduction to some of the main issues is very helpful in understanding limits to a model's applicability.

Much has been written elsewhere about analyzing variability. Recall that the purpose of the data survey is not to analyze problems. The data survey only points to possible problem areas, delivered by an automated sweep of the data set that quickly delivers clues to possible problems for a miner to investigate and analyze more fully if needed. In this vein, the manifold survey is intended to be quick rather than thorough, providing clues to where the miner might usefully focus attention.

Skewness

Variance was previously considered in producing the distribution of variables (Chapter 5) or in the multivariable distribution of the data set as a whole (Chapter 10). In this case, the data survey examines the variance of the data points in state space as they surround the manifold. In a totally noise-free state space, the data points are all located exactly on (or in) the manifold. Such perfect correspondence is almost unheard of in practice, and the data points hover around the manifold like a swarm of bees. All of the points in state space affect the shape of every part of the manifold, but the effect of any particular data point diminishes with distance. This is analogous to the gravity of Pluto—a remote and small body in the solar system—that does have an effect on the Earth, but as it is so far away, it is almost unnoticeable. The Moon, on the other hand, although not a particularly massive body as solar system bodies go, is so close that it has an enormous effect (on the tides, for instance).

Figure 11.5 shows a very simplified state space with 10 data points. The data points form two columns, and the straight line represents a manifold to fit these points. Although the two columns cover the same range of values, it's easy to see that the left column's values cluster around the lower values, while the right column has its values clustered around the higher values. The manifold fits the data in a way that is sensitive to the clustering, as is entirely to be expected. But the nature of the clustering has a different pattern in different parts of the state space. Knowing that this pattern exists, and that it varies, can be of great interest to a miner, particularly where entropy indicates possible problems. It is often the case that by knowing patterns exist, the miner can use them, since pattern implies some sort of order.

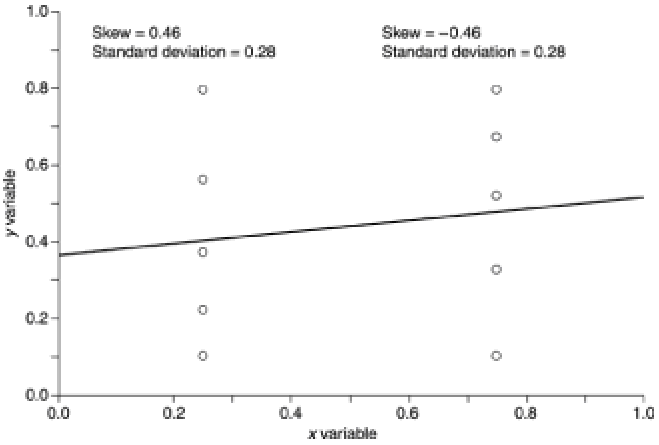


Figure 11.5 A simplified state space with 10 data points.

The survey looks at the local data affecting the position of the manifold and maps the data distribution around the manifold. The survey reports the standard deviation (see Chapter 5 for a description of this concept) and skew of the data points around the manifold. *Skewness* measures exactly what the term seems to imply—the degree of asymmetry, or lopsidedness, of a distribution about its mean. In this example the number is the same, but the sign is different. Zero skewness indicates an evenly balanced distribution. Positive skew indicates that the distribution is lighter in its values on the positive side of the mean. Negative skew indicates that the distribution is lighter in the more negative values of its range. Although not shown in the figure, the survey also measures how close the distribution is to being multivariably normal.

Why choose these measures? Recall that although the individual variables have been redistributed, the multivariable data points have not. The data set can suffer from outliers, clusters, and so on. All of the problems already mentioned for individual variable distributions are possible in multivariable data distributions too. Multivariable redistribution is not possible since doing so removes all of the information embedded in the data. (If the data is completely homogenous, there is no density variation—no way to decide how to fit a manifold—since regardless of how the manifold is fitted to the data, the uniform density of state space would make any one place and orientation as good as any other.) These particular measures give a good clue to the fact that, in some particular area, the data has an odd pattern.

Manifold Thickness

So far, the description of the manifold has not addressed any implications of its thickness. In two or three dimensions, the manifold is an imaginary line or a sheet, neither of which have any thickness. Indeed, for any particular data set there is always some specific best way to fit a manifold to that data. There are various ways of defining how to make the manifold fit the data—or, in other words, what actually constitutes a best fit. But it always results in some particular way of fitting the manifold to the data.

However, in spite of the fact that there is always a best fit, that does not mean that the manifold always represents the data over all parts of state space equally well. A glance at Figure 11.6 shows the problem. The manifold itself is not actually shown in this illustration, but the mean value of the *x* variable across the whole range of the *y* variable is 0.5. This is where the manifold would be fitted to this data by many best-fit metrics. What the illustration does show are the data points and envelopes estimating the maximum and minimum values across the *y* dimension. It is clear that where the envelope is widely spaced, the values of *x* are much less certain than where the envelope is narrower. The variability of *x* changes across the range of *y*. Assuming that this distribution represents the population, uncertainty here is not caused by a lack of data, but by an increase in variability. It is true that in this illustration density has fallen in the balloon part of the envelope. However, even if more data were added over the appropriate range of *y*, variability of *x* would still be high, so this is not a problem of lack of data in terms of *x* and *y*.

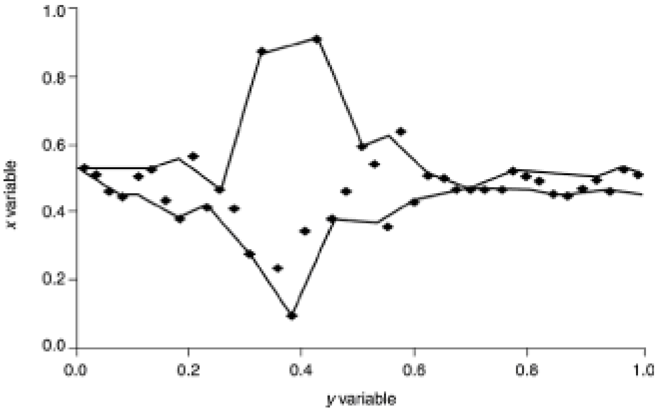


Figure 11.6 State space with a nonuniform variance. This envelope represents uncertainty due to local variance changes across the manifold.

Of course, adding data in the form of another variable might help the situation, but in terms of *x* and *y* the manifold’s position is hard to determine. This increase in the variability leaves the exact position of the manifold in the “balloon” area uncertain and ill defined. More data still leaves predicting values in this area uncertain as the uncertainty is inherent in the data—not caused by, say, lack of data. Figure 11.7 illustrates the variability of *x* across *y*.

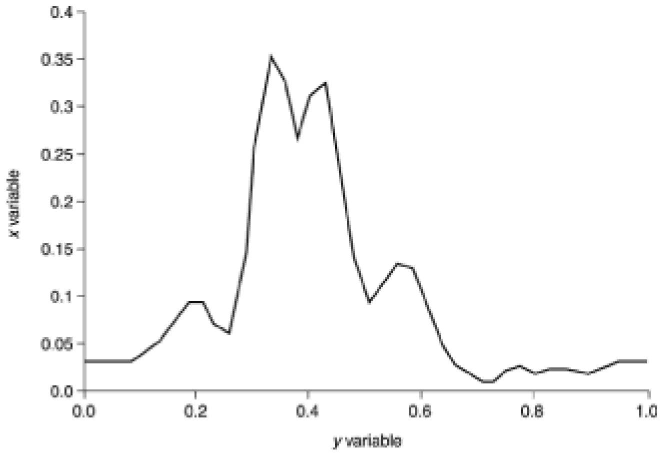


Figure 11.7 The variability in *x* is shown across the range of the variable *y*. Where variability is high, the manifold's position and shape are less certain.

The caveat with these illustrations is that in multidimensional state space, the situation is much more complex indeed than can be illustrated in two dimensions. It may be, and in practice it usually is, that some restricted part of state space has particular problems. In any case, recall that the individual variable values have been carefully redistributed and normalized, so that state space is filled in a very different way than illustrated in these examples. It is this difficulty in visualizing problem areas that, in part, makes the data survey so useful. A computer has no difficulty in making the multidimensional survey and pointing to problem areas. The computer can easily, if sometimes seemingly slowly, perform the enormous number of calculations required to identify which variables, and over which parts of their ranges, potential problems lurk. "Eyeballing" the data would be more effective at detecting the problems—if it were possible to look at all of the possible combinations. Humans are the most formidable pattern detectors known. However, for just one large data set, eyeballing all of the combinations might take longer than a long lifetime. It's certainly quicker, if not as thorough, to let the computer crunch the numbers to make the survey.

Very Complex Relationships

Relationships between input and output can be complex in a number of different ways. Recall that the relationship described here is represented by a manifold. The required values that the model will ideally predict fall exactly on the manifold. This means that describing the shape of the manifold necessarily has implications for a predictive model that has to re-create the shape of the manifold later. So, for the sake of discussion, it is easy to consider the problem as being with the shape of the manifold. This is simpler for descriptive purposes than looking at the underlying model. In fact, the problem is for the model to capture the shape of the manifold.

Where the manifold has sharp creases, or where it changes direction abruptly, many modeling tools have great difficulty in accurately following the change in contour. There are a number of reasons for this, but essentially, abrupt change is difficult to follow. This phenomenon is encountered even in everyday life—when things are changing rapidly, and going off in a different direction, it is hard to follow along, let alone predict what is going to happen next! Modeling tools suffer from exactly this problem too.

The problem is easy to show—dealing with it is somewhat harder! Figure 11.8 shows a manifold that is noise free and well defined, together with one modeling tool's estimate of the manifold shape. It is easy to see that the "point" at the top of the manifold is not well modeled at all. The modeled function simply smoothes the point into a rounded hump. As it happens, the "sides" of the manifold are slightly concave too—that is, they are curves bending in toward the center. Because of this concavity, which is in the opposite direction to the flexure of the point, the modeled manifold misses the actual manifold too. Learning this function requires a more complex model than might be first imagined.

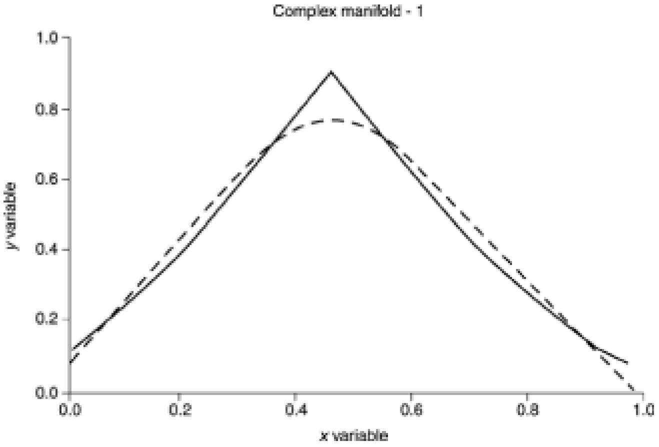


Figure 11.8 The solid arch defines the data points of the actual manifold and the dotted line represents one model's best attempt to represent the actual manifold.

However, the relative complexity of the manifold in Figure 11.9 is far higher. This manifold has two "points" and a sudden transition in the middle of an otherwise fairly sedate curve. The modeled estimate does a very poor job indeed. It is the "points" and sudden transitions that make for complexity. If the discontinuity is important to the model, and it is likely to be, this mining technique needs considerable augmentation to better capture the actual shape of the relationship.

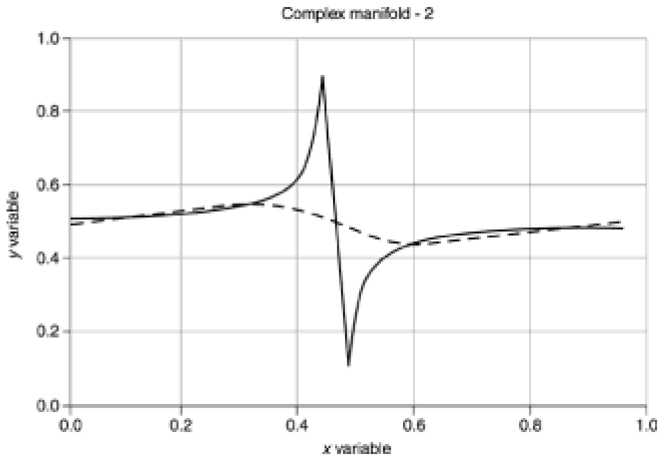


Figure 11.9 This manifold is fairly smooth except around the middle. The model (dotted line) entirely misses the sharp discontinuity in the center of the manifold—even though the manifold is completely noise-free and well-defined.

Curves such as this are more common than first glance might suggest. The curve in Figure 11.9, for instance, could represent the value of a box of seats during baseball season. For much of the season, the value of the box increases as the team keeps winning. Immediately before the World Series, the value rises sharply indeed since this is the most desirable time to have a seat. The value peaks at the beginning of the last game of the series. It then drops precipitously until, when the game is over, the value is low—but starts to rise again at the start of a new season. There are many such similar phenomena in many areas. But accurately modeling such transitions is difficult.

There is plenty of information in these examples, and the manifolds for the examples are perfectly defined, yet still a modeling tool struggles. So complexity of the manifold presents the miner with a problem. What can the survey do about detecting this?

In truth, the answer is that the survey does little. The survey is designed to make a “quick once over” pass of the data set looking, in this case, for obvious problem areas. Fitting a function to a data set—that is, estimating the shape of the manifold—is the province of modeling, not surveying. Determining the shape of the manifold and measuring its complexity are computationally intensive, and no survey technique can do this short of building different models.

However, all is not completely lost. The output from a model is itself a data set, and it should estimate the shape of the manifold. Most modeling techniques indicate some measure of “goodness of fit” of the manifold to the data, but this is a general, overall measure. It is well worth the miner’s time to exercise the model over a representative range of inputs, thus deriving a data set that should describe the manifold. Surveying this *derived* (or predicted) data set will produce a survey map that looks at the predicted manifold shape and points to potential problem areas. Such a survey reveals exactly how much information was captured across the surface of the manifold. Where particularly problematic areas show up, building smaller models of the restricted, troublesome area very often produces better results in the restricted area than the general model. As a result, some models are used in some areas, while other models are used on other parts of the input space. But this is a modeling technique, rather than a surveying technique. Nonetheless, a sort of “post-survey survey” can point to problem areas with any model.

11.5 Clusters

Earlier, this chapter used the term “meaningful system states.” What exactly is a meaningful system state? The answer varies, and the question can only be answered within the framework of the problem domain. It might be that some sort of binning (described in Chapter 10) assigns continuous measurements to more meaningful labels. At other times, the measurements are meaningfully continuous, limited only by the granularity of the measurement (to the nearest penny, say, or the nearest degree). However, the system may inherently contain some system states that appear, from wholly internal evidence, to be meaningful within the system of variables. (This does not imply that they are necessarily meaningful in the real world.) The system “prefers” such internally meaningful states.

Recall that at this stage the data set is assumed to represent the population. Chapter 6 discussed the possibility that apparently preferred system states result from sampling bias preferentially sampling some system states over others. The miner needs to take care to eliminate such bias wherever possible. Those preferred system states that remain should tell something about the “natural” state of the system. But how does the miner find and identify any such states?

Chapter 6 discussed the idea that density of data points across state space varies. If areas that are more dense than average are imagined as points lower than average, and less dense points imagined to be higher, the density manifold can be conceived of as peaks and valleys. Each peak (the locally highest point) is surrounded by lower points. Each valley is surrounded by peaks and ridges. The ridges surrounding a particular valley actually are defined by a contour running through the lowest density surrounding a higher-density cluster. The valley bottoms actually describe the middle of higher-than-the-mean density clusters. These clusters represent the preferred states of the system of variables describing state space.

Such clusters, of course, represent likely system states. The survey identifies the borders and centers of these clusters, together with their probability. But more than that, it is often useful to aggregate these clusters as meaningful system states. The survey also makes an entropy map from all of the input clusters to all of the identified output clusters. This discovers if knowing which cluster an input falls into helps define an output.

For many states this is very useful information. Many models, both physical and behavioral, can make great use of such state models, even when precise models are not available. For instance, it may be enough to know for expensive and complex process machinery that it is “ok” or “needs maintenance” or is “about to fail.” If the output states fall naturally into one of these categories and the input states map well to the output states, a useful model may result even when precise predictions are not available from the model. Knowing what works allows the miner to concentrate on the borderline areas. Again, from behavioral data, it may be enough to map input and output states reliably to such categories as “unhappy customer warning,” “likely to churn,” and “candidate for cross-sell product X.”

Clustering is also useful when the miner is trying to decide if the data is biased.

11.6 Sampling Bias

Sampling bias is a major bugaboo and very hard to detect, but it’s easy to describe. When a sampling method repeatedly takes samples of data from a population that differ from the true population measures in the same way and in the same direction, then that method is introducing *sampling bias*. It is a distortion of the true values in the sample from those in the population that is introduced by the selection method itself, independent of other factors biasing the data. It is difficult to avoid since it may be quite unconsciously introduced. Since miners often work with data collected for purposes uncertain, by methods unknown, and with measurements obscure, after the fact detection of sampling bias may be all but impossible. Yet if the data does not reflect the real world, neither will any model mined, regardless of how assiduously it is checked against test and evaluation sample data sets.

The best that can be had from internal evaluation of a data set are clues that perhaps the data is biased. The only real answer lies in comparing the data with the world! However, that said, what can be done? There are two main types of sampling bias: errors of omission and errors of commission.

Errors of omission, of course, involve leaving out data that should be put in, whereas errors of commission involve putting in what should be left out. For instance, many interest groups seem to be able to prove a point completely at odds with the point proved by interest groups opposing them. Both sets of conclusions are solidly based on the data collected by each group, but, unconsciously or not, if the data is carefully selected to support desired conclusions, it can only tell a partial story. This may or may not be deliberately introduced bias. If an honest attempt to collect all the relevant data was made, but it still leads to dispute, it may be the result of sampling bias, either omission or commission. In spite of all the heat and argument, the only real answer is to collect all relevant data and look hard for possible bias.

As an example of the problem, an automobile manufacturer wanted to model vehicle reliability. A lot of data was available from the dealer network service records. But here was a huge problem. Quite aside from trying to decide what constitutes “reliability,” the data was very troublesome. For instance, those people who regularly used the dealer for service tended to be those people who took care of their vehicle and thus had reliable vehicles. On the other hand, repair work was often done for people who had no maintenance record with the dealer network. Conclusion: maintenance enhances reliability? Perhaps. But surely some people had maintenance outside of the dealer network. Some, perhaps, undertook their own maintenance and minor repairs, only having major work done at a dealer. There are any number of other possible biasing factors. Regardless of the possibilities, this was a very selective sample, almost certainly not representative of the population. So biased was this data that it was hard to build models of reliability even for those people who visited dealers, let alone the population at large!

Detecting such bias from the internal structure of the data is not possible. Any data set is what it is, and whether or not it accurately reflects the worldly phenomenon, it can never for sure be known just by looking at the data. But there might be clues.

The input data set covers a particular area. (A reminder that the term “area” is really applicable to a two-dimensional state space only, but it is convenient to use this term in general for the *n*-dimensional analog of area in other than two dimensions.) The output data set similarly covers its area. Any space in the input area maps, or points to, some particular space in the output area. This is illustrated in Figure 11.10. Exactly which part of the input space points to which part of the output space is defined by the relationship between them. The relevant spaces may be patches of different sizes and shapes from place to place, but the input points to some part of the output space, therefore being identified with some particular subsample, or patch, of the output sample.

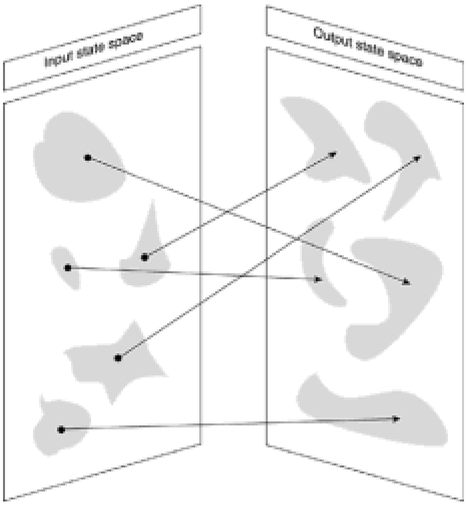


Figure 11.10 At least two data sets are used when modeling: an input data set (left) and an output data set (right).

It is often found in practice that for unbiased data sets, while the values of specific output variables change as the values of an input variable change, the distribution of data points at the different output values is fairly constant. For example, suppose, as illustrated in Figure 11.11, that the output patch of data points is normally distributed for some specific input value. As the input value changes, the output values will be expected to change (the patch moves through the output space), and the number of points in the output patch too is expected to change. However, if this assumption holds, wherever the output patch is located, the distribution of the points in it is expected to remain normally distributed.

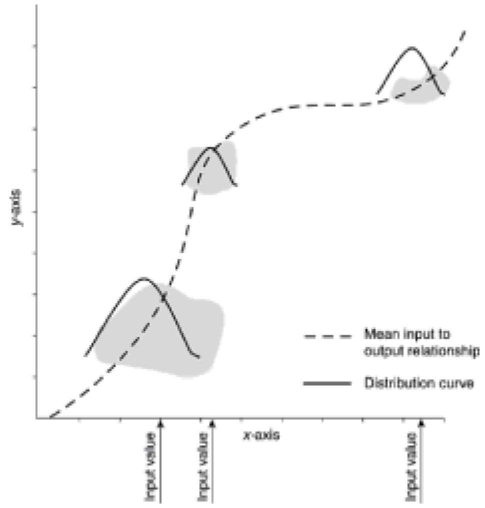


Figure 11.11 The output state space (made up of the *x* and *y* variables) has a manifold representing the input to output relationship. Any specific input value maps to some area in the output space, forming a “patch” (gray areas).

Figure 11.12 illustrates the idea that the distribution doesn’t change as the value of a variable changes. The effect of changing an input variable’s value is expected to change the output value and the number of instances in the subsample, but other factors are expected to remain the same so the shape of the distribution isn’t changed. Given that this often is true, what does it suggest when it is not true? Figure 11.13 illustrates a change in distribution as the *x* value changes. This distribution shift indicates that something other than just the *y* value has changed about the way the data responds to a change in the *x* value. Some other factor has certainly affected the way the data behaves at the two *x* values, and it is something external to the system of variables. This change may be caused by sampling bias or some other bias, but whatever the cause, the miner should account for the otherwise unaccounted change in system behavior between the two *x* values.

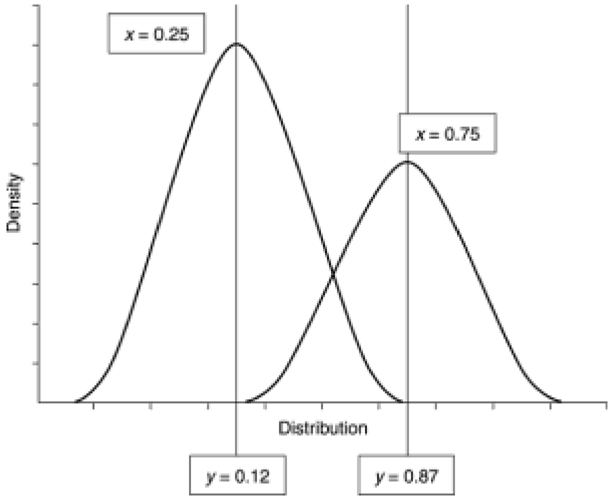


Figure 11.12 Distribution curves for the *x* values change the *y* values, but the curve remains similar in shape and not in size.

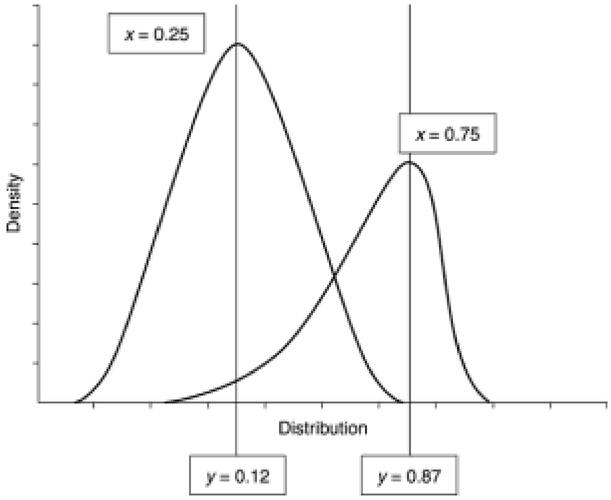


Figure 11.13 The change in *x* values is accompanied by a change in distribution shape as well as size. The tail is longer toward the low values for an *x* value of 0.75 than it is for an *x* value of 0.25.

The data survey samples the distributions, moving the input variables across their ranges of values. It makes a measurement of how much the distribution of output variables changes as inputs are moved, which is based on changes in both variability and skew.

11.7 Making the Data Survey

The components so far discussed form the basic backbone of the data survey. The survey's purpose is a quick look at the data. While modeling is a time-consuming process and focuses on detail, the survey deliberately focuses on the broad picture. The idea is not to resolve problems discovered, but to discover in broad terms the use and limits to use of the data and to be forewarned of any possible problem areas before modeling. The survey will not, and is not intended to, discover all of the problems that may lurk in data, and does little, if anything, toward fixing them. That is for the miner. But the miner can only fix known problems, and knowing comes from looking at the survey results. Survey software automatically makes these measurements so the miner can focus on problem areas by producing a unified numerical and graphical output. Even without such a tool, many statistical and data analysis tools have functions that allow the data to be surveyed manually. In spite of the brevity of the look at surveying in this chapter, time spent—manually if necessary—surveying data will repay the miner dividends.

Briefly, the steps for surveying data are

1. Sampling confidence—estimate level of multivariable distribution capture. (This confidence puts all of the other measures into perspective, whatever they are. If you're only 50% confident that the variability has been captured, it hardly matters what the other survey measurements are!)

2. Entropic analysis (normalized ranges)

   a. Input data set entropy—should be as high as possible

   b. Output data set entropy—should be as high as possible

   c. Other data set entropy—should be as high as possible, and similar among all of the data sets. (They should all be representative of the same population. Differences in these metrics mean something is fishy with the data!)

   d. Conditional entropy of outputs given inputs—should be as low as possible. If it is high, there is a problem! Either the data is snarled in some way, or the input simply doesn't contain sufficient information to predict the output. (Then try conditional entropy of outputs to inputs for comparison. If that's low, suspect a problem, not lack of information content. If it's high also, suspect insufficient information content.)

   e. Mutual information inputs to outputs—should be as high as possible

   f. Individual variable entropy input to output

   g. Individual between-variable entropy of the input (measures independence—may be useful too for data reduction)

3. Cluster analysis

   a. Plot peak, valley, and contour positions ("natural" clusters—do these make sense?

Why are they where they are? Could this be bias?)

b. Entropy of input clusters to output clusters—should be low. If not, there is a problem.

c. Cluster overlays—do input clusters map to output clusters, or do input clusters map across output clusters? (Overlaying each other is generally best, with small overlap.)

4. Manifold (maps only potential problem areas)

a. Sparsity—do sparse areas map to important output states? (If they do, it's a problem.)

b. Variability map (High variability will match areas of high uncertainty, but additional information given in distribution measures may help identify a problem.)

5. Sampling bias

a. Individual variable distribution input-driven output mapping—flag areas of high distribution drift. If there are many, is it sampling bias? In any case, why?

11.8 Novelty Detection

A novelty detector is not strictly part of the data survey, but is easily built from some of the various components that are discovered during the survey. The novelty detector is mainly used during the execution stage of using a model. Novelty detectors address the problem of ensuring that the execution data continues to resemble the training and test data sets.

Given a data set, it is moderately easy to survey it, or even to simply take basic statistics about the individual variables and the joint distribution, and from them to determine if the two data sets are drawn from the same population (to some chosen degree of confidence, of course). A more difficult problem arises when as assembled data set is not available, but the data to be modeled is presented on an instance-by-instance basis. Of course, each of the instances can be assembled into a data set, and that data set examined for similarity to the training data set, but that only tells you that the data set now assembled was or wasn't drawn from the same population. To use such a method requires waiting until sufficient instances become available to form a representative sample. It doesn't tell you if the instances arriving *now* are from the training population or not. And knowing if the current instance is drawn from the same population can be very important indeed, for reasons discussed in several places. In any case, if the distribution is not stationary (see Chapter 9 for a brief discussion of stationarity), no representative sample of instances is going to be assembled! So with a nonstationary distribution, collecting instances to form a representative sample to measure against the training sample presents problems.

Novelty detectors can also be used with enormously large data sets to help extract a more representative sample than the techniques alluded to in Chapter 10. Many large data sets contain very low-level fluctuations that are nonetheless important in a commercial sense, although insignificant in a statistical sense. The credit card issuer example in Chapter 12 demonstrates just such a case. When a representative sample is taken to some degree of confidence, it is the low-level fluctuations that are the most likely to be underrepresented in the sample. It is these low-level fluctuations that fall below the confidence threshold most readily. Finding some way to include these low-level fluctuations without bloating the sample can have great business value. It is in this role that a novelty detector can also contribute much.

So what exactly is a novelty detector? While there is no room here to go into the details of how a novelty detector is constructed, the principle is easy to see. Essentially a novelty detector is a device that estimates the probability that any particular instance value comes from the training population. The data survey has estimated the multidimensional distribution, and from that it is possible to estimate how likely any given instance value is to be drawn from that population. For a single variable, if normally distributed, such an estimate is provided by the standard deviation, or *z* value. So here, a novelty detector can be seen for what it really is—no more than the nonnormal-distribution, multidimensional equivalent of standard deviation.

Naturally, with a multidimensional distribution, and one that is most likely multidimensionally nonnormal at that, constructing such a measure is not exactly straightforward, but in principle there is little difficulty in constructing such a measure. It is convenient to construct such a measure to return a value that resembles the *z* score (mentioned in Chapter 5), and such measures are sometimes called pseudo-*z* scores (*pz*). It is convenient to embed a novelty detector generating a *pz* score into the PIE-I, although it is not a necessary part of the PIE as it plays no role in actually preparing data for the model. However, with such a *pz* score available from the PIE-I, it is relatively easy to monitor any “drift” in the execution values that might indicate that, at the least, some recalibration of the model is needed.

11.9 Other Directions

This whistle-stop tour of the territory covered by the data survey has only touched briefly on a number of useful topics. A survey does much more, providing the miner with a considerable amount of information prior to mining. The survey looks at data sets from several other perspectives.

For instance, individual variable distributions are mapped and compared. Recall from Chapter 7 that much useful information can be discovered from the actual shape of the distribution—some idea of underlying generating processes or biases. The survey maps similarities and differences. Sensitivity analysis is another area surveyed—areas where the manifold is most sensitive to changes in the data.

The survey also uses three metaphors for examining data. Two have been used in the chapters on data preparation—“manifolds” and “shapes.” A manifold is a flexible structure fitted in some way to the data set. The metaphor of a shape regards all of the data points as “corners” and regards the data set as creating some multidimensional structure.

The other useful metaphor used in the survey is that of a *tensegrity structure*. Tensegrity structures are sometimes seen as sculpture. The tensegrity structure is made of beams and wires. The beams are rigid and resist compression. The wires are in tension. This “string and poles” structure, as a sculpture, forms a three-dimensional object that is self-supporting. The compression in the beams is offset by the tension in the wires. (As a matter of interest, tensegrity structures occur naturally as, for instance, the internal “scaffolding” of body cells.) A data set can be imagined as a structure of points being pulled toward each other by some forces, and pushed apart by others, so that it reaches some equilibrium state. This is a sort of multidimensional tensegrity data structure. The data survey uses techniques to estimate the strength of the tension and compression forces, the “natural” shape of the data set, how much “energy” it contains, and how much “effort” it takes to move the shape into some other configuration. All of these measures relate to real-world phenomena and prove very useful in assessing reliability and applicability of models. It is also a useful analogy for finding “missing” variables. For instance, if the tensegrity structure is in some way distorted (not in equlibrium), there must be some missing part of the structure that, if present, holds the tensegrity shape in balance. Knowing what the missing information looks like can sometimes give a clue as to what additional data is needed to balance the structure. On the other hand, it might indicate sampling bias.

Being very careful not to confuse correlation with causality, the survey also looks at the “direction” of the influencing forces between variables and variable groupings. This uses techniques that measures the “friction” between variables or states. As one variable or state moves (changes state or value), so others move, but there is some loss (friction) or gain (amplification) in the interaction between all of the components of the variable system. Measuring this gives “directional” information about which component is “driving” which. There is much that can be done with such information, which is sometimes also called “influence.”

Other, new methods of looking at data are coming to the fore that offer, or promise to offer, useful survey information. The problem with many of them is that they are computationally intensive—which defeats the whole purpose of surveying. The whole idea is a quick once-over, searching deeper only where it looks like there might be particular problems. But new techniques from such fields as fractal analysis, chaos theory, and complexity theory hold much promise.

Fractals use a measurement of self-similarity. So far we have assumed that a modeler is looking at a relationship in data to model. The model pushes and pulls (usually mathematically) at the manifold in order to best fit it to the data set. Essentially, the manifold is a simple thing, manipulated into a complex shape. The model is complex; the manifold is simple. Fractals take a different approach. They assume that many areas of the manifold are indeed complex in shape, but similar to each other. It may then be enough to simply model a little bit of the complex manifold, and for the rest, simply point to where it fits, and how big it is in that location. Fractals, then, take a complex manifold and fit it together in simple ways, stretching and shrinking the shape as necessary. With fractals, the manifold is complex; the model is simple.

When it works, this is an enormously powerful technique. If a manifold does exhibit self-similarity, that alone is powerful knowledge. A couple of particularly useful fractal measures are the cluster correlation dimension and the Korack patchiness exponent. The problem with these techniques is, especially for high dimensionalities, they become computationally intensive—too much so, very often, for the data survey.

Chaos theory allows a search for attractors—system states that never exactly repeat, but around which the system orbits. Looking in data sets for attractors can be very useful, but again, these tend to be too computationally expensive for the survey, at least at present. However, computers are becoming faster and faster. (Indeed, as I write, there is commentary in the computer press that modern microprocessors are “too” fast and that their power is not needed by modern software! Use it for better surveying and mining!) Additional speed is making it possible to use these new techniques for modeling and surveying. Indeed, the gains in speed allow the automated directed search that modern surveying accomplishes. Very soon, as computer power increases survey techniques, new areas will provide practical and useful results.

Supplemental Material

Entropic Analysis—Example

After determining the confidence that the multivariable variability of a data set is captured, entropic analysis forms the main tool for surveying data. The other tools are useful, but used largely for exploring only where entropic or information analysis points to potential problems. Since entropic analysis is so important to the survey, this section shows the way in which the entropy measures were derived for the example in this chapter. Working through this section is not necessary to understand the topics covered here.

Calculating Basic Entropy

The example used two variables: the input variable and the output variable. The full range of calculations for forward and reverse entropy, signal entropy and mutual information, even for this simplified example, are quite extensive. For instance, determining the entropy of each of these two variables requires a fair number of calculations.

All probability-based statistics is based on counting the frequency of occurrence of values and joint combinations of values. Where truly continuously valued variables are used, this requires limiting the number of discrete values in the continuous range in some way, perhaps by binning the values. Bins are used in this example. Figure 11.14 has a reprise of Figure 11.1 in the upper image for comparison with the lower image. The upper image shows the original data set together with the manifold of a fitted function. The lower image shows the binned values for this data set. For comparison, a modeled manifold has been fitted to the binned data too, and although not identical in shape to that for the unbinned data, it is very similar.

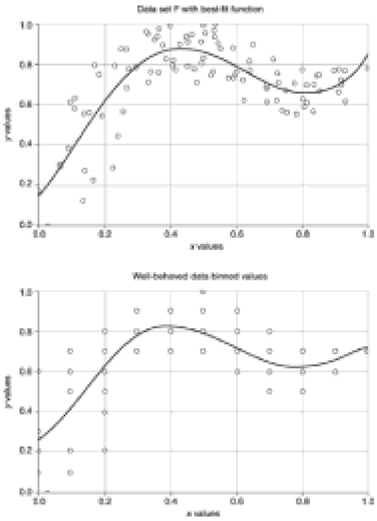


Figure 11.14 Test data set together with a manifold fitted by a modeling tool (top). The effect of binning the data (bottom); the circles show the center of the "full" bin positions.

Binning divides the state space into equally sized areas. If a data point falls into the area of the bin, that bin is denoted as "full." In an area with no data points, the appropriate bin is denoted as "empty." A circle shown on the lower image indicates a full bin. For simplicity, every bin is considered equally significant regardless of how many data points fall into it. This slightly simplifies the calculations and is reasonably valid so long as the bins are relatively small and each contains approximately the same number of points. (There are several other ways of binning such data. A more accurate method might weight the bins according to the number of data points in each. Another method might use the local density in each bin, rather than a count weighting.)

To make the calculation, first determine the frequency of the bins for X and Y. Simply count how many bins there are for each binned value of X, and then of Y, as shown in Table 11.9.

TABLE 11.9 Bin frequencies.

Bin	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
X	5	5	6	3	3	4	4	4	3	2	1
Y	1	2	3	1	1	4	7	10	6	4	1

This table indicates, for instance, that for the X value of 0, there are five bins. To discover this, look at the X value of 0, and count the bins stacked above that value.

From this, determine the relative frequency. (For instance, there are 40 bins altogether. For X bin value 0.0, there are five occurrences, and 5/40 = 0.125, which is the relative frequency.) This gives a relative frequency for the example data shown in Table 11.10.

TABLE 11.10 Bin relative frequencies.

Bin	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
X	0.125	0.125	0.150	0.075	0.075	0.100	0.100	0.100	0.075	0.050	0.025
Y	0.025	0.050	0.075	0.025	0.025	0.100	0.175	0.250	0.150	0.100	0.025

The reasoning behind the entropy calculations is already covered in the early part of this chapter and is not reiterated here. The relative frequency represents the probability of occurrence that directly allows the entropy determination as shown in Table 11.11.

TABLE 11.11 Entropy determination.

Log2(Px)	3.00	3.00	2.74	3.74	3.74	3.32	3.32	3.32	3.74	4.32	5.32

Log$_2$(Py)	5.32	4.32	3.74	5.32	5.32	3.32	2.52	2.00	2.74	3.32	5.32
P log$_2$(Px)	0.38	0.38	0.41	0.28	0.28	0.33	0.33	0.33	0.28	0.22	0.13
P log$_2$(Py)	0.13	0.22	0.28	0.13	0.13	0.33	0.44	0.50	0.41	0.33	0.13

The theoretical maximum entropy is P log$_2$(40), since there are 40 bins. The calculated entropy in this data set is, for X, <F128P9.5M%6>SP log$_2$(Px), and for Y, <F128P9.5M%6>SP log$_2$(Py):

Maximum entropy	3.459
Entropy X	3.347
Entropy Y	3.044

Clearly, there is not much to be estimated from simply knowing the values of X or Y as the entropy values are near to the maximum. Obviously, absolute entropy values will change with the number of bins. However, entropy can never be less than 0, not greater than the maximum entropy, so it is always possible to normalize these values across the range of 0–1.

Information-Driven Binning Strategies

Since bin count changes entropy measurements, some bin counts result in better or worse information measures than others. For any data set there is some optimum number of bins that best preserves the data set’s information content. Discovering this optimum bin size/count requires a search across several to many bin sizes. Any binning strategy loses some information, but some modeling tools require binning and others are enormously faster with binned data rather than continuous data. The performance and training trade-offs versus the information lost (usually small, particularly if an optimal binning strategy is used) frequently favor binning as a worthwhile practical strategy. When the optimal bin count is used, it is described as *least information loss binning*.

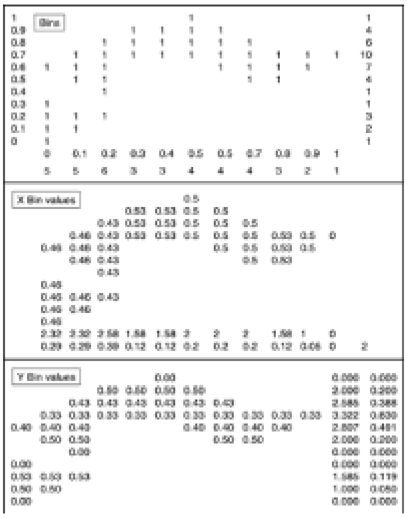
To complicate matters further, it is possible to use different bin sizes to best preserve information content. This is called *information-retentive adaptive binning* since the bin size adapts to optimize the information structure in the data.

Although the data survey derives these information-driven binning optimums, space constraints prevent a fuller discussion of the topic here.

Conditional Entropy and Mutual Information

Recall that mutual information between two data sets (individual variables in this example) is the entropy of one variable, less the entropy of the second, given the first. The first step is to find the entropy of all values of “the entropy of the second, given the first” for each discrete value of the first. This produces measures of conditional entropy for all values of one variable. Since both forward and reverse conditional entropy are surveyed, we must find both the conditional entropy of X given Y, and of Y given X for all values of X and Y.

Figure 11.15 shows the results of all the entropy calculations. The upper box labeled “Bins” reproduces in the pattern of ones the layout of bins shown in the lower part of Figure 11.14. For reference, to the left and immediately below the pattern of ones are shown the appropriate bin values to which each one corresponds. On the extreme right are the Y value bin counts, while on the extreme bottom are shown the X value bin counts. So, for instance, looking at the X value of 0, there are 5 ones in the vertical column above the X = 0 value, and so the bin count, shown below the X = 0 value, is 5. Similarly, for Y = 0.7 the horizontal bin count contains 10 ones, so the bin count is 10, as shown on the right.



1	Bins					1						1
0.9				1	1	1	1					4
0.8			1	1	1	1	1	1				6
0.7		1	1	1	1	1	1	1	1	1	1	10
0.6	1	1	1				1	1	1	1		7
0.5		1	1					1	1			4
0.4			1									1
0.3	1											1
0.2	1	1	1									3
0.1	1	1										2
0	1											1
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	
	5	5	6	3	3	4	4	4	3	2	1	

X Bin values						0.5						
				0.53	0.53	0.5	0.5					
			0.43	0.53	0.53	0.5	0.5	0.5				
		0.46	0.43	0.53	0.53	0.5	0.5	0.5	0.53	0.5	0	
	0.46	0.46	0.43				0.5	0.5	0.53	0.5		
		0.46	0.43					0.5	0.53			
			0.43									
	0.46											
	0.46	0.46	0.43									
	0.46	0.46										
	0.46											
	2.32	2.32	2.58	1.58	1.58	2	2	2	1.58	1	0	
	0.29	0.29	0.39	0.12	0.12	0.2	0.2	0.2	0.12	0.06	0	2

Y Bin values					0.00						0.000	0.000
			0.50	0.50	0.50	0.50					2.000	0.200
		0.43	0.43	0.43	0.43	0.43	0.43				2.585	0.388
	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.322	0.830
0.40	0.40	0.40				0.40	0.40	0.40	0.40		2.807	0.491
	0.50	0.50					0.50	0.50			2.000	0.200
		0.00									0.000	0.000
0.00											0.000	0.000
0.53	0.53	0.53									1.585	0.119
0.50	0.50										1.000	0.050
0.00											0.000	0.000

Figure 11.15 Calculating mutual entropy.

The two boxes “X Bin values” and “Y Bin values” maintain the same pattern, but show the entropy values for each state and bin. As an example, continue to look at the X bins for the value X = 0. There are five bins that match X = 0. These bins correspond to valid system states, or signals, and in this example we assume that they are each equally likely. Each of these five states has a probability of 1/5, or 0.2. The P $\log_2(P)$ for each of these five equally likely states is therefore 0.46. Thus the ones in the “Bins” box in the figure are replaced with 0.46s in the “X Bin values” box for the value of X = 0. This replacement is continued for all of the X and Y bin values in the appropriate boxes and with the appropriate values for each.

For all of the X bins, their values are summed and shown below the appropriate column. Thus, continuing to look at the X = 0 bins in the “X Bin values” box, the sum of these five values of 0.46 is 2.32, which is shown immediately below the bin column. (Rounding errors to two decimal places means that the figures shown seem slightly off. In fact, of course, the P $\log_2$(P) is slightly greater than 0.46, so that five of them sum to 2.32, not 2.30!) For the Y bins the sum is shown to the immediate right of the bin pattern in the “Y Bin values” box, as these are summarized horizontally.

Recall that altogether there are 40 signals (bins). For the value of X = 0, the probability of the 5 bins occurring is 5/40 = 0.125. So the value X = 0 occurs with probability 0.125. This 0.125 is the probability weighting for the system state X = 0 and is applied to the total bin sum of 2.32, giving an entropy measure for X = 0 of 0.29, shown on the lowest line below the X = 0 value in the “X Bin values” box. Similarly, the corresponding entropy values for all of the Y values are shown on the extreme right of the “Y Bin values” box. Summing the totals produces the measures shown in Table 11.12. Mapping the entropy values calculated here has already been shown in Figure 11.3.

TABLE 11.12 Example data set entropies.

Measure	Actual	Norm
Maximum entropy	3.459	
Entropy X	3.347	0.968
Entropy Y	3.044	0.880
Entropy (Y|X)	1.975	0.649
Entropy (X|Y)	2.278	0.681
Mutual info (X;Y)	1.069	0.351

Surveying Data Sets

As with so much else in life, there is a gap between theory and practical application with entropic analysis. Three sample data sets are included on the accompanying CD-ROM: CARS, SHOE, and CREDIT. This section looks at what useful insights the entropic analysis part of the data survey discovers before modeling this data. Just as there is not enough space in the chapter to make more than a brief introduction to some elements of the data survey, so too there is not space to look at, and discuss, more than a small portion of the entropic analysis of the data sets, let alone a full data survey. This section limits its attention to a small part of what the survey shows about the example data sets, specifically how entropic analysis can discover information about a data set, and how the miner can use the discovered information.

Introductory Note: Sequential Natural Clustering

Before looking at extracts of surveys of these data sets, the explanation needs a couple of introductory notes to give some perspective as to what these survey extracts reveal. Information analysis bases its measurements on features of system states. This means that some way of identifying system states has to be used to make the survey. There are many possible ways of identifying system states: several have to be included in any surveying software suite since different methods are appropriate for different circumstances. The method used in the following examples is that of *sequential natural clustering*.

Natural clusters form in the state space constructed from a representative sample. A group of points forms a natural cluster when a low-density boundary around the group separates those points inside the boundary from those points outside. The mean location of the center of such a cluster can itself be used as a representative point for the whole cluster. When this is done, it is possible then to move to another stage and cluster these representative points—a sort of cluster of clusters. Continuing this as far as possible eventually ends with a single cluster, usually centered in state space. However, since the clusters begin as very small aggregations, which lead to larger but still small aggregations, there can be many steps from start to finish. Each step has fewer clusters than the preceding step. At each step of clustering, the group of clusters that exist at that step are called a *layer*. Every input point maps into just one cluster at each layer. Each layer in the sequence is built from a natural clustering of the points at the previous layer—thus the name “sequential natural clustering.”

Both the input states and the output states are clustered. In the examples that follow, the output is limited to the states of a single variable. There is no reason to limit the output to a single variable save that it makes the explanation of these examples easier. In practice, miners often find that however large the input data set, the output states are represented by the states of a single variable. Sticking to a single variable as output for the examples here is not unrealistic. However, the tools used to make the survey are not limited to using only a single variable as output.

Sequential natural clustering has several advantages, one of which is that it allows the survey to estimate the complexity of the model required to explicate the information enfolded into the data set. There is no room here to look at the underlying explanation for why this is so, but since it is of particular interest to miners, it is shown in the survey extracts discussed.

A full survey digests a vast amount of metadata (data about the data set) and makes available an enormous amount of information about the entropic relationships between all of the variables, and between all of the layers. Unfortunately, a full discussion of a single survey is beyond the scope intended for this overview. Rather, we briefly examine the main points of what the entropic measures in a survey show, and why and how it is useful in practice to a miner.

The Survey Extract

The survey extracts used in the following examples report several measures. Each of the measures reveals useful information. However, before looking at what is reported about the data sets, here is a brief summary of the features and what they reveal.

Input layer 0 to output layer 0 In these extracts, the survey directly reports the input and output layer 0's entropic information. Layer 0 uses unclustered signals so that the entropies reported are of the raw input and output signal states. Using input layer 0 and output layer 0 measures the maximum possible information about the input and output. Thus, the layer 0 measures indicate the information content under the best possible circumstances—with the maximum amount of information exposed. It is possible, likely even, that modeling tools used to build the actual mined models cannot use all of the information that is exposed. As discussed in the examples, a miner may not even want to use all of this information. However, the layer 0 measures indicate the best that could possibly be done using a perfect modeling tool and using only the analyzed data set.

Any number of factors can intrude into the modeling process that prevent maximum information utilization, which is not necessarily a negative since trade-offs for modeling speed, say, may be preferable to maximum information extraction. For example, using a less complex neural network than is needed for maximum information extraction may train tens or hundreds of times faster than one that extracts almost all of the information. If the model is good enough for practical application, having it tens or hundreds of times earlier than otherwise may be more important than wringing all of the information out of a data set. This is always a decision for the miner, based, of course, on the business needs of the required model. However, the reason the extracts here show only the entropy measures for layer 0 is that this is the theoretical maximum that cannot be exceeded given the data at hand.

The complexity graph, mentioned below, uses information from other layers, as does the measurement of noise in the data set.

Signal H(X) Entropy can evaluate the relationship between input signals and output signals. However, it can also be used to evaluate the signals in a single variable. Recall that when the signals are evenly distributed, entropy is 1. The usual symbol for entropy is “H.” “X” symbolizes the input. Signal H(X) is evaluating the entropy of the input signal. These signals originate not from a single variable but from the whole input data set. (Recall that “signal” is definitely not synonymous with “variable.”) The measure indicates how much entropy there is in the data set input signal without regard to the output signal. It measures, among other things, how well “balanced” the input signal states are. An ideal data set needs each of the input signals to be equally represented, therefore equally uncertain. Thus Signal H(X) should be as high as possible, measured against the maximum possible entropy for the number of signal states. The ratio measurement makes this comparison of actual entropy against maximum possible entropy, and ideally it should be as close to 1 as possible.

The ratio is calculated as Signal H(X):$\log_2({}^{n}\text{input states})$.

Signal H(Y) Whereas “X” indicates the input states, “Y” indicates the output states. Signal H(Y) measures the entropy of the output signal states, and again, its ratio should be as high as possible. In other respects it is similar to Signal H(X) in that it too measures the entropy of the output states without regard to the input states.

The ratio is Signal H(Y):$\log_2({}^{n}\text{output states})$.

Channel H(X) The channel measurements are all taken about the data set as a whole. In all of the channel measurements the relationship between the input signals and the output signals is of paramount importance. It is called “channel entropy” because these measures regard the data set as a communication channel and they all indicate something about the fidelity of the communication channel—how well the input signals communicate information about the output, how much of all the information enfolded into the data set is used to specify the output, and how much information is lost in the communication process.

Channel H(X) is usually similar in its entropic measure to Signal H(X). The difference is that Signal H(X) measures the average information content in the signal without reference to anything else. Channel H(X) measures the average information per signal at the input of the communication channel. The output signals may interact with the input when the channel is considered. If, for instance, some of the discrete input signals all mean the same thing at the output, the information content of the input is reduced. (For instance, the words “yes,” “aye,” “positive,” “affirmative,” and “roger” may seem to be discrete signals. In some particular communication channel where all these signals indicate agreement and are completely synonymous with each other, they actually map to effectively the same input state. For these signals, signal entropy is based on four signals, whereas channel entropy is based on only one composite signal. For a more practical example, look back to Chapter 4 where “M-Benz,” “Merc,” and so on are all different signals for Signal H(X), but comprise a single signal for Channel H(X).)

Channel H(X) gives a measure of how well the input channel signals are balanced. If Signal H(X) and Channel H(X) differ considerably, it may be well worth the miner’s time to reconfigure the data to make them more similar. Channel H(X) is almost always less than Signal H(X). If Channel H(X) is much less, it implies that the model may have to be far more complex than if the two entropy measures are more nearly equal. Once again, the ratio measure should be as large as possible.

The ratio is Channel H(X):$\log_2$(*n*input states).

Note: In order to differentiate Channel H(X) and SignalH(X), and other entropy measures where confusion may occur, the symbols are preceded by “s” or “c” as appropriate to indicate signal or channel measures. Thus sH(X) refers to a signal measure, and cH(X) refers to a channel measure.

Channel H(Y) Channel H(Y) is like cH(X) except, of course, that the “Y” indicates measures about the output states. Another difference from the input is that any reduction in entropy from signal to channel indicates a potential problem since it means that some of the output states simply cannot be distinguished from each other as separate states. Ratio measures should be as close to 1 as possible.

The ratio is cH(Y):$\log_2$(^{n}output states).

Channel H(X|Y) Although listed first in the survey report, cH(X|Y) represents reverse entropy—the entropy of the input given the output. If it is less than the forward entropy—cH(Y|X)—then the problem may well be ill formed. (The example shown in the first part of this section shows an ill-formed relationship.) If this is the case, the miner will need to take corrective action, or get additional data to fix the problem. Channel H(X|Y) measures how much information is known about the input, given that a specific output has occurred.

The ratio is cH(X|Y):$\log_2$(^{n}input states).

Channel H(Y|X) This is a measure of the forward entropy, that is, how much information is known about the state of the output, given that a particular input has occurred. The ratio of this number needs to be as near to 0 as possible. Remember that entropy is a measure of uncertainty, and ideally there should be no uncertainty on average about the state of the output signal given the input signals. When the ratio of this measure is high (close to 1), very little is known about the state of the output given the state of the input.

The ratio is cH(Y|X):$\log_2$(^{n}output states).

Channel H(X,Y) (Note the comma, not a vertical bar.) Channel H(X,Y) measures the average information for every pair of input and output signals and the average uncertainty over the data set as a whole. In a sense, it measures how efficiently the data set enfolds (carries or represents) its information content. Perhaps another way of thinking of it is as yielding a measure of how much information in the data set isn't being used to define the output signals. (See cI(X;Y) below.)

The ratio is cH(X,Y):cH(X) + cH(Y).

Channel I(X;Y) This measures the mutual information between input and output (also between output and input since mutual information is a completely reciprocal measure). When the output signals are perfectly predicted by the input signals, ratio cI(X;Y) = 1. Note also that when the data set perfectly transfers all of its information, then cI(X;Y) = cH(X,Y).

The ratio is cI(X;Y):cH(Y).

Variable entropy and information measures Following the data set entropy and information measures are ratio measures for the individual variables. Due to space limitations, these examples do not always show all of the variables in a data set. Each variable has four separate measures:

H(X)—signal entropy for the individual variable

- H(Y|X)—showing how much information each variable individually carries about the output states

- I(X;Y)—the mutual information content for each individual variable with the output

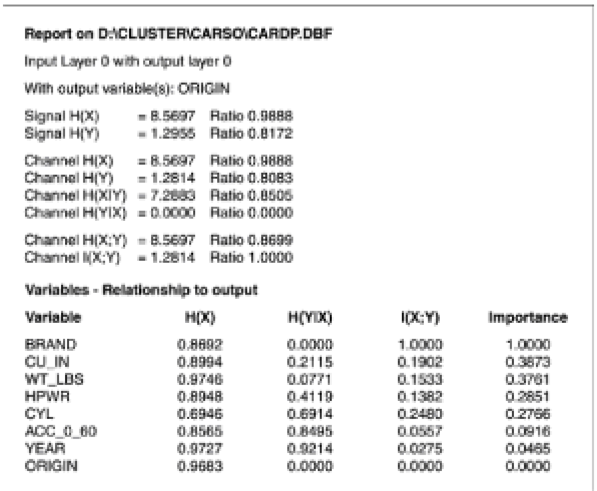
- Importance—an overall estimate of the uniqueness of the information contributed by each variable

The CARS Data Set

The CARS data set is fairly small, and although it is not likely to have to be mined to glean an understanding of what it contains, that property makes it a useful example! The data is intuitively understandable, making it easy to relate what the survey reveals about the relationships back to the data. For that reason, the examples here examine the CARS data set more extensively than is otherwise warranted. The meanings of the variable names are fairly self-evident, which makes interpretation straightforward. Also, this data set is close to actually being the population! Although the measure of possible sampling error (not shown) indicates the possible presence of sampling error, and although the “sample” is “small,” the miner can establish that details of most of the car models available in the U.S. for the period covered are actually in the data set.

Predicting Origin

Information metrics Figure 11.16 shows an extract of the information provided by the survey. The cars in the data set may originate from Europe, Japan, or the U.S. Predicting the cars’ origins should be relatively easy, particularly given the brand of each car. But what does the survey have to say about this data set for predicting a car’s origin?



```
Report on D:\CLUSTER\CARSO\CARDP.DBF

Input Layer 0 with output layer 0

With output variable(s): ORIGIN

Signal H(X)       = 8.5697   Ratio 0.9888
Signal H(Y)       = 1.2955   Ratio 0.8172

Channel H(X)      = 8.5697   Ratio 0.9888
Channel H(Y)      = 1.2814   Ratio 0.8083
Channel H(X|Y)    = 7.2883   Ratio 0.8505
Channel H(Y|X)    = 0.0000   Ratio 0.0000

Channel H(X;Y)    = 8.5697   Ratio 0.8699
Channel I(X;Y)    = 1.2814   Ratio 1.0000

Variables - Relationship to output

Variable            H(X)      H(Y|X)     I(X;Y)     Importance

BRAND               0.8692    0.0000     1.0000     1.0000
CU_IN               0.8994    0.2115     0.1902     0.3873
WT_LBS              0.9746    0.0771     0.1533     0.3761
HPWR                0.8948    0.4119     0.1382     0.2851
CYL                 0.6946    0.6914     0.2480     0.2766
ACC_0_60            0.8565    0.8495     0.0557     0.0916
YEAR                0.9727    0.9214     0.0275     0.0465
ORIGIN              0.9683    0.0000     0.0000     0.0000
```

Figure 11.16 Extract of the data survey report for the CARS data set when predicting the cars ORIGIN. Cars may originate from Japan, the U.S., or Europe.

First of all, sH(X) and sH(Y) are both fairly close to 1, showing that there is a reasonably good spread of signals in the input and output. The sH(Y) ratio is somewhat less than 1, and looking at the data itself will easily show that the numbers of cars from each of the originating areas is not exactly balanced. But it is very hard indeed for a miner to look at the actual input states to see if they are balanced—whereas the sH(X) entropy shows clearly that they are. This is a piece of very useful information that is not easily discovered by inspecting the data itself.

Looking at the channel measures is very instructive. The signal and channel H(X) are identical, and signal and channel H(Y) are close. All of the information present in the input, and most of the information present in the output, is actually applied across the channel.

cH(X|Y) is high, so that the output information poorly defines the state of the input, but that is of no moment. More importantly, cH(X|Y) is greater than cH(Y|X)—much greater in this case—so that this is not an ill-defined problem. Fine so far, but what does cH(Y|X) = 0 mean? That there is no uncertainty about the output signal given the input signal. No uncertainty is exactly what is needed! The input perfectly defines the output. Right here we immediately know that it is at least theoretically possible to perfectly predict the origin of a car, given the information in this data set.

Moving ahead to cI(X;Y) = 1 for a moment, this too indicates that the task is learnable, and that the information inside the channel (data set) is sufficient to completely define the output. cH(X;Y) shows that not all of the information in the data set is needed to define the output.

Let us turn now to the variables. (All the numbers shown for variables are ratios only.) These are listed with the most important first, and BRAND tells a story in itself! Its cH(Y|X) = 0 shows that simply knowing the brand of a vehicle is sufficient to determine its origin. The cH(Y|X) says that there is no uncertainty about the output given only brand as an input. Its cI(X;Y) tells the same story—the 1 means perfect mutual information. (This conclusion is not at all surprising in this case, but it's welcome to have the analysis confirm it!) It's not surprising also that its importance is 1. It's clear too that the other variables don't seem to have much to say individually about the origin of a car.

This illustrates a phenomenon described as *coupling*. Simply expressed, coupling measures how well information used by a particular set of output signals connects to the data set as a whole. If the coupling is poor, regardless of how well or badly the output is defined by the input signals, very little of the total amount of information enfolded in the data set is used. The higher the coupling, the more the information contained in the data set is used.

Here the output signals seem only moderately coupled to the data set. Although a coupling ratio is not shown on this abbreviated survey, the idea can be seen here. The prediction of the states of ORIGIN depends very extensively on states of BRAND. The other variables do not seem to produce signal states that well define ORIGIN. So, superficially it seems that the prediction of ORIGIN requires the variable BRAND, and if that were removed, all might be lost. But what is not immediately apparent here (but is shown in the next example to some extent) is that BRAND couples to the data set as a whole quite well. (That is, BRAND is well integrated into the overall information system represented by the variables.) If BRAND information were removed, much of the information carried by this variable can be recovered from the signals created by the other variables. So while ORIGIN seems coupled only to BRAND, BRAND couples quite strongly to the information system as a whole. ORIGIN, then, is actually more closely coupled to this data set than simply looking at individual variables may indicate. Glancing at the variable's metrics may not show how well—or poorly—signal states are in fact coupled to a data set. The survey looks quite deeply into the information system to discover coupling ratios. In a full survey this coupling ratio can be very important, as is shown in a later example.

When thinking about coupling, it is important to remember that the variables defining the manifold in a state space are all interrelated. This is what is meant by the variables being part of a system of variables. Losing, or removing, any single variable usually does not remove all of the information carried by that variable since much, perhaps all, of the information carried by the variable may be duplicated by the other variables. In a sense, coupling measures the degree of the total interaction between the output signal states and all of the information enfolded in the data set, regardless of where it is carried.

Complexity map A *complexity map* (Figure 11.17) indicates highest complexity on the left, with lower complexity levels progressively further to the right. Information recovery indicates the amount of information a model could recover from the data set about the output signals: 1 means all of it, 0 means none of it. This one shows perfect predictability (information recovery = 1) for the most complex level (complexity level 1). The curve trends gently downward at first as complexity decreases, eventually flattening out and remaining almost constant as complexity reduces to a minimum.

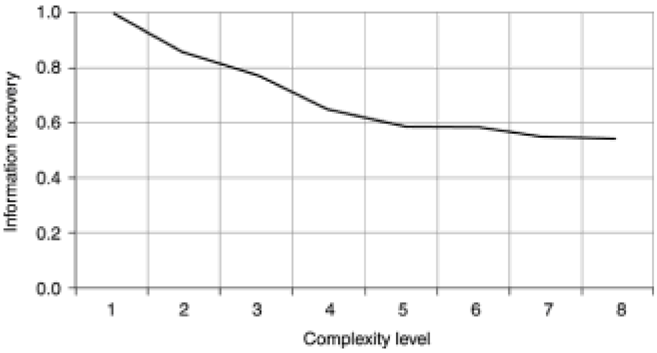


Figure 11.17 Complexity map for the CARS data set when predicting ORIGIN. Highest complexity is on the left, lowest complexity is on the right. (Higher numbers mean less complexity.)

In this case the data set represents the population. Also, a predictive model is not likely to be needed since any car can be looked up in the data. The chances are that a miner is looking to understand relationships that exist in this data. In this unusual situation where the whole population is present, noise is not really an issue. There may certainly be erroneous entries and other errors that constitute noise. The object is not to generalize relationships from this data that are then to be applied to other similar data. Whatever can be discovered in this data is sufficient, since it works in this data set, and there is no other data set to apply it to.

The shallow curve shows that the difficulty of recovering information increases little with increased complexity. Even the simplest models can recover most of the information. This complexity map promises that a fairly simple model will produce robust and effective predictions of origin using this data. (Hardly stunning news in this simple case!)

State entropy map A *state entropy map* (Figure 11.18) can be one of the most useful maps produced by the survey. This map shows how much information there is in the data set to define each state. Put another way, it shows how accurately, or confidently, each output state is defined (or can be predicted). There are three output signals shown, indicated as “1,” “2,” and “3” along the bottom of the map. These correspond to the output signal states, in this case “U.S.,” “Japan,” and “Europe.” For this brief look, the actual list of which number applies to which signal is not shown. The map shows a horizontal line that represents the average entropy of all of the outputs. The entropy of each output signal is shown by the curve. In this case the curve is very close to the average, although signal 1 has slightly less entropy than signal 2. Even though the output signals are perfectly identified by the input signals, there is still more uncertainty about the state of output signal 2 than of either signal 1 or signal 3.

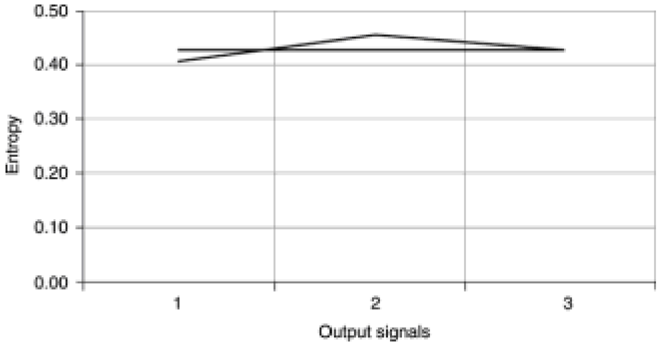


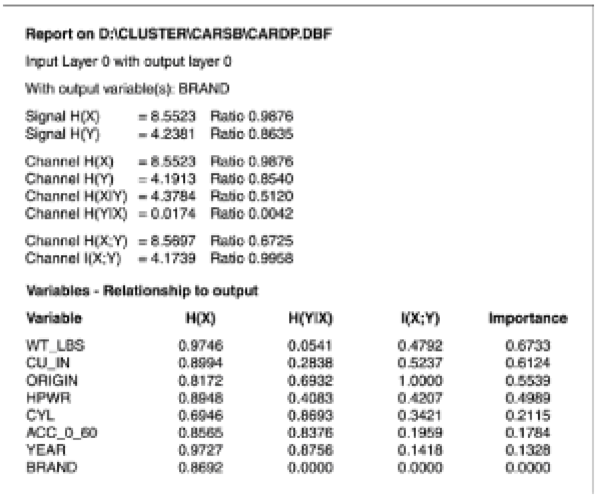
Figure 11.18 State entropy map for the CARS data set when predicting ORIGIN. The three states of ORIGIN are shown along the bottom of the graph (U.S., Japan, and Europe).

Summary No really startling conclusions jump out of the survey when investigating country of origin for American cars! Nevertheless, the entropic analysis confirmed a number of intuitions about the CARS data that would be difficult to obtain by any other means, particularly including building models.

This is an easy task, and only a simple model using a single-input variable, BRAND, is needed to make perfect predictions. However, no surprises were expected in this easy introduction to some small parts of the survey.

Predicting Brand

Information metrics Since predicting ORIGIN only needed information about the BRAND, what if we predict the BRAND? Would you expect the relationship to be reciprocal and have ORIGIN perfectly predict BRAND? (Hardly. There are only three sources of origin, but there are many brands.) Figure 11.19 shows the survey extract using the CARS data set to predict the BRAND.



```
Report on D:\CLUSTER\CARSB\CARDP.DBF

Input Layer 0 with output layer 0

With output variable(s): BRAND

Signal H(X)        = 8.5523   Ratio 0.9876
Signal H(Y)        = 4.2381   Ratio 0.8635

Channel H(X)       = 8.5523   Ratio 0.9876
Channel H(Y)       = 4.1913   Ratio 0.8540
Channel H(X|Y)     = 4.3784   Ratio 0.5120
Channel H(Y|X)     = 0.0174   Ratio 0.0042

Channel H(X;Y)     = 8.5697   Ratio 0.6725
Channel I(X;Y)     = 4.1739   Ratio 0.9958

Variables - Relationship to output

Variable          H(X)        H(Y|X)      I(X;Y)      Importance

WT_LBS            0.9746      0.0541      0.4792      0.6733
CU_IN             0.8994      0.2838      0.5237      0.6124
ORIGIN            0.8172      0.6932      1.0000      0.5539
HPWR              0.8948      0.4083      0.4207      0.4989
CYL               0.6946      0.8693      0.3421      0.2115
ACC_0_60          0.8565      0.8376      0.1959      0.1784
YEAR              0.9727      0.8756      0.1418      0.1328
BRAND             0.8692      0.0000      0.0000      0.0000
```

Figure 11.19 Part of the survey report for the CARS data set with output signals defined by the variable BRAND.

A quick glance shows that the input and output signals are reasonably well distributed (H(X) and H(Y)), the problem is not ill formed (H(X|Y) and H(Y|X)), and good but not perfect predictions of the brand of car can be made from this data (H(Y|X) and I(X;Y)).

BRAND is fairly well coupled to this data set with weight and cubic inch size of the engine carrying much information. ORIGIN appears third in the list with a cI(X;Y) = 1, which goes to show the shortcoming of relying on this as a measure of predictability! This is a completely reciprocal measure. It indicates complete information in one direction or the other, but without specifying direction, so which predicts what cannot be determined. Looking at the individual cH(Y|X)s for the variables, it seems that it carries less information than horsepower (HPWR), the next variable down the list.

Complexity map The diagonal line is a fairly common type of complexity map (Figure 11.20). Although the curve appears to reach 1, the cI(X;Y), for instance, shows that it must fall a minute amount short, since the prediction is not perfect, even with a highest degree of complexity model. There is simply insufficient information to completely define the output signals from the information enfolded into the data set.

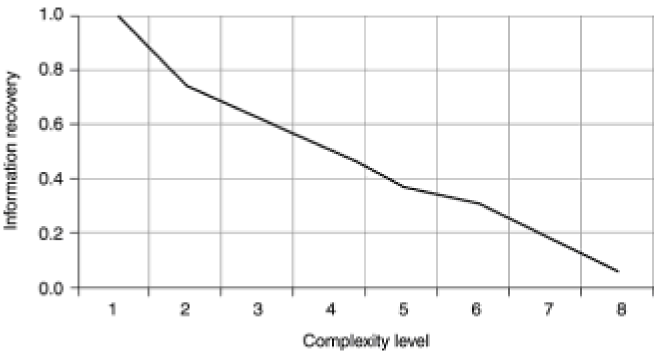


Figure 11.20 Complexity map for the CARS data set using output signals from the variable BRAND.

Once again, noise and sample size limitations can be ignored as the entire population is present. This type of map indicates that a complex model, capturing most of the complexity in the information, will be needed to build the model.

State entropy map Perhaps the most interesting feature of this survey is the *state entropy map* (Figure 11.21). The variable BRAND, of course, is a categorical variable. Prior to the survey it was numerated, and the survey uses the numerated information. Interestingly, since the survey looks at signals extracted from state space, the actual values assigned to BRAND are not important here, but the ordering reflected out of the data set is important. The selected ordering reflected from the data set shown here is clearly not a random choice, but has been somehow arranged in what turns out to be approximately increasing levels of certainty. In this example, the exact labels that apply to each of the output signals is not important, although they will be very interesting (maybe critically important, or may at least lend a considerable insight) in a practical project!

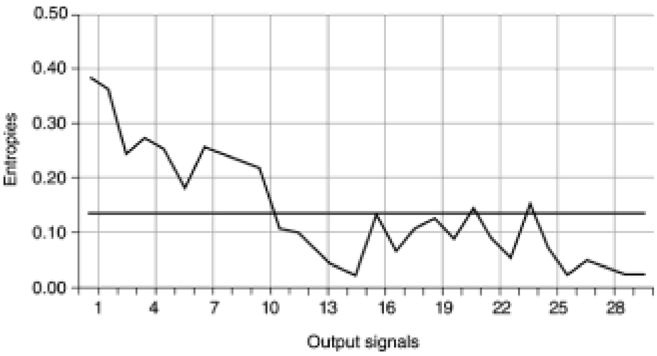


Figure 11.21 State entropy map for the CARS data set and BRAND output signals. The signals corresponding to positions on the left are less defined (have a higher entropy) than those on the right.

Once again, the horizontal line shows the mean level of entropy for all of the output signals. The entropy levels plotted for each of the output signals form the wavy curve. The numeration has ordered the vehicle brands so that those least well determined—that is, those with the highest level of entropy—are on the left of this map, while the best defined are on the right. From this map, not only can we find a definitive level of the exact confidence with which each particular brand can be predicted, but it is clear that there is some underlying phenomenon to be explained. Why is there this difference? What are the driving factors? How does this relate to other parts of the data set? Is it important? Is it meaningful?

This important point, although already noted, is worth repeating, since it forms a particularly useful part of the survey. The map indicates that there are about 30 different brands present in the data set. The information enfolded in the data set does, in general, a pretty good job of uniquely identifying a vehicle's brand. That is measured by the cH(Y|X). This measurement can be turned into a precise number specifying exactly how well—in general—it identifies a brand. However, much more can be gleaned from the survey. It is also possible to specify, for each individual brand, how well the information in the data specifies that a car is or is not that brand. That is what the state entropy map shows. It might, for instance, be possible to say that a prediction of "Ford" will be correct 999 times in 1000 (99.9% of the time), but "Toyota" can only be counted on to be correct 75 times in 100 (75% of the time).

Not shown, but also of considerable importance in many applications, it is possible to say which signals are likely to be confused with each other when they are not correctly specified. For example, perhaps when "Toyota" is incorrectly predicted, the true signal is far more likely to be "Honda" than "Nissan"—and whatever it is, it is very unlikely to be "Ford." Exact confidence levels can be found for confusion levels of all of the output signals. This is very useful and sometimes crucial information.

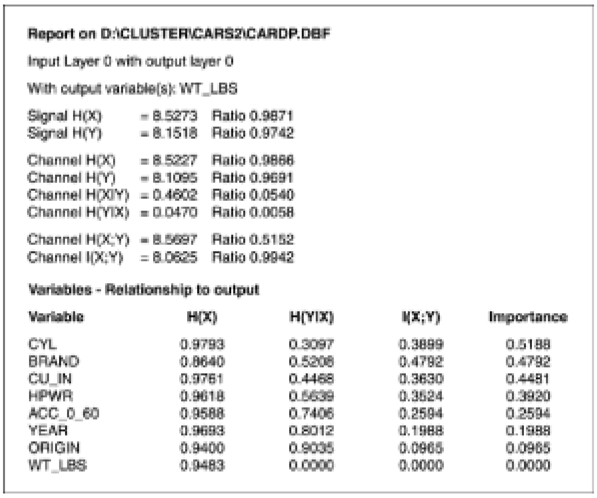
Recall also that this information is all coming out of the survey before any models have been built! The survey is not a model as it can make no predictions, nor actually identify the nature of the relationships to be discovered. The survey only points out potential—possibilities and limitations.

Summary Modeling vehicle brand requires a complex model to extract the maximum information from the data set. Brand cannot be predicted with complete certainty, but limits to accuracy for each brand, and confidence levels about confusion between brands, can be determined. The output states are fairly well coupled into the data set, so that any models are likely to be robust as this set of output signals is itself embedded and intertwined in the complexity of the system of variables as a whole. Predictions are not unduly influenced only by some limited part of the information enfolded in the data set.

There is clearly some phenomenon affecting the level of certainty across the ordering of brands that needs to be investigated. It may be spurious, evidence of bias, or a significant insight, but it should be explained, or at least examined. When a model is built, precise levels of certainty for the prediction of each specific brand are known, and precise estimates of which output signals are likely to be confused with which other output signals are also known.

Predicting Weight

Information metrics There seem to be no notable problems predicting vehicle weight (WT_LBS). In Figure 11.22, cH(X|Y) seems low—the input is well predicted by the output—but as we will see, that is because almost every vehicle has a unique weight. The output signals seem well coupled into the data set.



```
Report on D:\CLUSTER\CARS2\CARDP.DBF

Input Layer 0 with output layer 0

With output variable(s): WT_LBS

Signal H(X)        = 8.5273   Ratio 0.9871
Signal H(Y)        = 8.1518   Ratio 0.9742

Channel H(X)       = 8.5227   Ratio 0.9866
Channel H(Y)       = 8.1095   Ratio 0.9691
Channel H(X|Y)     = 0.4602   Ratio 0.0540
Channel H(Y|X)     = 0.0470   Ratio 0.0058

Channel H(X;Y)     = 8.5697   Ratio 0.5152
Channel I(X;Y)     = 8.0625   Ratio 0.9942

Variables - Relationship to output

Variable            H(X)        H(Y|X)       I(X;Y)      Importance

CYL                 0.9793      0.3097       0.3899       0.5188
BRAND               0.8640      0.5208       0.4792       0.4792
CU_IN               0.9761      0.4468       0.3630       0.4481
HPWR                0.9618      0.5639       0.3524       0.3920
ACC_0_60            0.9588      0.7406       0.2594       0.2594
YEAR                0.9693      0.8012       0.1988       0.1988
ORIGIN              0.9400      0.9035       0.0965       0.0965
WT_LBS              0.9483      0.0000       0.0000       0.0000
```

Figure 11.22 Survey extract for the CARS data set predicting vehicle weight (WT_LBS).

There is a clue here in cH(Y|X) and cH(X|Y) that the data is overly specific, and that if generalized predictions were needed, a model built from this data set might well benefit from the use of a smoothing technique. In this case, but only because the whole population is present, that is not the case. This discussion continues with the explanation of the state entropy map for this data set and output.

Complexity map Figure 11.23 shows the complexity map. Once again, a diagonal line shows that a more complex model gives a better result.

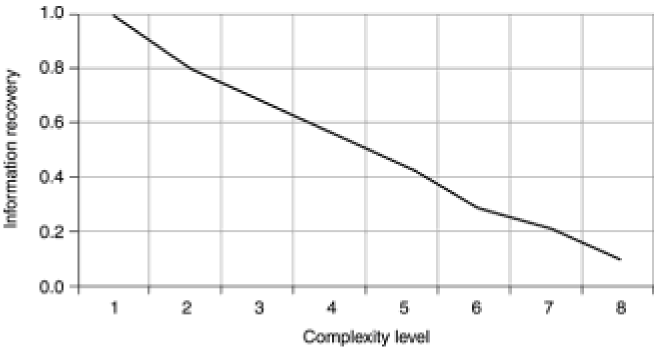


Figure 11.23 Complexity map for the CARS data set predicting vehicle weight.

State entropy map This state entropy map (Figure 11.24) shows many discrete values. In fact, as already noted, almost every vehicle has a unique weight. Since the map shows spikes—in spite of the generally low level of entropy of the output, which indicates that the output is generally well defined—the many spikes show that several, if not many, vehicles are not well defined by the information enfolded into the data set. There is no clear pattern revealed here, but it might still be interesting to ask why certain vehicles are (anomalously?) not well specified. It might also be interesting to turn the question around and ask what it is that allows certainty in some cases and not others. A complete survey provides the tools to explore such questions.

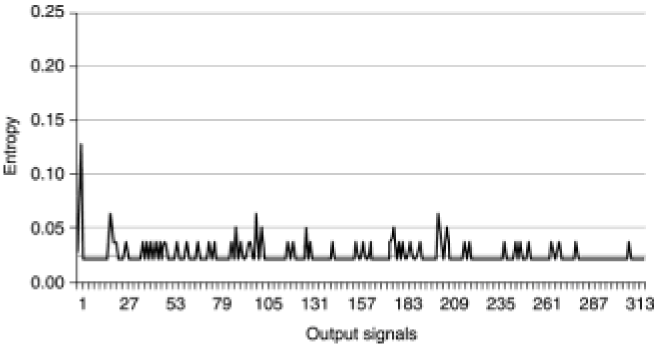


Figure 11.24 State entropy map for the CARS data set with output vehicle weight. The large number of output states reflects that almost every vehicle in the data set weighs a different amount than any of the other vehicles.

In this case, essentially the entire population is present. But if some generalization were needed for making predictions in other data sets, the spikes and high number of discrete values indicate that the data needs to be modified to improve the generalization. Perhaps least information loss binning, either contiguously or noncontiguously, might help. The clue that this data might benefit from some sort of generalization is that both cH(Y|X) and cH(X|Y) are so low. This can happen when, as in this case, there are a large number of discrete inputs and outputs. Each of the discrete inputs maps to a discrete output.

The problem for a model is that with such a high number of discrete values mapping almost directly one to the other, the model becomes little more than a lookup table. This works well only when every possible combination of inputs to outputs is included in the training data set—normally a rare occurrence. In this case, the rare occurrence has turned up and all possible combinations are in fact present. This is due entirely to the fact that this data set represents the population, rather than a sample. So here, it is perfectly valid to use the lookup table approach.

If this were instead a small but representative sample of a much larger data set, it is highly unlikely that all combinations of inputs and outputs would be present in the sample. As soon as a lookup-type model (known also as a *particularized mode*l) sees an input from a combination that was not in the training sample, it has no reference or mechanism for generalizing to the appropriate output. For such a case, a useful model generalizes rather than particularizes. There are many modeling techniques for building such generalized models, but they can only be used if the miner knows that such models are needed. That is not usually hard to tell. What is hard to tell (without a survey) is what level of generalization is appropriate.

Having established from the survey that a generalizing model is needed, what is the appropriate level of generalization? Answering that question in detail is beyond the scope of this introduction to a survey. However, the survey does provide an unambiguous answer to the appropriate level of generalization that results in least information loss for any specific required resolution in the output (or prediction).

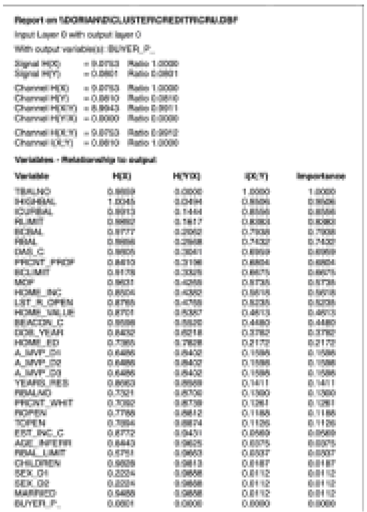
Summary Apart from the information discussed in the previous examples, looking at vehicle weight shows that some form of generalized model has to be built for the model to be useful in other data sets. A complete survey provides the miner with the needed information to be able to construct a generalized model and specifies the accuracy and confidence of the model's predictions for any selected level of generalization. Before modeling begins, the miner knows exactly what the trade-offs are between accuracy and generalization, and can determine if a suitable model can be built from the data on hand.

The CREDIT Data Set

The CREDIT data set represents a real-world data set, somewhat cleaned (it was assembled from several disparate sources) and now ready for preparation. The objective was to build an effective credit card solicitation program. This is data captured from a previous program that was not particularly successful (just under a 1% response rate) but yielded the data with which to model customer response. The next solicitation program, run using a model built from this data, generated a better than 3% response rate.

This data is slightly modified from the actual data. It is completely anonymized and, since the original file comprised 5 million records, it is highly reduced in size!

Information metrics Figure 11.25 shows the information metrics. The data set signals seem well distributed, sH(X) and cH(X), but there is something very odd about sH(Y) and cH(Y)—they are so very low. Since entropy measures, among other things, the level of uncertainty in the signals, there seems to be very little uncertainty about these signals, even before modeling starts! The whole purpose of predictive models is to reduce the level of uncertainty about the output signal given an input signal, but there isn't much uncertainty here to begin with! Why?



```
Report on \DORIAN\CLUSTER\CREDIT\CRU.DBF
Input Layer 0 with output layer 0
With output variable(s): BUYER_P_

Signal H(X)        = 9.8753   Ratio 1.0000
Signal H(Y)        = 0.0801   Ratio 0.0801

Channel H(X)       = 9.8753   Ratio 1.0000
Channel H(Y)       = 0.0810   Ratio 0.0810
Channel H(X|Y)     = 8.8943   Ratio 0.9011
Channel H(Y|X)     = 0.0000   Ratio 0.0000

Channel H(X,Y)     = 9.8753   Ratio 0.9912
Channel I(X;Y)     = 0.0810   Ratio 1.0000

Variables - Relationship to output

Variable           H(X)      H(Y|X)      I(X;Y)     Importance

TBALNO             0.9889    0.0000      1.0000     1.0000
IHIGHBAL           1.0045    0.0494      0.8506     0.8506
ICURBAL            0.8913    0.1444      0.8556     0.8556
RLIMIT             0.9862    0.1617      0.8383     0.8383
BCBAL              0.9777    0.2062      0.7938     0.7938
RBAL               0.9866    0.2568      0.7432     0.7432
DMS_C              0.9805    0.3041      0.6959     0.6959
PRCNT_PROF         0.8610    0.3196      0.6804     0.6804
BCLIMIT            0.9178    0.3325      0.6675     0.6675
MOF                0.9631    0.4265      0.5735     0.5735
HOME_INC           0.8504    0.4382      0.5618     0.5618
LST_R_OPEN         0.8765    0.4765      0.5235     0.5235
HOME_VALUE         0.8701    0.5387      0.4613     0.4613
BEACON_C           0.9598    0.5520      0.4480     0.4480
DOB_YEAR           0.8432    0.6218      0.3782     0.3782
HOME_ED            0.7365    0.7828      0.2172     0.2172
A_MVP_D1           0.6486    0.8402      0.1598     0.1598
A_MVP_D2           0.6486    0.8402      0.1598     0.1598
A_MVP_D3           0.6486    0.8402      0.1598     0.1598
YEARS_RES          0.8663    0.8589      0.1411     0.1411
RBALNO             0.7321    0.8700      0.1300     0.1300
PRCNT_WHIT         0.7092    0.8739      0.1261     0.1261
ROPEN              0.7788    0.8812      0.1188     0.1188
TOPEN              0.7884    0.8874      0.1126     0.1126
EST_INC_C          0.8772    0.9431      0.0569     0.0569
AGE_INFERR         0.8443    0.9625      0.0375     0.0375
RBAL_LIMIT         0.5751    0.9463      0.0537     0.0537
CHILDREN           0.9828    0.9813      0.0187     0.0187
SEX_D1             0.2224    0.9888      0.0112     0.0112
SEX_D2             0.2224    0.9888      0.0112     0.0112
MARRIED            0.5488    0.9888      0.0112     0.0112
BUYER_P_           0.0801    0.0000      0.0000     0.0000
```

Figure 11.25 Information metrics for the CREDIT data set.

The reason, it turns out, is because this is the unmodified response data set with a less than 1% response rate. The fact is that if you guessed the state of a randomly selected record, you would be right more than 99% of the time by guessing that record referred to a nonbuyer. Not really much uncertainty about the output at all!

Many modeling techniques—neural networks or regression, for example—cannot deal with such low levels of response. In fact, very many methods have trouble with such low levels of response as this unless especially tuned to deal with it. However, since information metrics measure the nature of the manifold in state space, they are remarkably resistant to any distortion due to very low-density responses. Continuing to look at this data set, and later comparing it with a balanced version, demonstrates the point nicely.

With a very large data set, such as is used here, and a very low response rate, the rounding to four places of decimals, as reported in the information metrics, makes the ratio of cH(Y|X) appear to equal 0, and cI(X;Y) appears to be equal to 1. However, the state entropy map shows a different picture, which we will look at in a moment.

Complexity map This is an unusual, and really a rather nasty-looking, complexity map seen in Figure 11.26. The concave-shaped curve indicates that adding additional complexity to the model (starting with the simplest model on the right) gains little in predictability. It takes a really complex model, focusing closely on the details of the signals, to extract any meaningful determination of the output signals.

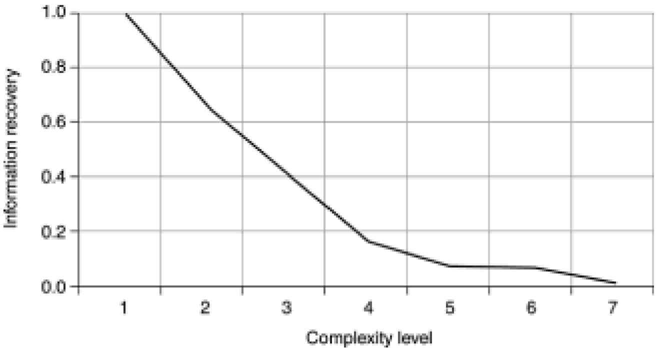


Figure 11.26 Complexity map for the CREDIT data set predicting BUYER. This curve indicated that the data set is likely to be very difficult to learn.

If this data set were the whole population, as with the CARS data set, there would be no problem. But here the situation is very different. As discussed in many places through the book (see, for example, Chapter 2), when a model becomes too complex or learns the structure of the data in too much detail, overtraining, or learning spurious patterns called noise, occurs. That is exactly the problem here. The steep curve on the left of the complexity map indicates that meaningful information is only captured with a high complexity model, and naturally, that is where the noise lies! The survey measures the amount of noise in a data set, and although a conceptual technical description cannot be covered here, it is worth looking at a noise map.

Noise Figure 11.27 shows the *information and noise map* for the CREDIT data set. The curve beginning at the top left (identical with that in Figure 11.26) shows how much information is recovered for a given level of complexity and is measured against the vertical scale shown on the left side of the map. The curve ending at the top right shows how much noise is captured for a given level of complexity and is measured against the vertical scale shown on the right side of the map.

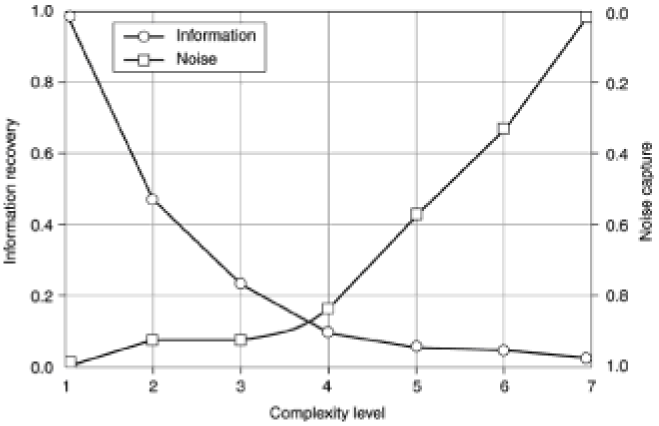


Figure 11.27 Information and noise map for the CREDIT data set.

The information capture curve and its interpretation are described above. Maximum complexity captures information uniquely defining each output state, so the curve starts at a level of 1 shown on the left scale. The noise curve starts at 1 too, but that is shown on the right scale. It indicates that the most complex model captures all of the noise present in the data set. This is very often the case for many data sets with the highest degree of complexity. Maximum complexity obviously captures all of the noise and often captures enough information to completely define the output signals within that specific data set.

At complexity level 2, the information capture curve has already fallen to about 0.5, showing that even a small step away from capturing all of the complexity in the data set loses much of the defining information about the output states. However, even though at this slightly reduced level of complexity much information about the output state is lost, the noise curve shows that any model still captures most of the noise! Noise capture falls from about 1.0 to about 0.95 (shown on the right scale). A model that captures most of the noise and little of the needed defining information is not going to be very accurate at predicting the output.

Complexity level 3 is even worse! The amount of noise captured is almost as much as before, which still amounts to almost all of the noise in the data set. While the amount of noise captured is still high, the amount of predictive information about the output has continued to fall precipitously! This is truly going to be one tough data set to get any decent model from!

By complexity level 4, the information capture curve shows that at this level of complexity, and on for all of the remaining levels too, there just isn’t much predictive information that can be extracted. The noise capture begins to diminish (the rising line indicates less noise), but even if there is less noise, there just isn’t much of the needed information that a relatively low-complexity model can capture.

By complexity level 7, although the noise capture is near 0 (right scale), the amount of information about the output is also near 0 (left scale).

No very accurate model is going to come of this. But if a model has to be built, what is the best level of complexity to use, and how good (or in this case, perhaps, bad) will that model be?

Optimal information capture points Given the noise and information characteristics at every complexity level shown in Figure 11.27, is it possible to determine how much noise-free information is actually available? Clearly the amount of noise-free information available isn't going to be much since the data set is so noisy. However, the curve in Figure 11.28 is interesting. At complexity level 1, to all intents and purposes, noise swamps any meaningful signal.

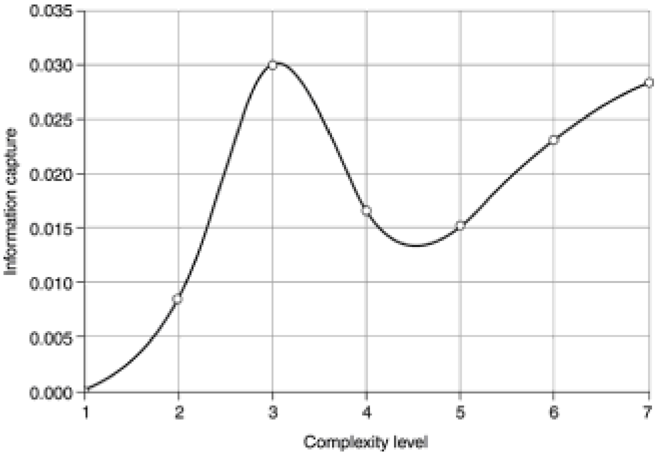


Figure 11.28 Information capture map showing the amount of noise-free information captured at different levels of complexity in the CREDIT data set.

Noise, of course, represents information patterns that are present in this specific data set, but not in any other data set or in the population. Since the noise map in the previous figure showed that perfect information about the output is available at level 1, for this specific data set the output can be perfectly learned. However, what the noise curve points out is that none, or essentially none, of the information relationships used to make these perfect predictions of the output state will be present in any other data set. So the noise map shows that there is almost no noise-free information available at level 1. (Although the graph does indeed appear to show 0 at level 1, it is in fact an infinitesimally small distance away from 0—so small that it is impossible to show graphically and is in any case of no practical use.)

By complexity level 3, the amount of noise-free information has risen to a maximum, although since the scale is in ratio entropy, it turns out to be precious little information! After that it falls a bit and rises back to nearly its previous level as the required model becomes less complex.

Unfortunately, Figure 11.28 has exaggerated the apparent amount of information capture by using a small scale to show the curve so that its features are more easily visible.

The maps shown and discussed so far were presented for ease of explanation. The most useful information map generally used in a survey combines the various maps just discussed into one composite map, as shown for this data set in Figure 11.29. This map and the state entropy map are the pair that a miner will mainly use to get an overview of the high-level information relationships in a data set. At first glance, Figure 11.29 may not appear so different from Figure 11.27, and indeed it is mainly the same. However, along the bottom is a low wavy line that represents the amount of available noise-free information. This is exactly the same curve that was examined in the last figure. Here it is shown to the same scale as the other curves. Clearly, there really isn't much noise-free information available in this data set! With so little information available, what should a miner do here? Give up? No, not at all!

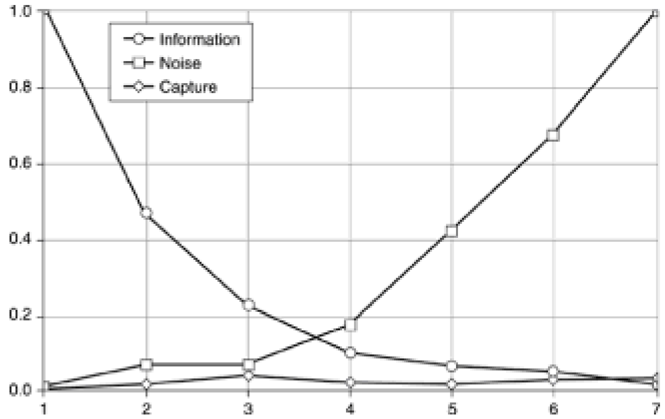


Figure 11.29 The information/noise/capture (INC) map is the easiest summary for a miner to work with. In this case it summarizes the information content, amount of noise captured, and noise-free information level into a single picture, and all at the same scale.

Recall that the original objective was to improve on a less than 1% response rate. The model doesn't seem to need much information to do that, and while there is little noise-free information available, perhaps noisy information will do. And in fact, of course, noisy information *will* do. Remember that the miner's job is *to solve a business problem*, not to build a perfect model! Using the survey and these maps allows a miner to (among other things) quickly estimate the chance that a model good enough to solve the business problem can actually be built. It may be surprising, but this map actually indicates that it very likely can be done.

Without going into the details, it is possible to estimate exactly how complex a model is needed to yield the best response for the problem at hand. It turns out that a model of complexity level 5.7 (approximately) is a good trade-off between speed, noise resistance, and improved accuracy for this application. Without regard to any other insights gained, or understanding of the data set that the survey yields, it can be determined that a model built to about a 5.7 level of complexity will capture enough information to make improved predictions of BUYER possible to a sufficient degree to have economic benefit.

When the model is built, it can be useful to see how much information has actually been captured. Surveying the modeled data *after* the model is built, together with the model predictions, can be used to measure how much information the model captured, if it has learned noise, and if so, how much noise—all useful information for the miner.

Not all information/noise/capture maps look like this one does. For comparison, Figure 11.30 shows a map for a different data set.

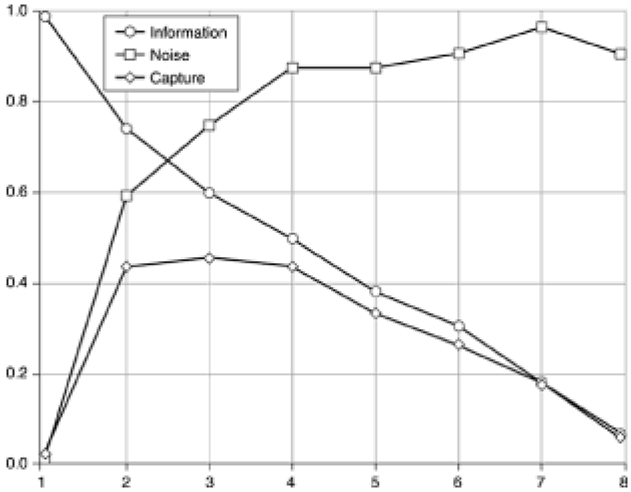


Figure 11.30 An INC map from a different data set. Far more noise-free information is available in this data set, and a complexity level 4 model looks to be a good choice. (Depending always, of course, on the exact nature of the business problem!)

State entropy map After having given so much attention to the complexity map, there is still the state entropy map for the CREDIT data set in Figure 11.31 that carries useful information. In spite of the apparent perfect predictions possible from the information enfolded in this data (shown in the information metrics **I**(X;Y) = 1 and cH(Y|X) = 0), the state entropy map tells a different tale. One of the two states has low entropy (uncertainty), the other high. It is only the minute proportion of uncertain states in the data set (less than 1%) that leads to the misleading entropic and mutual information measures shown in the information metrics. If the low cH(Y) isn't warning enough, the tilt shown in this map is a clear warning sign that further investigation is needed.