NAME

infot — calculate information theory measures

SYNOPSIS

infot -b [**-H**] [**-x** *regexp*] [*inputfile* ...]
infot -n [**-H**] [**-x** *regexp*] [*inputfile* ...]
infot -p [**-H**] [**-x** *regexp*] [*inputfile* ...]
infot -s [**-x** *regexp*] [*inputfile* ...]

DESCRIPTION

The **infot** command provides a general-purpose tool for measuring the probability relationships between user-selected data tokens. Given a specified input stream, **infot** can calculate one of several pertinent information-theoretic values. The values may be calculated with reference to an independent repertoire, or may be calculated as so-called "self-information."

In conjunction with other Humdrum tools (notably the **context** and **humsed** commands), **infot** permits sophisticated information-theoretic analyses to be carried out, including calculations of information flow, short-term conditional probabilities, and longer-term *m-dependency* analyses. Alternatively, a simple set of summary statistics can be requested. In most cases, users will want to use **infot** to generate outputs that are suitable for further processing.

Input to **infot** is restricted to a single spine. However, the input data tokens may contain multiple-stops representing complex contextual information (such as produced by the **context** command).

For the entire input, **infot** tabulates the total number of occurrences of each unique data record (hereafter referred to as 'states'). For the **-n, -p** and **-b** options, **infot** outputs a two-column list where the left column indentifies each unique state and the right column provides one of several corresponding calculated measures. With the **-n** option, this measure is merely an integer count of the number of occurrences of each corresponding state. With the **-p** option, this measure is a probability of occurrence for each state. With the **-b** option, this measure identifies the information content for the corresponding state in bits.

Information content (H) in bits is calculated according to the classic equation devised by Shannon and Weaver (see REFERENCES):

$$H = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}$$

where H is the average information (in bits), N is the number of possible unique states in the

repertoire, and p_i is the probability of occurrence of state *i* from the repertoire.

Note that the outputs produced by **infot** do not conform to the Humdrum syntax.

OPTIONS

The **infot** command provides the following options:

-b	output information (in bits) for each unique data token
-h	displays a help screen summarizing the command syntax
-H	format output as **humsed** commands
-n	output frequency count for each unique data token
-p	output probability value for each unique data token
-s	output information-related summary statistics
-x *regexp*	exclude tokens matching *regexp* from calculations

Options are specified in the command line.

With the **-n** option, **infot** outputs a two-column list where the left column indentifies each unique state present in the input, and the right column provides an integer *count* indicating the number of occurrences for the corresponding state.

With the **-p** option, **infot** outputs a two-column list where *probabilities* of occurrence are output in the right-hand column, rather than counts.

With the **-b** option, **infot** outputs the information (in bits) as calculated according to the Shannon-Weaver equation.

EXAMPLES

The use of **infot** is illustrated in the following examples. Consider the following input:

```
**foo
A
B2
C-c
A
B2
A
A
B2
C-c
A
A
X Y
*-
```

A simple command invocation would use the **-n** option to count the number of occurrences of each unique data token (or state):

```
infot -n input
```

The corresponding output is:

```
A       6
B2      3
C-c     2
X Y     1
```

The tallies indicate that state 'A' occurs 6 times, and that the least common state ('X Y') occurs just once. If we had invoked the **-p** option, the counts would be replaced by probabilities. The command:

```
infot -p input
```

produces the following output:

```
A       0.500
B2      0.250
C-c     0.167
X Y     0.083
```

Alternatively, the **-b** option:

```
infot -b input
```

would output information measures for each state, in bits:

```
A       1.000
B2      2.000
C-c     2.585
X Y     3.585
```

In the case of the **-s** option, summary statistics would be output, rather than a two-column list. For the above input, the following summary statistics would be generated:

```
Total number of input states in message:   4
Total information of message  (in bits):   20.7549
Total possible  information for message:   24
Info  per state  for equi-prob  distrib:   2
Average information conveyed  per state:   1.72957
Percent redundancy  evident  in message:   13.5213
```

The first line of output merely indicates the number of unique states found in the input (in this case just 4). The fifth output line indicates the average information conveyed per state (in bits). The fourth output line indicates the theoretical maximum average information per state that could be communicated by a system having four states. The third line indicates the maximum possible information that could be communicated in a message of the same length as the input — given the theoretical maximum average information. (Since there are 12 data records, this value is simply 12×2 bits, or 24 bits.) The second output line gives the actual total information for the given input message. (This is always less-than, or equal-to the maximum theoretical value.) The final line indicates the amount of redundancy — as a

percentage. That is, this value contrasts the actual information conveyed with the theoretical maximum.

In general, note that as the probabilities of the input states approach equivalence, the redundancy approaches zero and the average information content approaches the theoretical maximum.

Consider now an example where a large number of messages from a repertoire (dubbed `repertoire`) is passed to **infot**:

```
infot -b repertoire
```

Suppose that the following output is produced:

```
ABC     3.124
BAC     1.306
C C D   1.950
X       5.075
XYZ     19.334
```

This result indicates that, although there might have been hundreds of data tokens processed in the repertoire, only five different unique states were present. The greatest information content (lowest probability) is associated with the state `XYZ` (19.334 bits), whereas the lowest information content (highest probability) is associated with the state `BAC` (1.306 bits). Notice that the multiple-stop `C C D` is treated as a single state.

Now imagine we had another message presumed to belong to the same repertoire as our input. We would like to trace how the information increases and decreases over the course of this new 'message'. This goal involves a two-part operation. First, we re-invoke **infot** adding the **-H** option, and redirect the output to a file `replace`:

```
infot -bH repertoire > replace
```

This causes **infot** to produce as output a set of **humsed** commands. Given the identical `repertoire` input, the following output is sent to the file `replace`:

```
s/^ABC$/3.124/g; s/^ABC    /3.124/g; s/     ABC$/3.124/g; s/ ABC     /3.124/g
s/^BAC$/1.306/g; s/^BAC    /1.306/g; s/     BAC$/1.306/g; s/ BAC     /1.306/g
s/^C C D$/1.95/g; s/^C C D /1.95/g; s/      C C D$/1.95/g; s/ C C D   /1.95/g
s/^X$/5.075/g; s/^X        /5.075/g; s/     X$/5.075/g; s/    X      /5.075/g
s/^XYZ$/19.334/g; s/^XYZ   /19.334/g; s/    XYZ$/19.334/g; s/ XYZ     /19.334/g
```

Although these commands may appear somewhat cryptic, they merely instruct the Humdrum stream editor (**humsed**) to replace all occurrences of the five data tokens (in any input file) by the corresponding numerical values — in this case, values that represent the number of bits of information.

The following file (called `input`) contains the message of interest:

```
**bar
BAC
BAC
C C D
.
=
*
C C D
XYZ
X
ABC
BAC
*-
```

This file can be transformed so that the data tokens are replaced by corresponding information values as determined from the original repertoire. This is done by invoking the **humsed** command, and providing it with the substitution commands held in the file `replace`:

```
humsed -f replace input > output
```

The resulting output file would be as follows:

```
**bar
1.306
1.306
1.950
.
=
*
1.950
19.334
5.075
3.124
1.306
*-
```

Note that data tokens in message that do not appear in the probability list (such as the equals-signs) remain unmodified.

Several interpretations may be made about this message. For example, the above passage appears to show a pattern of initially low information that increases and then decreases toward the end of the passage. This suggests that the beginning and ending of this passage are more highly constrained or stereotypic than the middle part of the passage.

Summing together the individual information values for this passage, the total message conveys 35.35 bits. For five states, the maximum average information is 2.322 bits per state, and so the expected maximum for a message consisting of 8 items would be 8×2.322 or 18.58 bits. This suggests that this message is considerably less banal, (less redundant or

more unique) than a typical message from the original repertoire. In particular, the occurrence of the state 'XYZ' has a low probability of occurrence — and is likely to be a distinctive feature of this passage.

In the above examples, only simple (zeroth-order) probabilities have been examined. Higher-order and *m*-dependency probabilities may be measured by reformulating the input using the **context** command.

PORTABILITY

DOS 2.0 and up, with the MKS Toolkit. OS/2 with the MKS Toolkit. UNIX systems supporting the *Korn* shell or *Bourne* shell command interpreters, and revised *awk* (1985).

SEE ALSO

context (4), **humsed** (4), **patt** (4), **pattern** (4), **simil** (4)

REFERENCES

Abraham Moles, *Information Theory and Esthetic Perception*, Urbana: University of Illinois Press, 1968.

Shannon, C. E., & Weaver, W. *The Mathematical Theory of Communication*. Urbana: University of Illinois Press, 1949.

Wong, A. K. C., & Ghahraman, D. A statistical analysis of interdependence in character sequences. *Information Sciences*, Vol. 8 (1975) pp. 173-188.