Calculating conditional complexities from the web

October 2, 2009

1 Plain simple conditional complexity

Let's say we have a goal of calculating a conditional complexity like $C('rider'|'horse')$. The NGD paper allows to do just that, in the following manner.

We can calculate the conditional complexity using the general formula $K(y|x) = K(x,y) - K(x)$; in our case, it comes down to $C('rider'|'horse') - C('horse')$. How is this done with Google?

First let's see how can a search engine be a compressor. A compressor is a function whose domain is a prefix code. If the search engine can return the number of hits for single terms x, y and for the combined $x \cap y$ (that is, the number of pages containing both terms), given the total number of documents M, we can calculate the probability of occurrence of either terms or their combination. This probability can be normalized to sum to 1, which allows us to infer the underlying information content of a corresponding prefix-code. In the sense of a function that generates a prefix code, a search engine can be seen as a compressor and we can apply the formula described above in order to calculate distances between words.

The number $|x \cap y|/M$ cannot be interpreted as the probability $Pr(x \cap y)$ because, as search terms overlap, the summed probabilities over the entire set of terms S is bigger than 1. In order to bring this probability to 1, we use N, the total number of hits obtained by trying every combination of x and y (including those with $x = y$). In this case, $N = \sum_{x,y \in S} |x \cap y|$. Defining the $g(x) = g(x,x)$ and $g(x,y) = |x \cap y|/N$, we get a probability density distribution which sums up to 1. The implied prefix code is $G(x) = -\log(g(x))$ which can be seen a search engine compressor.

In practice, the paper claims it is not very important to know the exact N, it is possible to replace it with M (the total number of documents in the corpus indexed by the search engine). So conditional complexity is $G(y|x) = G(x,y) - G(x) = \log(|x|) - \log(|x \cap y|$ when using a search engine. $|x|$ is the number of hits when searching for term x.

2 The classic distances

How does this compare to the distances proposed by the Bennett and Cilibrasi & Vitanyi ? Basically, information distance is described by Bennett as $E(x,y) = \max\{K(x|y), K(y|x)\} = K(x,y) - \min\{K(x), K(y)\}$. A further, normalization step, divides $E(x,y)$ by $\max\{K(x), K(y)\}$. If instead of K we use an existing compressor C, we get the normalized compression distance :

$$NCD(x,y) = \frac{\max\{C(x|y^*), C(y|x^*)\}}{\max\{C(x), C(y)\}} \quad (1)$$

$$= \frac{C(xy) - \min\{(C(x), C(y)\}}{\max\{C(x), C(y)\}} \quad (2)$$

The NGD formula is obtained by simply replacing C with G and further expliciting G into the number of search engine hits:

$$NGD(x,y) = \frac{G(x \cap y) - \min\{(G(x), G(y)\}}{\max\{G(x), G(y)\}} \quad (3)$$

$$= \frac{\log \max\{f_x, f_y\} - \log f_{xy}}{\log N - \log \min\{f_x, f_y\}} \quad (4)$$

with f_x, f_y and f_{xy}, the number of hits for x, y, and $x \cap y$.