Louis Eisenberg
10-8-04

CS 276A: Review session for problem set #1

Gamma codes

To encode positive integer x, find
base_power = floor($\log_2 x$)
base = 2^base_power
offset = x – base
gamma(x) = <base_power in unary, offset in binary using base_power digits).

Example: x = 85.
base_power = floor($\log_2 85$) = 6
base = $2^6 = 64$
offset = 85 – 64 = 21
gamma(85) = <1111110, 010101>

Wildcards

Permuterm index: maintain "inverted" index from all rotations to corresponding dictionary terms. This way we can always push the wildcard to the end of the query string, find the matching rotations, then find the matching dictionary terms. This match is necessary – but is it sufficient? Before we look at the postings lists of these terms, do we need to post-filter? What about words that are rotations of each other, e.g. "arch" and "char"?

Consider "arch" and "char":

arch$, rch$a, char, harc, $arch
char$, har$c, arch, rcha, $char

Their permuterm sets are actually disjoint. So we shouldn't get any false positives on single-wildcard queries.

What about multiple wildcards, e.g. X*Y*Z?

Search for X*Y and Y*Z → look up Y$X* and Z$Y*?

E.g. we're looking for c*h*r → look up h$c* and r$h*?

No – have to search for *Y* and X*Z (look up Y* and Z$X*) and then filter.

Zipfian distributions etc.

We often say that the ith term in a corpus has frequency proportional to f(i). E.g. in the case of a Zipfian distribution, f(i) = 1/i.

How do we use this to find the actual frequency of the ith term?

If the ith term, w_i, has frequency proportional to f(i), then the probability that a random word x that we pick from the corpus will be w_i is $P(x = w_i) = C*f(i)$, where C is some constant.

We can solve for C: as with any probability distribution, the sum of the probabilities of all possible events has to equal one. The sum of the frequencies of all the terms in the lexicon must be one. (Note that "frequency" can mean either "number of occurrences" or "number of occurrences divided by total number of terms in corpus." I'm using the latter sense here.)

So if we have m distinct terms in our corpus, we get:

$$\sum_{i=1}^{m} P(x = w_i) = \sum_{i=1}^{m} C \cdot f(i) = 1, \text{ which means } C = \frac{1}{\sum_{i=1}^{m} f(i)}$$

A similar strategy will help you solve #5 on the problem set.

Also remember that when it's a Zipfian distribution, we can use the harmonic approximation:

$$\sum_{i=1}^{m} 1/i = H_m \approx \ln m.$$

To solve these sums exactly – I have no idea how you would do it by hand. But you can use a TI-92 or write a few lines of code.