Maintaining Discriminatory Power in Quantized Indexes

Matt Crane, Andrew Trotman, Richard O'Keefe
Department of Computer Science, University of Otago, New Zealand

Introduction

What is quantization?

Quantization is the pre-calculation of retrieval scores for every term in every document for a given ranking function. These scores are then quantized to an integer value within a given range for compressibility reasons.

Why do we quantize?

For several reasons, including better compressibility — integer scores allow impact-ordered postings with denser buckets, leading to better compression, and score-at-a-time processing.

The quantization reduces the ranking function at search time to a series of integer additions, which is much faster than the floating point.

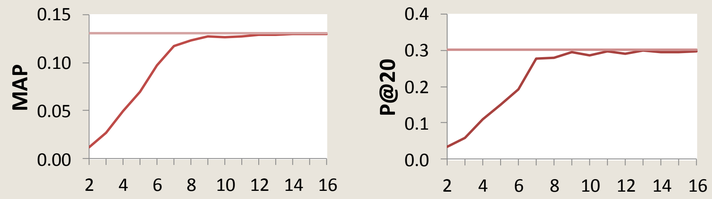
With the appropriate integer range, there is no loss in precision with this approximation. Which leads us to ask:

What effect does this range (expressed in bits) have on precision, search latency, and index size?

Results shown are for ClueWeb09 Category A with 70% spam removed using the TREC 2011 query set. Experiments were conducted across 5 TREC collections of varying sizes and 8 TREC query sets.

Precision

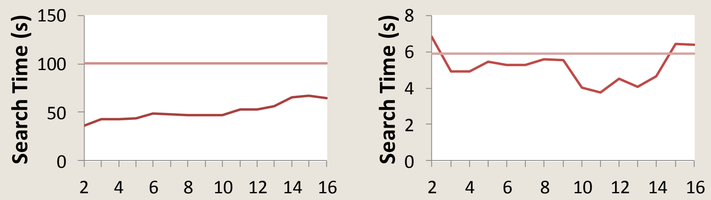
It is important that we don't lose precision when we perform quantization. We tested the precision as the number of bits increased, and compare this to the precision of a tf index (horizontal line).



Both MAP and P@20 show similar results. Visual inspection suggests 7 bits to be enough for either metric.

Latency

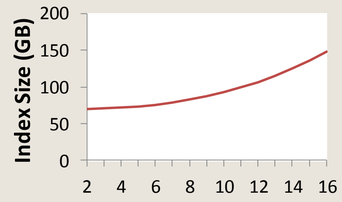
The effect the quantization has on search latency is also important. We tested a term-at-a-time approach, comparing both search to completion (left) and top-k (k=20, right) using a heap.



For search to completion, the time to search increases with the bits. With top-k there is a trade-off between heap operations and numbers touched.

Size

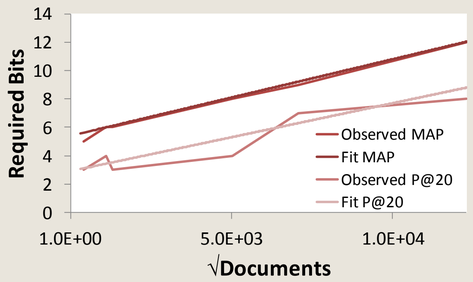
As the bits increase fewer documents share an impact value, lowering postings compression, which combined with the extra impact values, gives an increased index size.



Ideality

Ideally we wish to minimise search latency and index size without adversely affecting precision.

For each collection we identified the smallest number of bits that showed not statistically significant precision differences with a tf index.



A line was fitted to the results gathered from MAP (as this metric is more stable). This yields an equation to calculate the ideal number of bits:

$$b = \left\lceil g + h * 10^{-4} \sqrt{|D|} \right\rceil$$

with g=h=5.4 for MAP and g=2.9, h=4.3 for P@20.