Hashing as Tie-Aware Learning to Rank
Kun He, Fatih Cakir, Sarah Adel Bargal, Stan Sclaroff
Computer Science, Boston University

• Optimization: combinatorial, NP-hard (search over all bit combinations)
• To apply SGD: continuous relaxation!
• Main trick: reverse "midpoint rule": finite sums → continuous integrals

\[ AP_r(H^d) = \frac{c_d ^{d - 1} - \frac{c_d (c_d - 1)}{N^c (c_d - 1)}}{N^c (c_d - 1)} = \frac{c_d (C_d - 1)}{C_d - 1} \ln \frac{C_d}{C_d - 1} \]

\[ DCG_r(H^d) = \ln 2 \sum_{c \in C} \frac{G(v_{c, d})}{c_d} \frac{c_d}{c_{d - 1}} \ln \frac{1}{c_{d - 1}} \]

• End-to-end optimization (closed-form gradients)

\[ \text{How should we break the ties?} \]

Query \( d = 0 \) \( d = 1 \) \( d = 2 \) \( AP = 0.92 \)

\[ \text{AP = 0.59} \]

\[ \text{AP = 0.81} \]

• Tie-aware ranking metrics [1]: average over all permutations of tied items, in closed-form

\[ \text{Tie-Breaking Matters} \]

• No tie-awareness in training: optimization objective unclear!

\[ \text{Key operation: Histogram binning} [2] \]

\[ \text{References} \]


http://github.com/kunhe/TALR