Complexity analysis for Cortana's numeric strategies

Marvin Meeng

This document presents a complexity analysis for Cortana's numeric strategies, for exhaustive depth-first search and heuristic level-wise beam search. Cortana offers four options for its '*numeric strategy*' parameter: *all*, *best*, *bins*, and *bestbins*. These are actually combinations of the Granularity options fine and coarse, and Selection Strategy options all and best, presented elsewhere[1]. The table below shows a summary of the analysis. More detailed analyses, and a list of symbols, can be found thereafter. The summary shows that, when used in a beam setting, the numeric strategies *all* and *best* have the same worst-case complexity. The same is true for *bins* and *bestbins*.

So, when performing experiments in a beam setting, complexity arguments become irrelevant when considering *all* and *best*, or *bins* and *bestbins*. This leaves only a saturation argument as justification for the use of *best* and *bestbins*.

cortana	granularity	selection strategy	depth-first	level-wise beam
all	fine	all	$\mathcal{O}(U^D)$	$\mathcal{O}(U(D\omega - \omega + 1))$
best	fine	best	$\mathcal{O}(U\beta^{D\text{-}1})$	$\mathcal{O}(U(D\omega - \omega + 1))$
bins	coarse	all	$\mathcal{O}({U_c}^D)$	$\mathcal{O}(U_c(D\omega - \omega + 1))$
bestbins	coarse	best	$\mathcal{O}(U_c\beta^{D\text{-}1})$	$\mathcal{O}(U_c(D\omega - \omega + 1))$

Introduction

Below, a complexity analysis is first performed for the Selection Strategy and Granularity options separately. Thereafter, the analysis for the options offered by Cortana is presented. The table below lists the symbols used. For the heuristics Selection Strategy best, Granularity coarse, and all beam searches, the number of candidates tested on a certain depth-level differs from the number of candidates that is retained for the next level. The number of candidates tested at each depth level is given by 'at d*'. For Selection Strategy all+depth-first, and Granularity fine+depth-first, 'D=*' show the *total* number of candidates for a search up to and including depth-level D.

symbol	description
a	description attribute
c	number of cut points (number of bins-1)
d	depth-level
m	number of description attributes
D	maximum search depth
A_i	cardinality of a_i
A_i^c	coarse cardinality of a_i($A_i$for nominal, c for numeric)
o_i	number of operators for a_i(usually 1 (=) or 2 ($\leq, \geq$)
U	$\sum_{i=1}^{m} (o_i \times A_i)$, number of (single conjunct) descriptions
U_c	$\sum_{i=1}^{m} (o_i \times A_i^c)$, number of (single conjunct) descriptions for coarse
ω	beam width
β	$\sum_{i=1}^{m} (o_i)$, total number of *selected* candidates for best (o_i per a_i)

NOTE: U U can be larger than the set of (valid) depth-1 subgroups. U includes descriptions that cover more records than might be allowed for a subgroup because of a maximum coverage constraint. On the other hand, all descriptions that cover less records than is required by a minimum coverage constraint, should be purged from U beforehand. The same holds for U_c.

[1] Meeng and Knobbe; A Systematic Analysis of Strategies for Dealing with Numeric Data in Subgroup Discovery.

Selection Strategy

Summary Selection Strategy options all and best have the same computational complexity with respect to the number of refinements that are tested for a *single* candidate. This is because to determine the best refinement, all refinements need to be tested. In an exhaustive search, option all would be computationally much more demanding than option best, as at higher depths all refinements will be considered for further processing. In contrast, the complexity for beam search is essentially the same for the two options, irrespective of search depth. So, when performing a beam search, the best heuristic does not reduce the search space, but is likely to perform worse, as it selects only one candidate per attribute. Therefore, its use would be hard to justify.

selection strategy	depth-first	level-wise beam
all	D=1: U	at d1: U
	D=2: UU	at d2: $U\omega$
	D=3: UUU	at d3: $U\omega$
		$\mathcal{O}(U + \sum_{d=2}^{D} (U\omega))$
		$\mathcal{O}(U + D\text{-}1(U\omega))$
	$\mathcal{O}(U^D)$	$\mathcal{O}(U(D\omega{-}\omega{+}1))$
best	at d1: U	at d1: U
	at d2: $U\beta$	at d2: $U\times \min(\beta,\, \omega)$
	at d3: $U\beta^2$	at d3: $U\times \min(\beta^2,\, \omega)$
	$\mathcal{O}(\sum_{d=1}^{D} (U\beta^{\text{d-1}}))$	$\mathcal{O}(\sum_{d=1}^{D} (U\times \min(\beta^{\text{d-1}},\, \omega)))$
	$\mathcal{O}(U\beta^{D\text{-}1})$	$\mathcal{O}(U(D\omega{-}\omega{+}1))$

best+beam & all+beam For beam searches, best has the same worst-case complexity as all. When β is large, $\beta \geq \omega$, only the top-ω of the β candidates will available at the next search level. This situation occurs when the number of attributes is large, or the beam is small.

best+beam For beam searches, when β is small, the number of selected candidates up to a certain depth-level might be smaller than the beam size ($\beta^{\text{d-1}} < \omega$). In this case, less then ω candidates will be available. This situation occurs when the number of attributes is small, or the beam is large, and it continues until $\beta^{\text{d-1}} \geq \omega$. With increasing search depth, it becomes ever more unlikely, as $\beta^{\text{d-1}}$ grows fast.

best+beam & best+depth-first Related to the situation described above, when β is small, best+depth-first and best+beam have the same complexity. This situation occurs when the number of attributes is small, or the beam is large, and it continues until $\beta^{\text{d-1}} \geq \omega$.

all The situation where $U^{d-1} < \omega$ is unlikely and ignored, reasoning would be as above.

Granularity

Summary Granularity option fine and Selection Strategy option all have the same complexity. The analysis for coarse is proceeds the same as for the previous analyses. The only difference is the use of the reduced set of base conditions, or single conjunct descriptions, U_c instead of U.

pre-discretisation and dynamic discretisation For fine and coarse, the complexity for pre-discretisation and dynamic discretisation is identical. For fine this is obvious, so consider coarse. With pre-discretisation, the search-phase domain of an attribute consists of c values of the original domain of the attribute, and is determined beforehand. Every candidate, at every search depth, then uses all relevant of the c values of this search-phase domain. With dynamic discretisation, the search-phase domain consists of all original values, and from these, each candidate selects the c values that are most suitable for it. So the *number* of values is identical for the two scenarios.

coarse The analysis below ignores the possibility ${U_c}^{d-1} < \omega$, it is unlikely, reasoning is as before.

granularity	depth-first	level-wise beam
fine	D=1: U	at d1: U
	D=2: UU	at d2: $U\omega$
	D=3: UUU	at d3: $U\omega$
		$\mathcal{O}(U + \sum_{d=2}^{D} (U\omega))$
		$\mathcal{O}(U + D\text{-}1(U\omega))$
	$\mathcal{O}(U^D)$	$\mathcal{O}(U(D\omega - \omega + 1))$
coarse	D=1: U_c	at d1: U_c
	D=2: U_cU_c	at d2: $U_c\omega$
	D=3: $U_cU_cU_c$	at d3: $U_c\omega$
		$\mathcal{O}(U_c + \sum_{d=2}^{D} (U_c\omega))$
		$\mathcal{O}(U_c + D\text{-}1(U_c\omega))$
	$\mathcal{O}({U_c}^D)$	$\mathcal{O}(U_c(D\omega - \omega + 1))$

Cortana's numeric strategies: all, best, bins, bestbins

Summary Results for *all* and *best* are the same as those for Selection Strategy all and best, and use the base set U. So here, *all* and *best* have the same complexity in the beam setting. Results for *bins* and *bestbins* are based on Granularity option coarse, and use the base set U_c. In fact, *bins* is the same as coarse. Obviously, *bestbins* is a mixture of *best* and *bins*, or Selection Strategy option best and Granularity option coarse. Worst-case complexity for *bins* en *bestbins* is identical in the beam setting.

numeric strategy	depth-first	level-wise beam
all	D=1: U	at d1: U
	D=2: UU	at d2: $U\omega$
	D=3: UUU	at d3: $U\omega$
		$\mathcal{O}(U + \sum_{d=2}^{D}(U\omega))$
		$\mathcal{O}(U + D\text{-}1(U\omega))$
	$\mathcal{O}(U^D)$	$\mathcal{O}(U(D\omega - \omega + 1))$
best	at d1: U	at d1: U
	at d2: $U\beta$	at d2: $U\times \min(\beta, \omega)$
	at d3: $U\beta^2$	at d3: $U\times \min(\beta^2, \omega)$
	$\mathcal{O}(\sum_{d=1}^{D}(U\beta^{\text{d-1}}))$	$\mathcal{O}(\sum_{d=1}^{D}(U\times \min(\beta^{\text{d-1}}, \omega)))$
	$\mathcal{O}(U\beta^{D\text{-}1})$	$\mathcal{O}(U(D\omega - \omega + 1))$
bins	D=1: U_c	at d1: U_c
	D=2: U_cU_c	at d2: $U_c\omega$
	D=3: $U_cU_cU_c$	at d3: $U_c\omega$
		$\mathcal{O}(U_c + \sum_{d=2}^{D}(U_c\omega))$
		$\mathcal{O}(U_c + D\text{-}1(U_c\omega))$
	$\mathcal{O}({U_c}^D)$	$\mathcal{O}(U_c(D\omega - \omega + 1))$
bestbins	at d1: U_c	at d1: U_c
	at d2: $U_c\beta$	at d2: $U_c\times \min(\beta, \omega)$
	at d3: $U_c\beta^2$	at d2: $U_c\times \min(\beta^2, \omega)$
	$\mathcal{O}(\sum_{d=1}^{D}(U_c\beta^{\text{d-1}}))$	$\mathcal{O}(\sum_{d=1}^{D}(U_c\times \min(\beta^{\text{d-1}}, \omega)))$
	$\mathcal{O}(U_c\beta^{D\text{-}1})$	$\mathcal{O}(U_c(D\omega - \omega + 1))$