Honey, I Shrunk the Cube

Matteo Golfarelli
Stefano Rizzi

University of Bologna - Italy

Summary

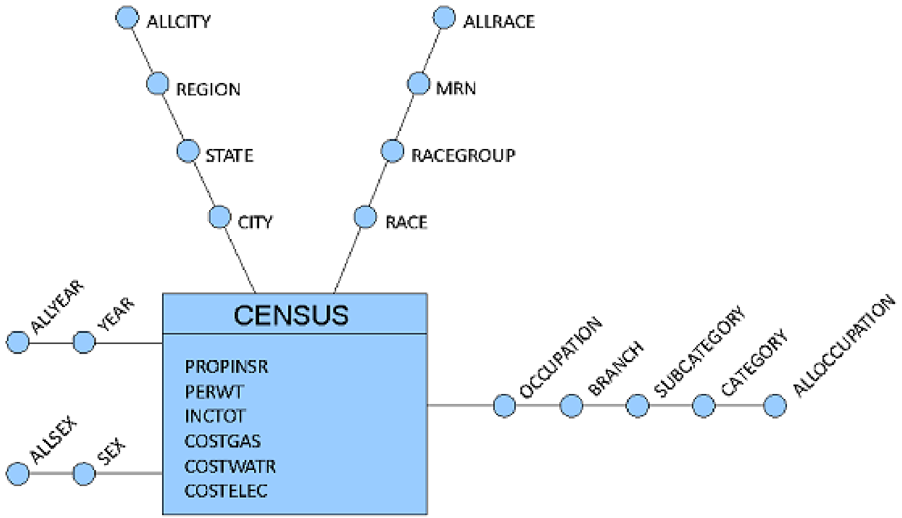
- Motivating scenario
- The shrink approach
- A Heuristic algorithm for shrinking
- Experimental results
- Summary and future work

DW & OLAP Analysis

- OLAP is the main paradigm for querying multidimensional databases

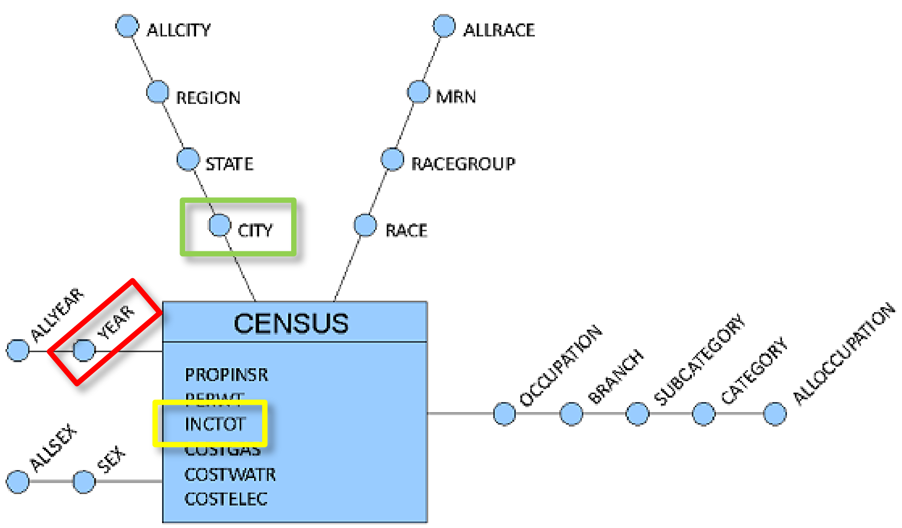
DW & OLAP Analysis

- OLAP is the main paradigm for querying multidimensional databases



DW & OLAP Analysis

- OLAP is the main paradigm for querying multidimensional databases
  - ✓ An OLAP query asks for returning the values of one or more numerical measures, grouped by a given set of analysis attributes

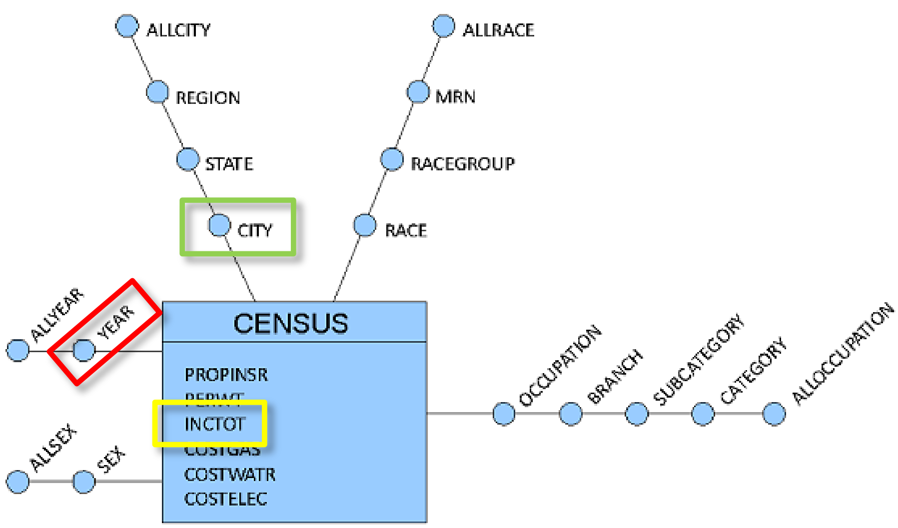


Average income in 2013 for each city

thousands of tuples in the resultset!!

DW & OLAP Analysis

- OLAP is the main paradigm for querying multidimensional databases
  - ✓ An OLAP query asks for returning the values of one or more numerical measures, grouped by a given set of analysis attributes
  - ✓ An OLAP analysis is typically composed by a sequence of queries (called session). Each obtained by transforming the previous one through the application of an OLAP operation

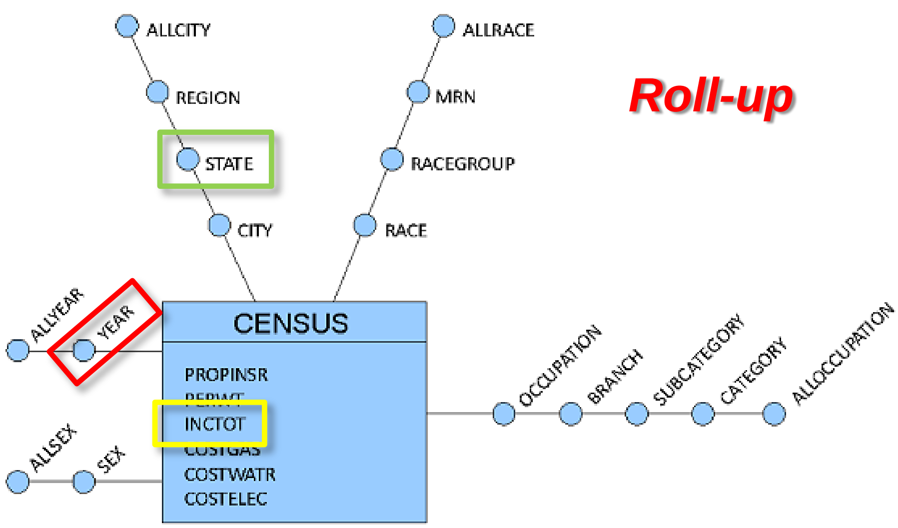


Average income in 2013 for each city

thousands of tuples in the resultset!!

DW & OLAP Analysis

- OLAP is the main paradigm for querying multidimensional databases
  - ✓ An OLAP query asks for returning the values of one or more numerical measures, grouped by a given set of analysis attributes
  - ✓ An OLAP analysis is typically composed by a sequence of queries (called session), each obtained by transforming the previous one through the application of an OLAP operation



Roll-up

Average income in 2013 for each state

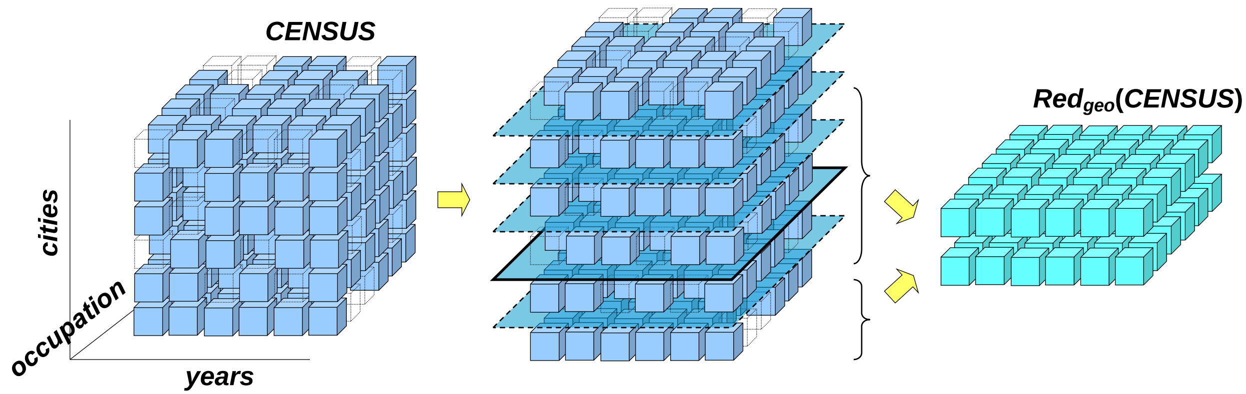
50 tuples in the resultset

Information flooding

- One of the problems that affect OLAP explorations is the risk the size of the returned data compromises their exploitation
  - ✓ more detail gives more information, but at the risk of missing the overall picture, while focusing on general trends of data may prevent users from observing specific small-scale phenomena
- Many approaches have been devised to cope with this problem:
  - ✓ Query personalization
  - ✓ Intensional query answering
  - ✓ Approximate query answering
  - ✓ OLAM On-Line Analytical Mining
- The shrink operator falls in the OLAM category
  - ✓ it is based on a clustering approach
  - ✓ **it can be applied to the cube resulting from an OLAP query to decrease its size while controlling the loss in precision**

The Shrink intuition

- The cube is seen as a set of slices, each slice corresponds to a value of the finest attribute of the shrinked hierarchy



- The slices are partitioned into a number of clusters, and all the slices in each cluster are fused into a single, approximate *f-slice* (reduction) by averaging their non-null measure values

AVG

City	2010	2011	2012
Miami	47	45	50
Orlando	44	43	52
Tampa	39	50	41
Washington	47	45	51
Richmond	43	46	49
Arlington	—	47	52

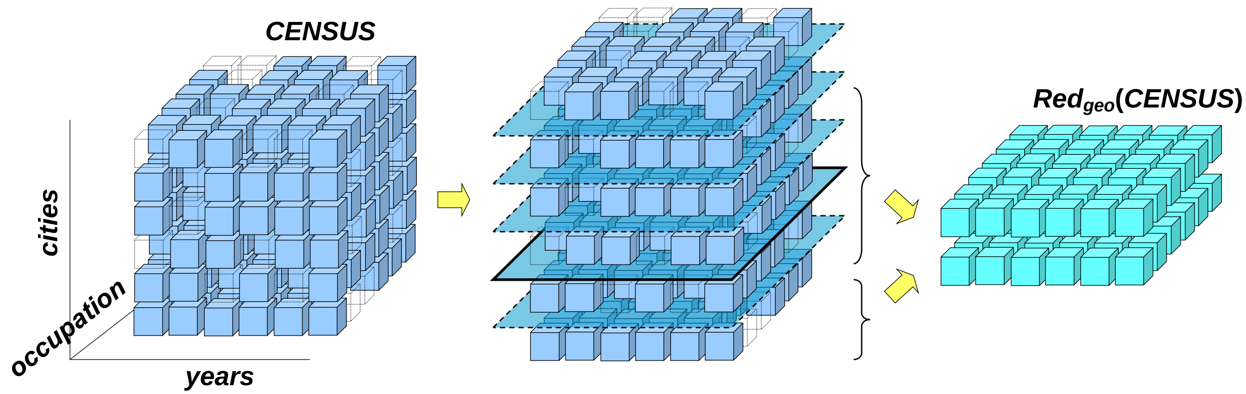
City	2010	2011	2012
Miami, Orlando	45.5	44	51
Tampa	39	50	41
Virginia	45	46	50.6

	2010	2011	2012
South-Atlantic	44	46	49.2

AVG

The Shrink intuition

- The cube is seen as a set of slices, each slice corresponds to a value of the finest attribute of the shrinked hierarchy



- The slices … in each cluster are fused … heir non-null measure v…

At each step the clusters to be merged must:

- **Minimize the approximation error (SSE)**
- **Respect the hierarchy structure**

Year

City	2010	2011	2012
Miami	47	45	50
Orlando	44	43	52
Tampa	39	50	41
Washington	47	45	51
Richmond	43	46	49
Arlington	—	47	52

City	2010	2011	2012
Miami, Orlando	45.5	44	51
Tampa	39	50	41
Virginia	45	46	50.6

Year

	2010	2011	2012
South-Atlantic	44	46	49.2

AVG

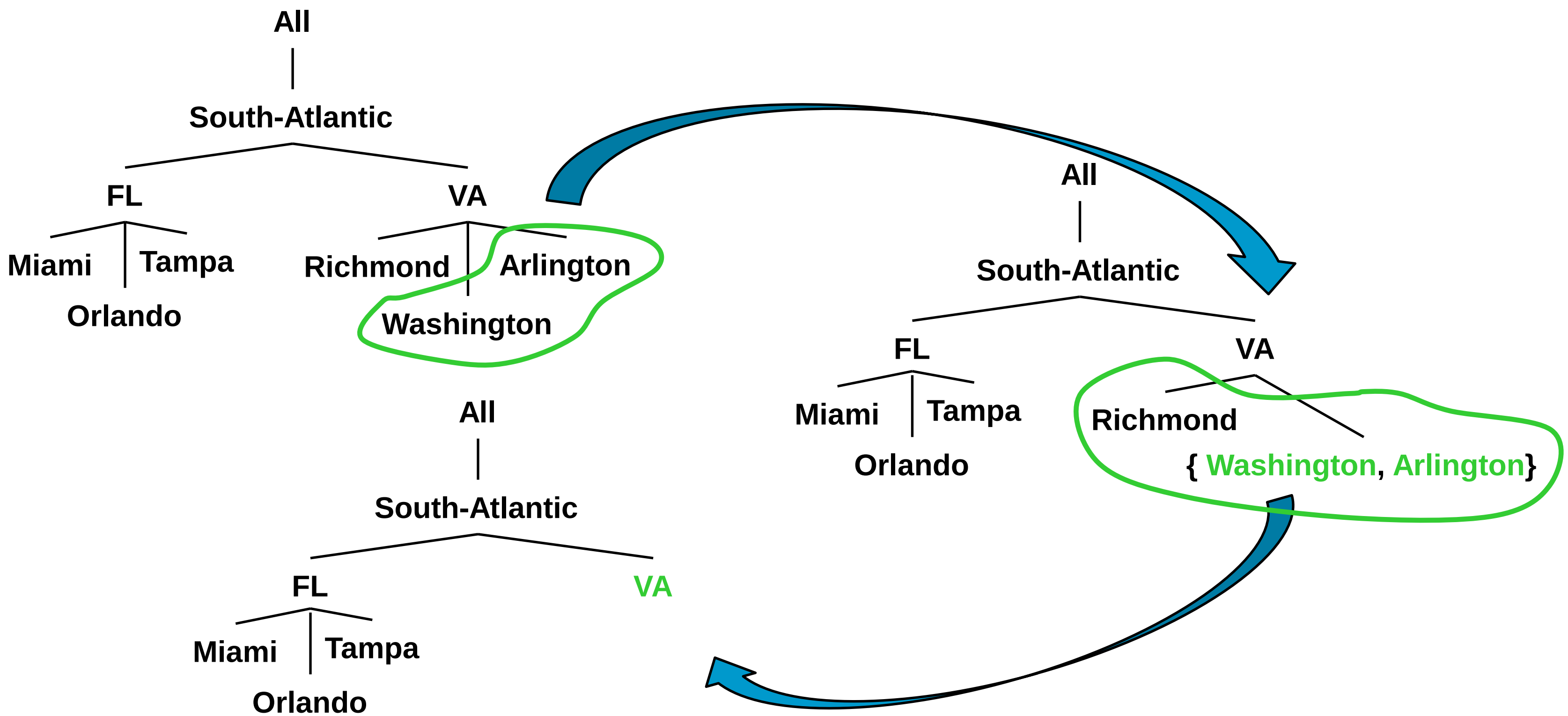
Shrink vs Roll-Up

- A roll-up operation:
  - ✓ reduces the size of the pivot table based on the hierarchy structure only
  - ✓ the level of detail is changed for all the attribute values at the same time
  - ✓ the size of the result depends on the attribute granularity and is not tuned by the user

- A shrink operation:
  - ✓ reduces the size of the pivot table considering the information carried by each slice while preserving the hierarchy structure
  - ✓ the level of detail of the result is changed only for specific attribute values
  - ✓ the size of the result is under the user control

The hierarchy constraints

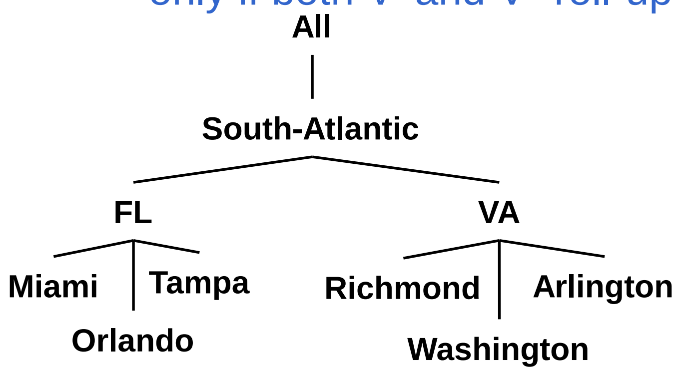
- To preserve the semantics of hierarchies in the reduction, the clustering of the f-slices at each fusion step must meet some further constraints besides disjointness and completeness:
  - ✓ Two slices corresponding to values V' and V'' can be fused in a single f-slice only if both V' and V'' roll-up to the same value of the ancestor attribute

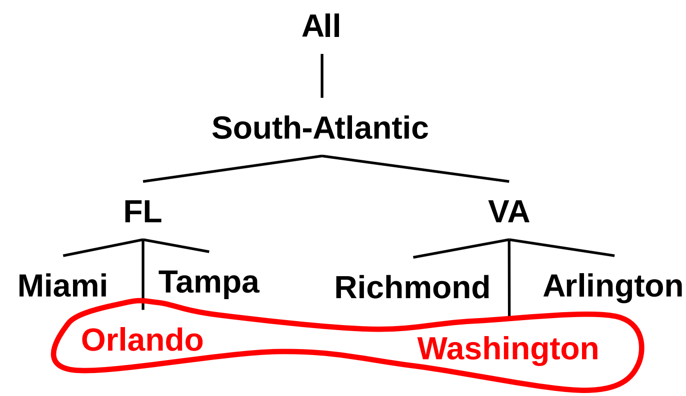


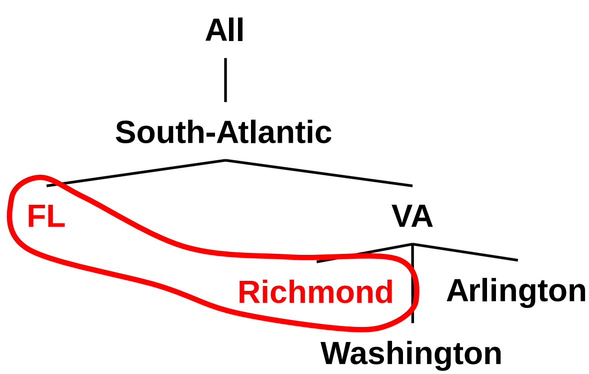
  - ✓ When a slice includes all the descendants of a given value, it is represented by that value

The hierarchy constraints

- To preserve the semantics of hierarchies in the reduction, the clustering of the f-slices at each fusion step must meet some further constraints besides disjointness and completeness:
  - ✓ Two slices corresponding to values V' and V'' can be fused in a single f-slice only if both V' and V'' roll-up to the same value of the ancestor attribute







The approximation error

- The SSE of a reduction can be incrementally computed
  - ✓ The SSE of a slice V obtained merging two slices V' and V'' can be computed from the SSEs of the slices to be merged as follows:

$$SSE\left(F^{V' \cup V''}\right) = SSE\left(F^{V'}\right) + SSE\left(F^{V''}\right) + \sum_{\bar{g} \in Dom(b) \times Dom(c) \ldots} \frac{H'_{\bar{g}} \cdot H''_{\bar{g}}}{H'_{\bar{g}} + H''_{\bar{g}}} \left(F^{V'}(\bar{g}) - F^{V''}(\bar{g})\right)^2$$

  - ✓ $H'_{\bar{g}}$ is the number of non-null *V'* descendants
  - ✓ $F^{V'}(\bar{g})$ is the value of the f-slice $F^{V'}$ at coordinate $\bar{g}$

- Incremental computation of the errors deeply impacts on the computation time of the shrink algorithms proposed next

A Heuristic Algorithm

- ***Fixed size-reduction problem***: find the reduction that yields the minimum SSE among those whose size is not larger than $size_{max}$
  - ✓ The search space has exponential size
  - ✓ The presence of hierarchy-related constraints reduces the problem search space
  - ✓ Worst case when no such constraints are present: the number of different partitions of a set with |Dom(a)| elements

$$B_{|Dom(a)|} = \sum_{k=0}^{|Dom(a)|-1} \binom{|Dom(a)|-1}{k} B_k$$

⇩

A heuristic approach is needed to satisfy the real-time computation required in OLAP

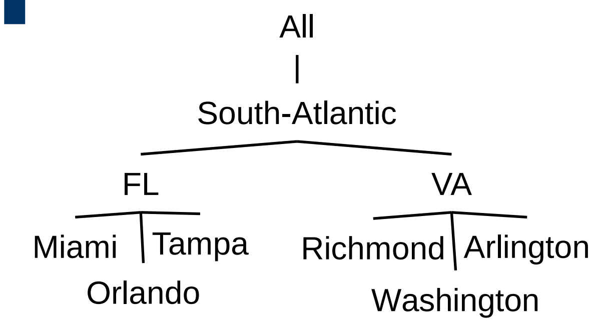
A Heuristic Algorithm

- We adopted an agglomerative hierarchical clustering algorithm with constraints
  - ✓ the algorithm starts from a clustering, where each cluster corresponds to an f-slice with a single value of the hierarchy.
  - ✓ merging two clusters means merging two f-slices
  - ✓ As a merging criterion we adopted the *Ward's minimum variance method*
    - at each step we merge the pair of f-slices that leads to minimum ΔSSE increase
  - ✓ Two f-slices can be merged only if the resulting reduction preserves the hierarchy semantics
  - ✓ The agglomerative process is stopped when the next merge meets the constraint expressed by $size_{max}$

- Our approach can solve the symmetric problem too
  - ✓ ***Fixed error-reduction problem***: find the reduction that yields the minimum size among those whose SSE is not larger than SSE_{max}

A Heuristic Algorithm

City	Year: 2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

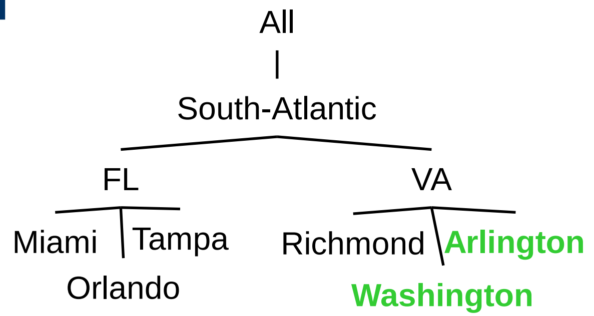


A Heuristic Algorithm

City	2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

Year

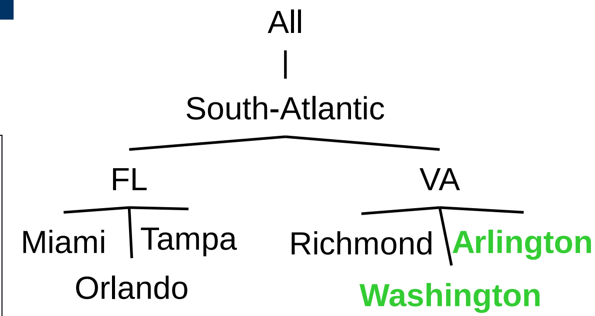
ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	



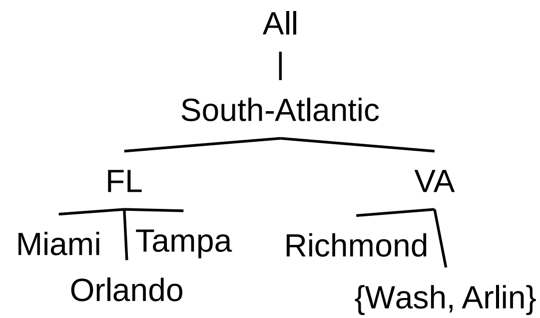
A Heuristic Algorithm

City	Year 2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	



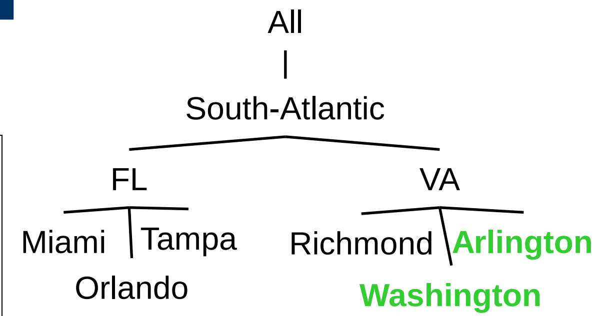
City	Year 2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				2.5



A Heuristic Algorithm

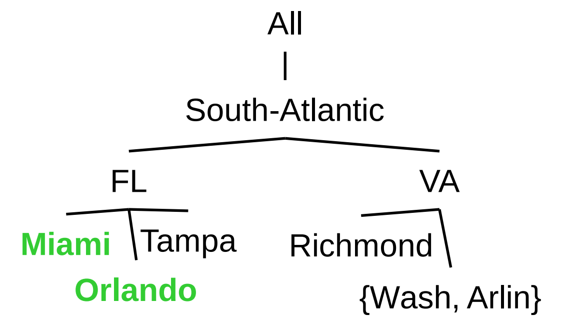
City	Year 2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	



City	Year 2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				2.5

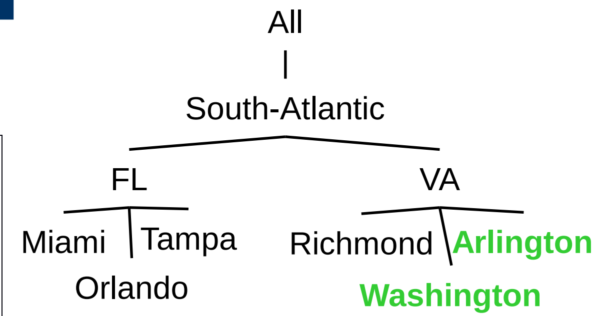
ΔSSE	Miami	Orlando	Tampa	Wash., Arlin.	Richmond
Miami					
Orlando	**8.5**				
Tampa	85	97.5			
Wash., Arlin.					
Richmond				14.7	



A Heuristic Algorithm

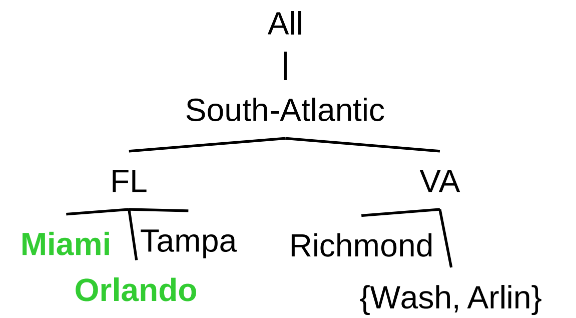
City	2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	

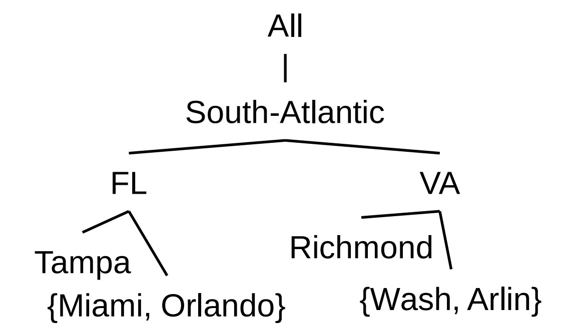


City	2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				2.5

ΔSSE	Miami	Orlando	Tampa	Wash., Arlin.	Richmond
Miami					
Orlando	**8.5**				
Tampa	85	97.5			
Wash., Arlin.					
Richmond				14.7	



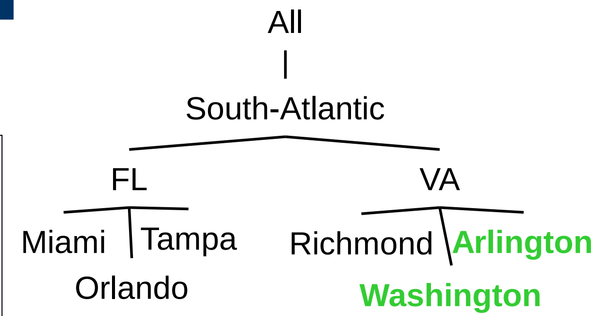
City	2010	2011	2012	*SSE*
Miami, Orlando	45.5	44	51	*8.5*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				11



A Heuristic Algorithm

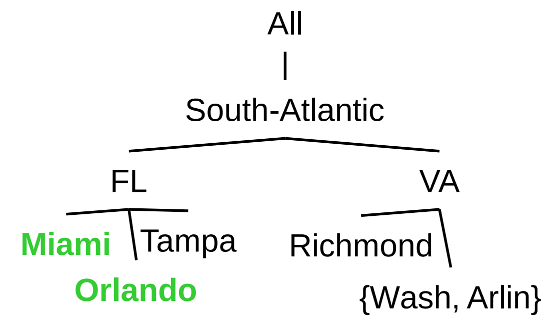
City	Year 2010	Year 2011	Year 2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	



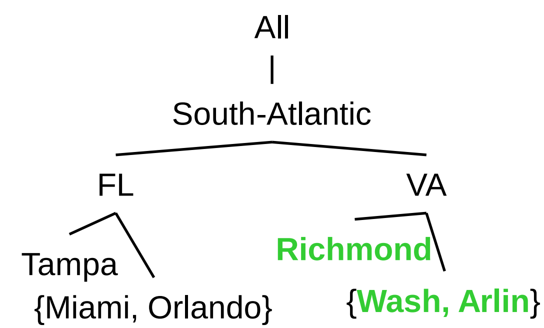
City	Year 2010	Year 2011	Year 2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				2.5

ΔSSE	Miami	Orlando	Tampa	Wash., Arlin.	Richmond
Miami					
Orlando	**8.5**				
Tampa	85	97.5			
Wash., Arlin.					
Richmond				14.7	



City	Year 2010	Year 2011	Year 2012	*SSE*
Miami, Orlando	45.5	44	51	*8.5*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				11

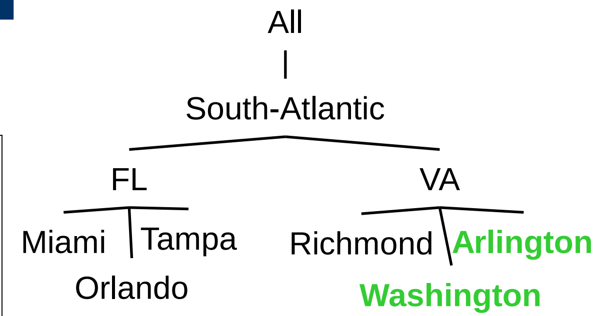
ΔSSE	Miami, Orlando	Tampa	Wash., Arlin.	Richmond
Miami, Orlando				
Tampa	127.3			
Wash., Arlin.				
Richmond			**14.7**	



A Heuristic Algorithm

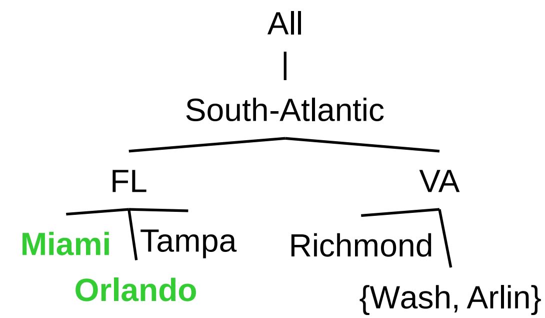
City	2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington	47	45	51	*0*
Richmond	43	46	49	*0*
Arlington	—	47	52	*0*
				0

ΔSSE	Miami	Orlando	Tampa	Washington	Richmond	Arlington
Miami						
Orlando	8.5					
Tampa	85	97.5				
Washington						
Richmond				10.5		
Arlington				**2.5**	5	



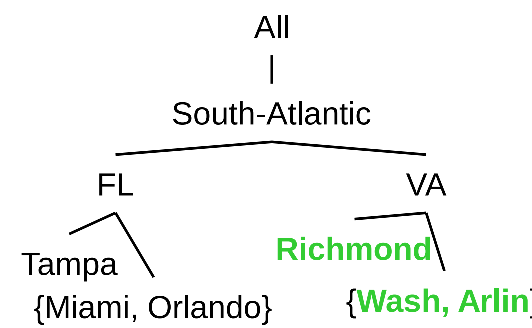
City	2010	2011	2012	*SSE*
Miami	47	45	50	*0*
Orlando	44	43	52	*0*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				2.5

ΔSSE	Miami	Orlando	Tampa	Wash., Arlin.	Richmond
Miami					
Orlando	**8.5**				
Tampa	85	97.5			
Wash., Arlin.					
Richmond				14.7	

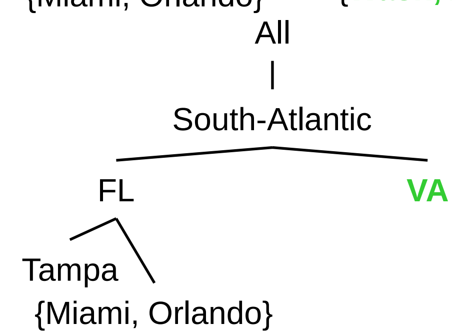


City	2010	2011	2012	*SSE*
Miami, Orlando	45.5	44	51	*8.5*
Tampa	39	50	41	*0*
Washington, Arlington	47	46	51.5	*2.5*
Richmond	43	46	49	*0*
				11

ΔSSE	Miami, Orlando	Tampa	Wash., Arlin.	Richmond
Miami, Orlando				
Tampa	127.3			
Wash., Arlin.				
Richmond			**14.7**	

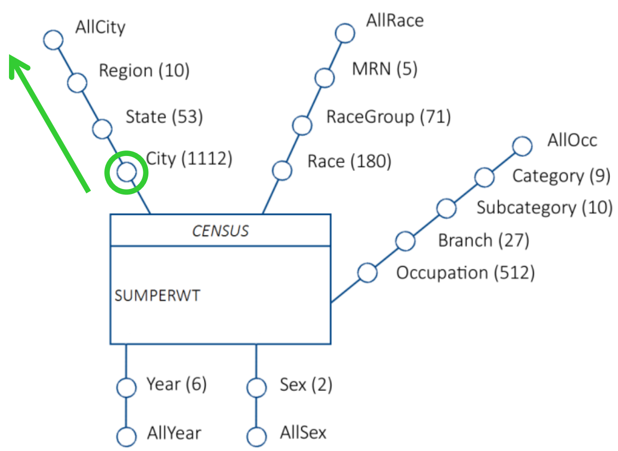


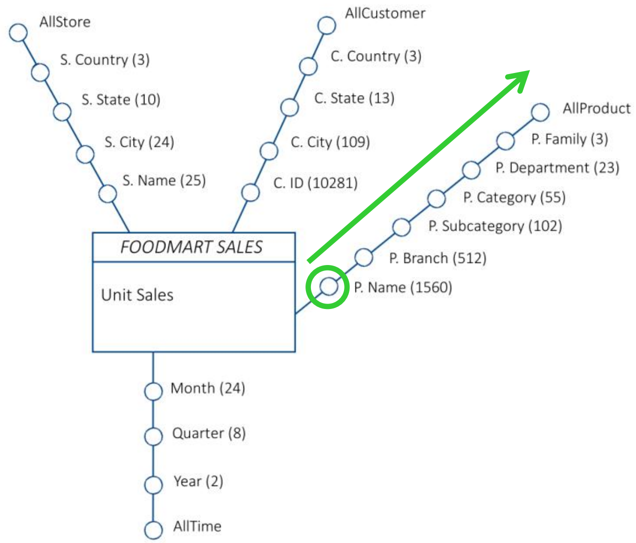
City	2010	2011	2012	*SSE*
Miami. Orlando	45.5	44	51	*8.5*
Tampa	39	50	41	*0*
Virginia	45	46	50.6	*14.7*
				23.2



Experimental Results

- 2 different datasets adopted, 4 reduction problems
  - ✓ Different hierarchy features
  - ✓ Different sparsity
  - ✓ Different sizes

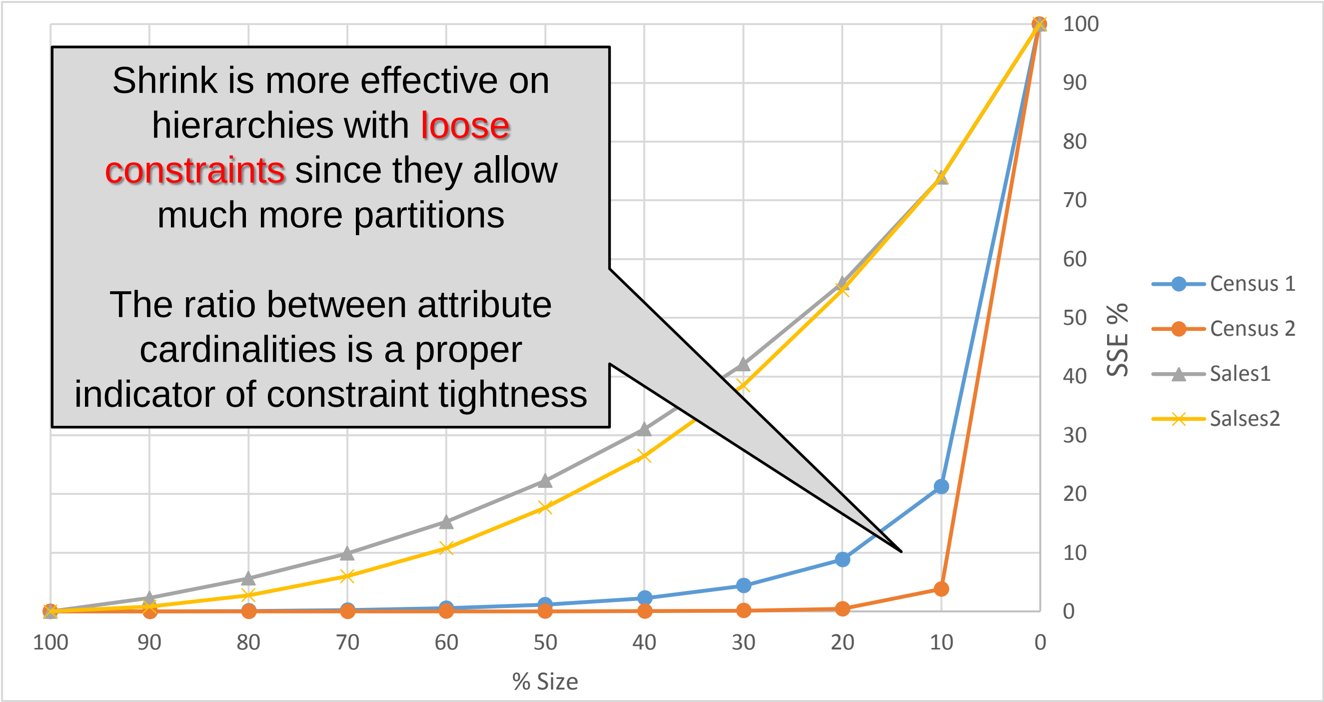




Fact	#Initial f-slice	# facts	#not-null facts	Density
Census1	1112	≈ 34 M	≈ 245 K	0,75%
Census2	1112	≈ 50 K	≈ 12 K	24,17%
Sales1	1560	≈ 34 M	≈ 200 K	0,58%
Sales2	1560	≈ 28 K	≈ 6 K	22,20%

Aprroximation errors

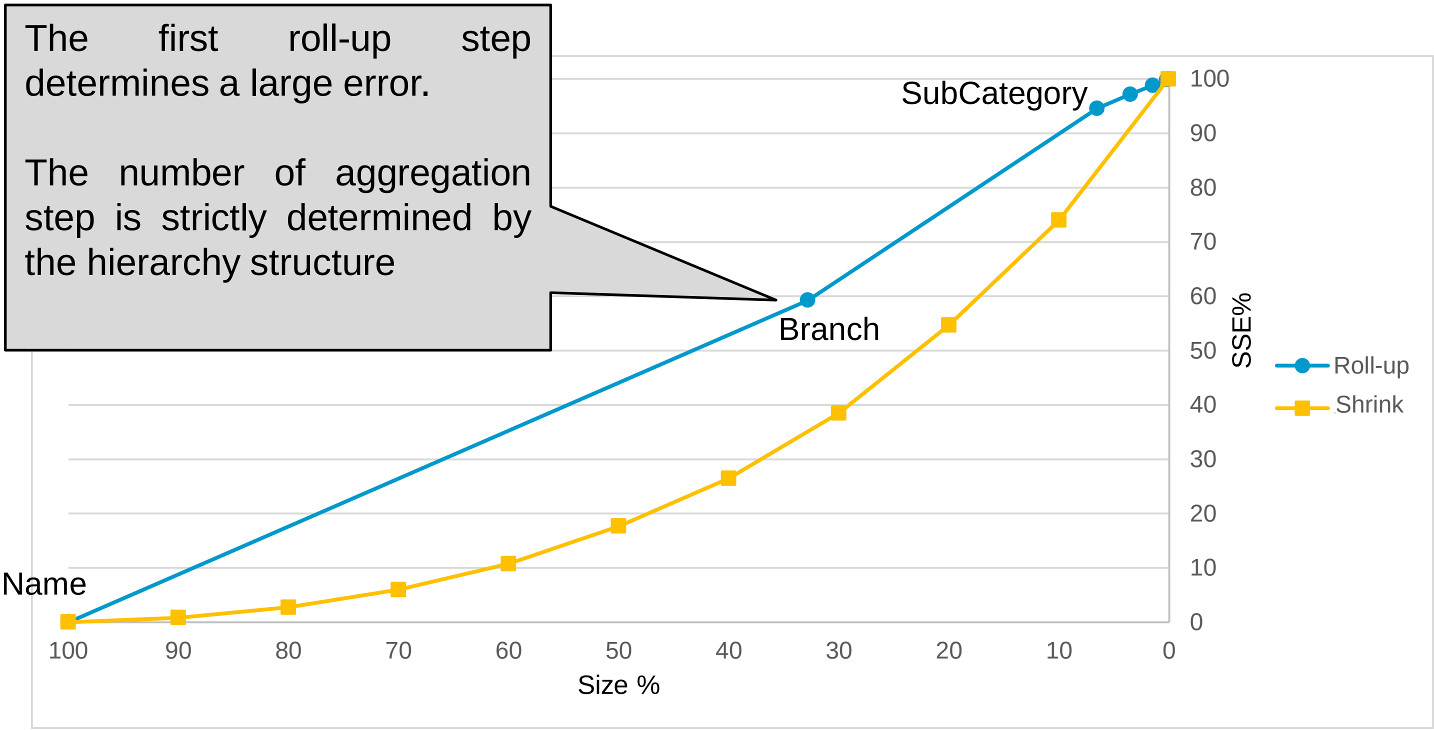
- The SSE has been normalized to allow comparisons
  - ✓ $SSE\% = \frac{SSE(Red_h(C))}{SSEMAX_h(C)}$



- Further cube features that impact on effectiveness are:
  - ✓ Sparsity: the higher the sparsity, the lower the SSE increase
  - ✓ Variance of the values: the higher the variance the cells to be merged, the higher the SSE increase

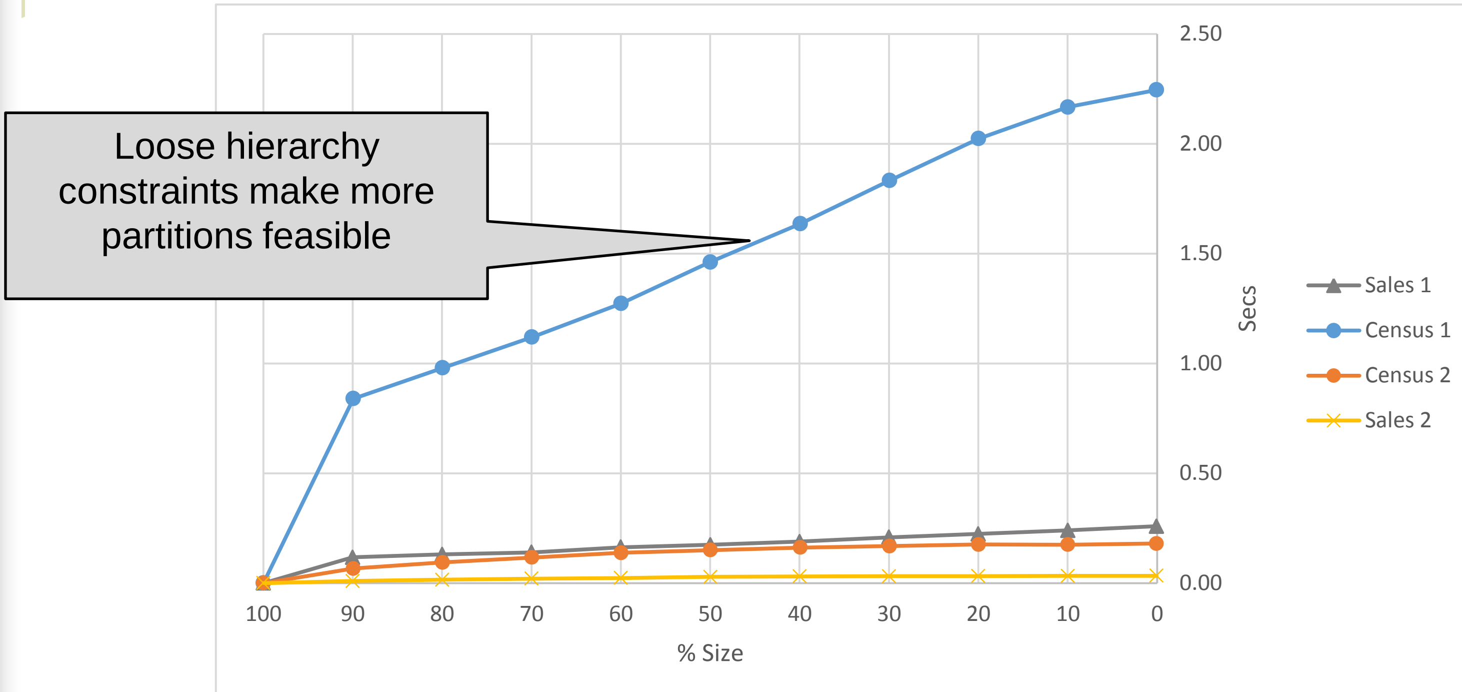
Shrink vs Roll-up

- We compared the two operators on the Sales 2 cube applying the AVG operator when rolling-up



Efficiency

- Tests are run on a Pentium i5 quad-core (2.67 GHz, 4 GB RAM)
  - ✓ Windows 7-64 bits



- Further cube features that impact on efficiency are:
  - ✓ Size of the f-slice
  - ✓ Size of the cube
- A shrink step requires less than 2 milliseconds in all of the previous test

Optimal vs Greedy

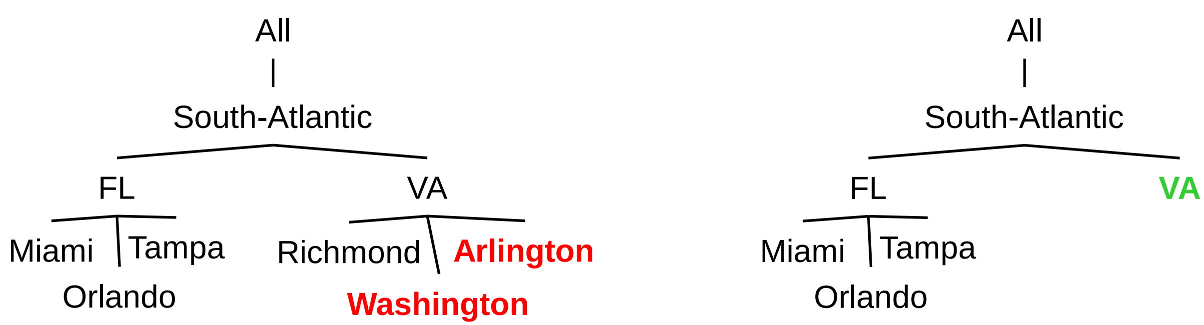
- We adopted a branch-and-bound approach based on an optimal enumeration process
  - ✓ We set $size_{max}$ = 0.3 |*Dom*(*a*)|
- Possible only on toy examples

#f-slice	# initial facts	# facts at $size_{max}$	Error	B&B execution time
23	184	90	8.31%	3 secs
24	192	90	0%	4 secs
27	135	60	0%	3 mins 12 secs
53	543	----	-----	> 6 hours

Conclusions

- Shrink: a new OLAP operation to cope with the information flooding problem
  - ✓ We proposed a heuristic implementation
  - ✓ We analyzed its effectiveness and efficiency

- Now working on:
  - ✓ Effectiveness: extending the formulation of the operator to work on several hierarchies simultaneously
  - ✓ Efficiency: studying smarter heuristics and different implementations of the shrink idea
    - The *eager* shrink operator collapses at each step all the children of a given value



  - ✓ Optimality: studying optimal algorithms exploiting a column generation technique in a set partitioning formulation
  - ✓ Visualization: find out visual metaphors for representing complex pivot tables