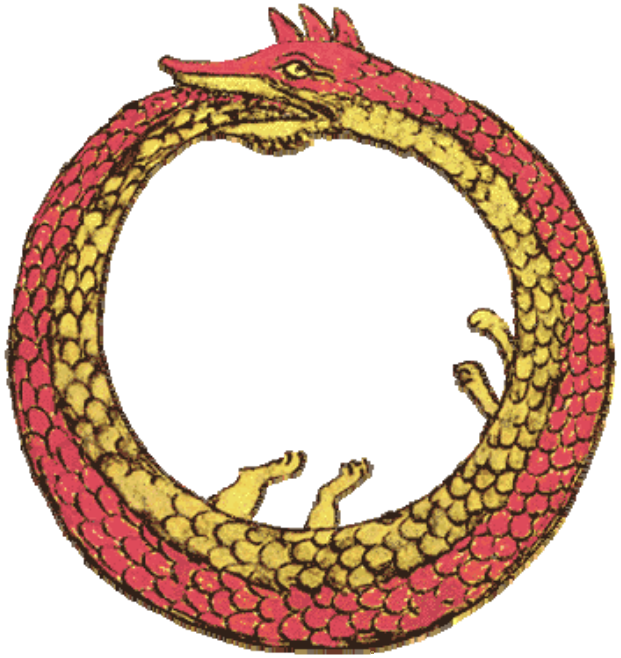
Autocannibalistic and Anyspace Indexing Algorithms with Applications to Sensor Data Mining

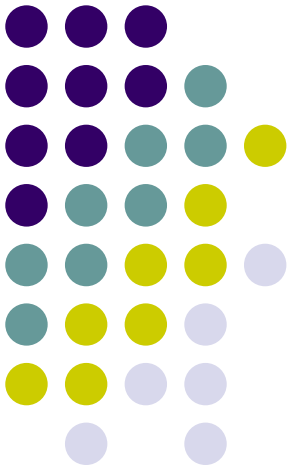
Lexiang Ye, Xiaoyue Wang, Eamonn Keogh

Dept. of Computer Science & Engineering

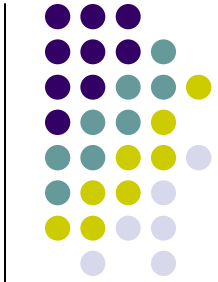
Agenor Mefra-Neto

ISCA Technologies

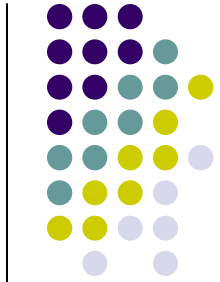




Summary



- Indexing problem
- Orchard's algorithm (1991)
- Fatal flaw: quadratic space complexity
- We propose:
  - Anyspace framework and Autocannibalistic algorithm
  - Flexible in memory size



Orchard's Algorithm

Triangular Inequality

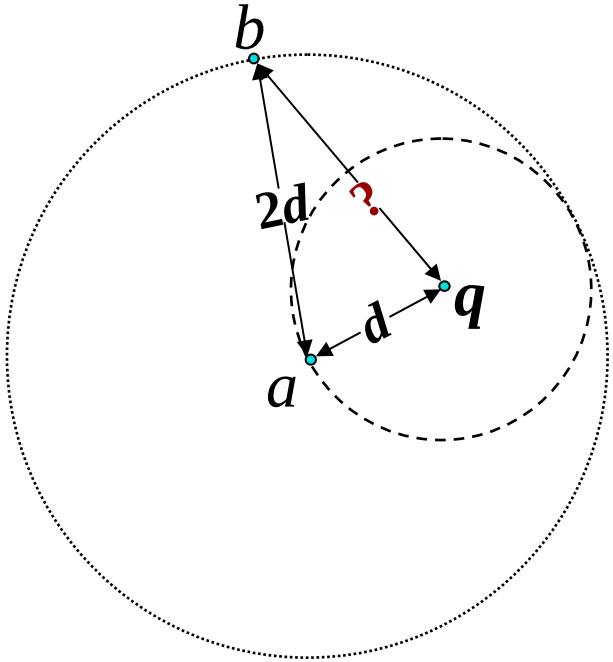
- DIST $(a, q) = d$ is calculated
- DIST (a, b) prior information
- DIST (b, q) >

  DIST (a, b) - DIST (a, q)
- Prune criteria:

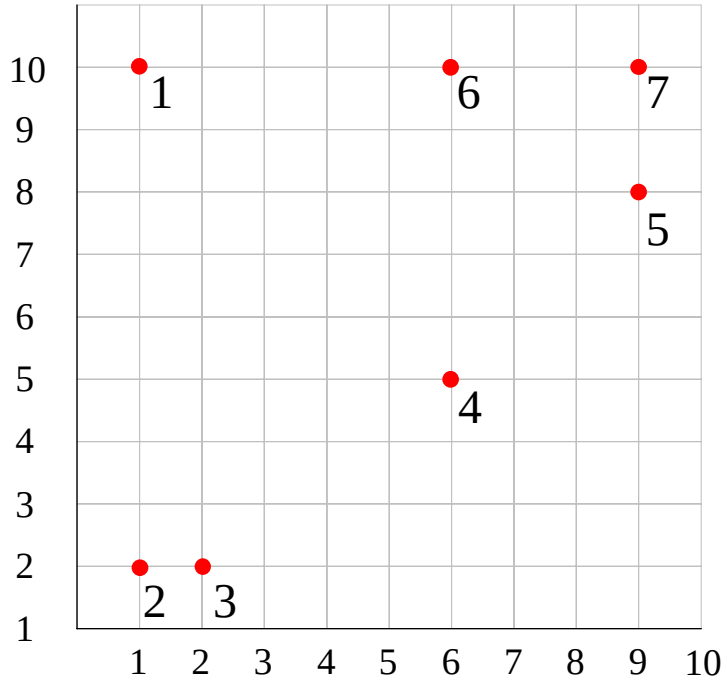
  DIST $(a, b) > 2d$

  ⇒ DIST $(b, q) > d$

non-nearest neighbor !



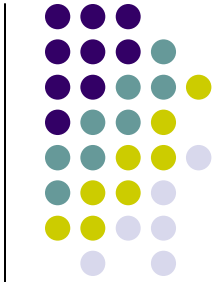
Orchard's algorithm Example



Dataset *D*

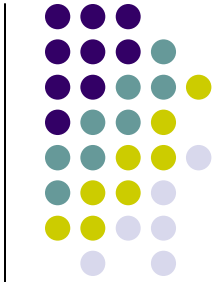
	X	Y
a_1	1	10
a_2	1	2
a_3	2	2
a_4	6	5
a_5	9	8
a_6	6	10
a_7	9	10

Data Structure: sorted list of neighbors



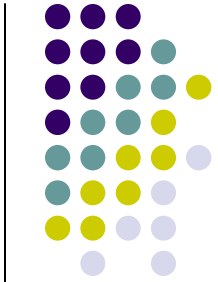
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1						
a_2						
a_3						
a_4						
a_5						
a_6						
a_7						

Data Structure: sorted list of neighbors



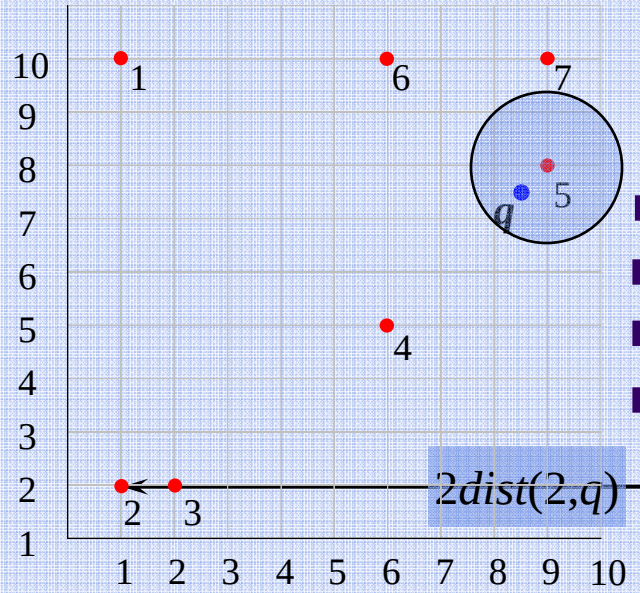
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_2						
a_3						
a_4						
a_5						
a_6						
a_7						

Data Structure: sorted list of neighbors



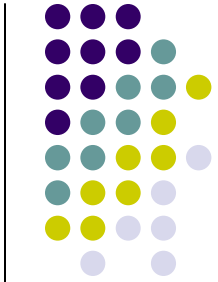
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}

Orchard's algorithm Example



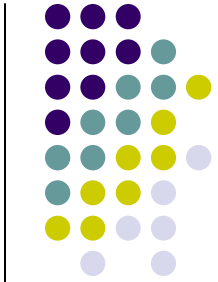
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}

Orchard's algorithm



- Simple
- Parameter free
- **Drawbacks?**

Data Structure:

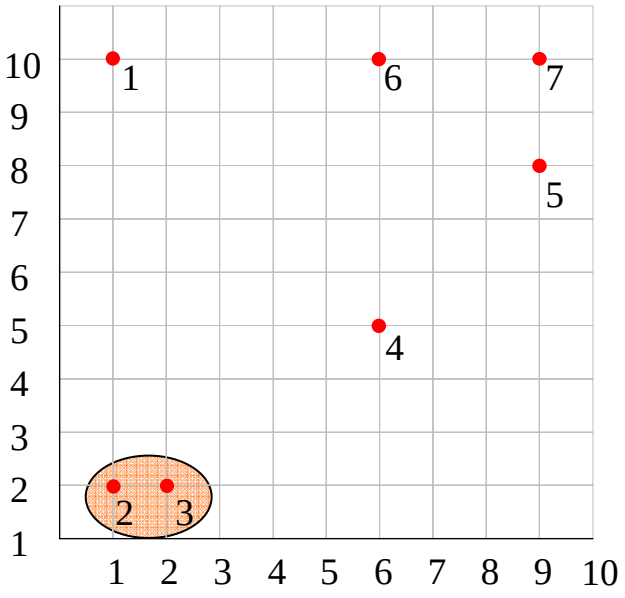


Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7				.2}
a_2	**3** {1.0}	**4** {5.8				**7** {11.3}
a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}

Quadratic Space

Observation

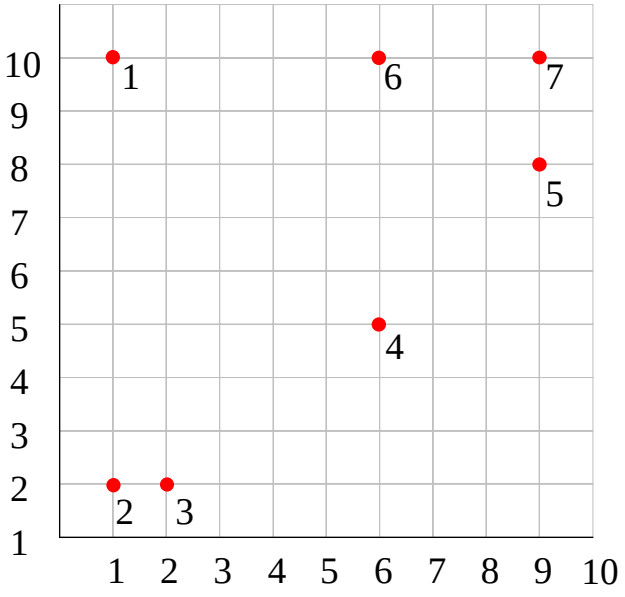
- Same structure
- Data redundancy



Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
a_3	*goto* a_2					
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}

Extreme Case

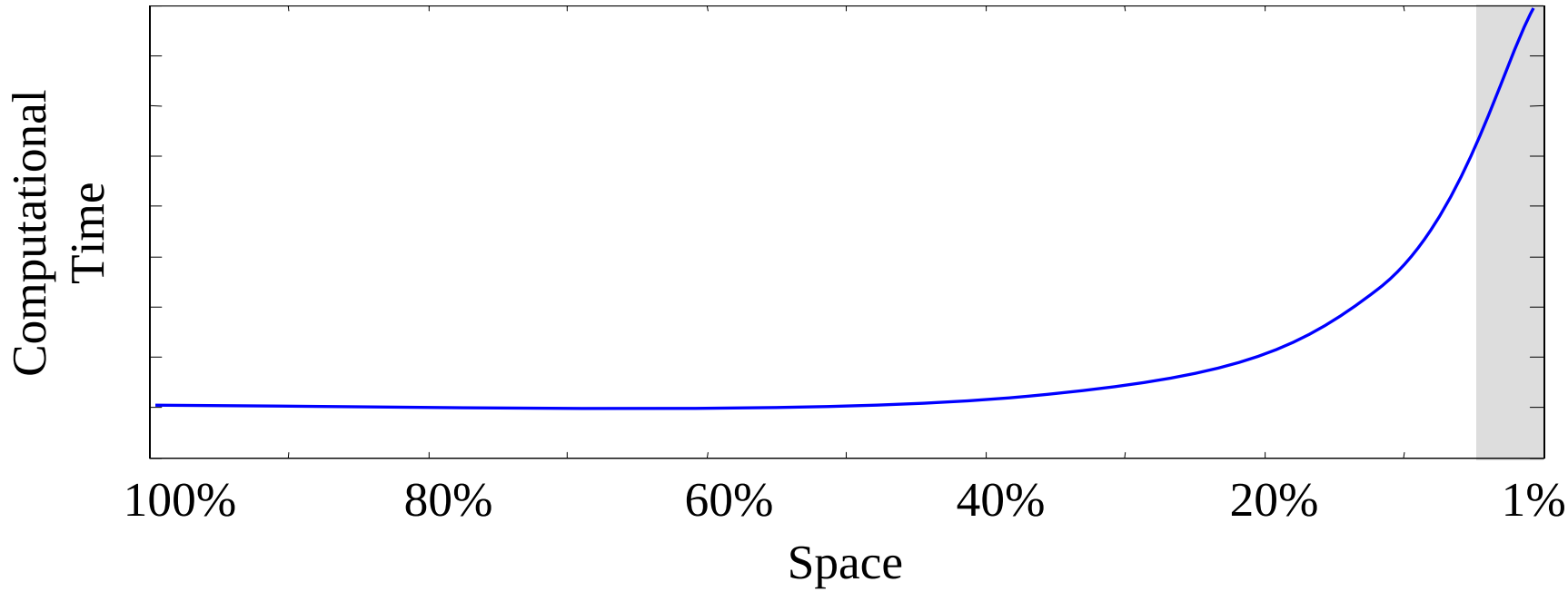
- Same structure
- Data redundancy



Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	*goto* a_6					
a_2	*goto* a_4					
a_3	*goto* a_2					
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	*goto* a_4					
a_6	*goto* a_7					
a_7	*goto* a_5					

Method

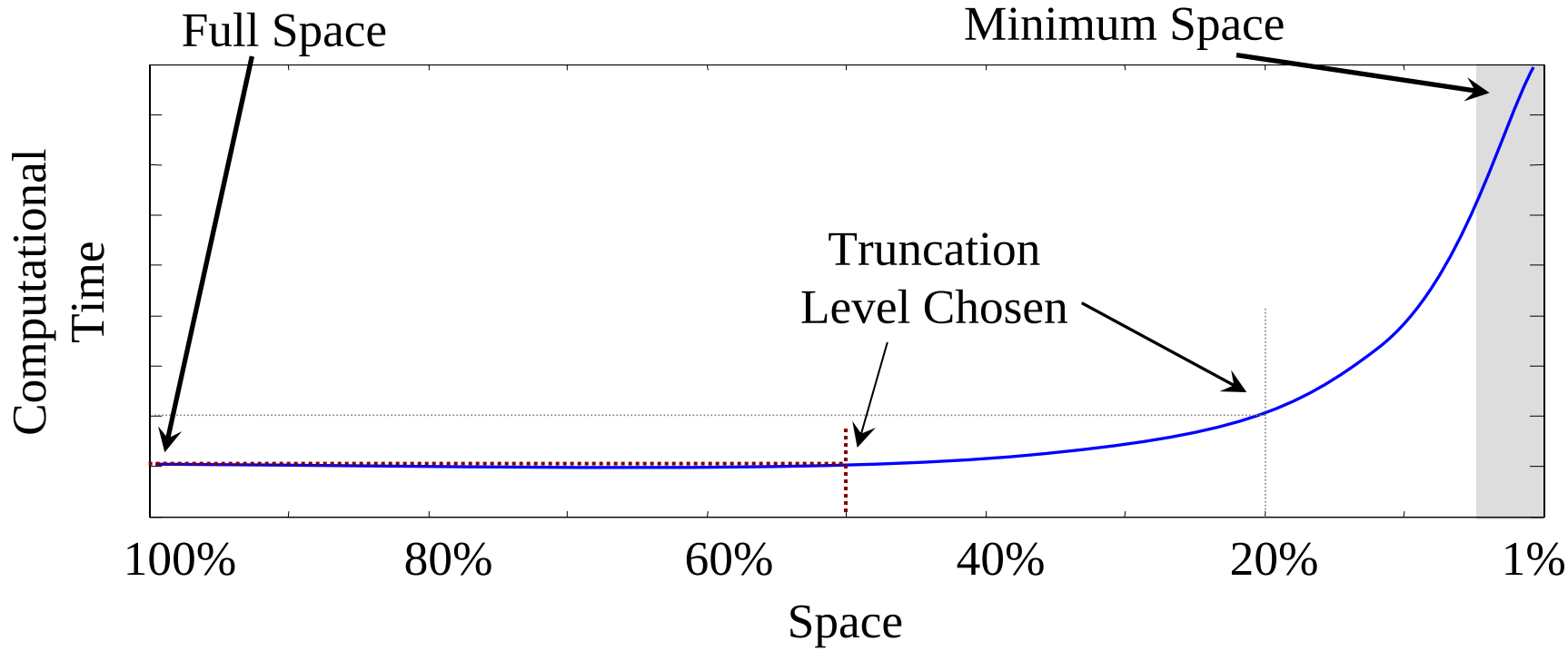
- Calculate available memory
- Delete “redundancy” lists
- *Anyspace* framework



Method

Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}

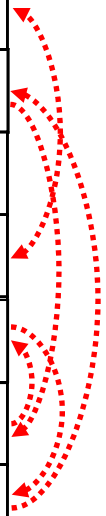
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	*goto* a_6					
a_2	*goto* a_4					
a_3	*goto* a_2					
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	*goto* a_4					
a_6	*goto* a_7					
a_7	*goto* a_5					



Modifications of Index

- Reorder the lists

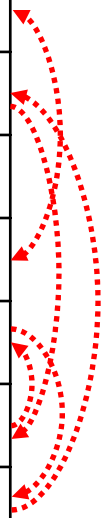
Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}



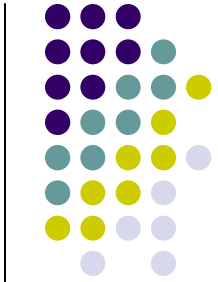
Modifications of Index

- Reorder the lists
- Add *goto list*

Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}
a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}



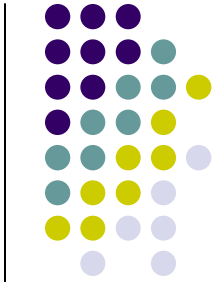
Fit the index in the memory



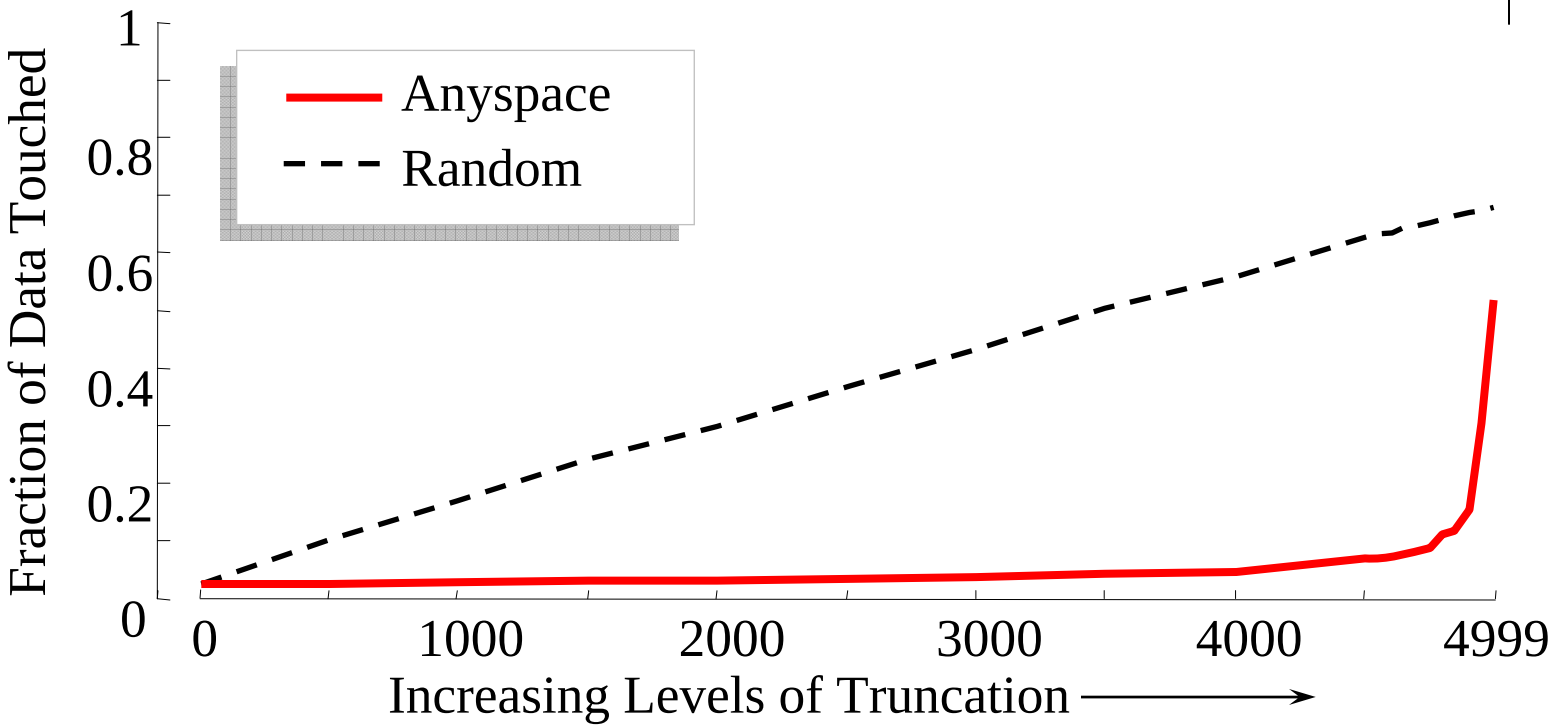
- 60% of index

goto list	Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
Linear	a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
goto a_4	a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}
goto a_4	a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
goto a_4	a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
	a_6	*goto* a_7					
	a_2	*goto* a_3					
	a_5	*goto* a_7					

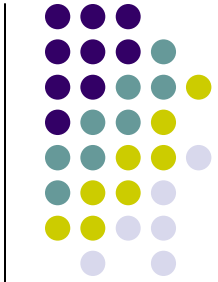
Experiment (1/2)



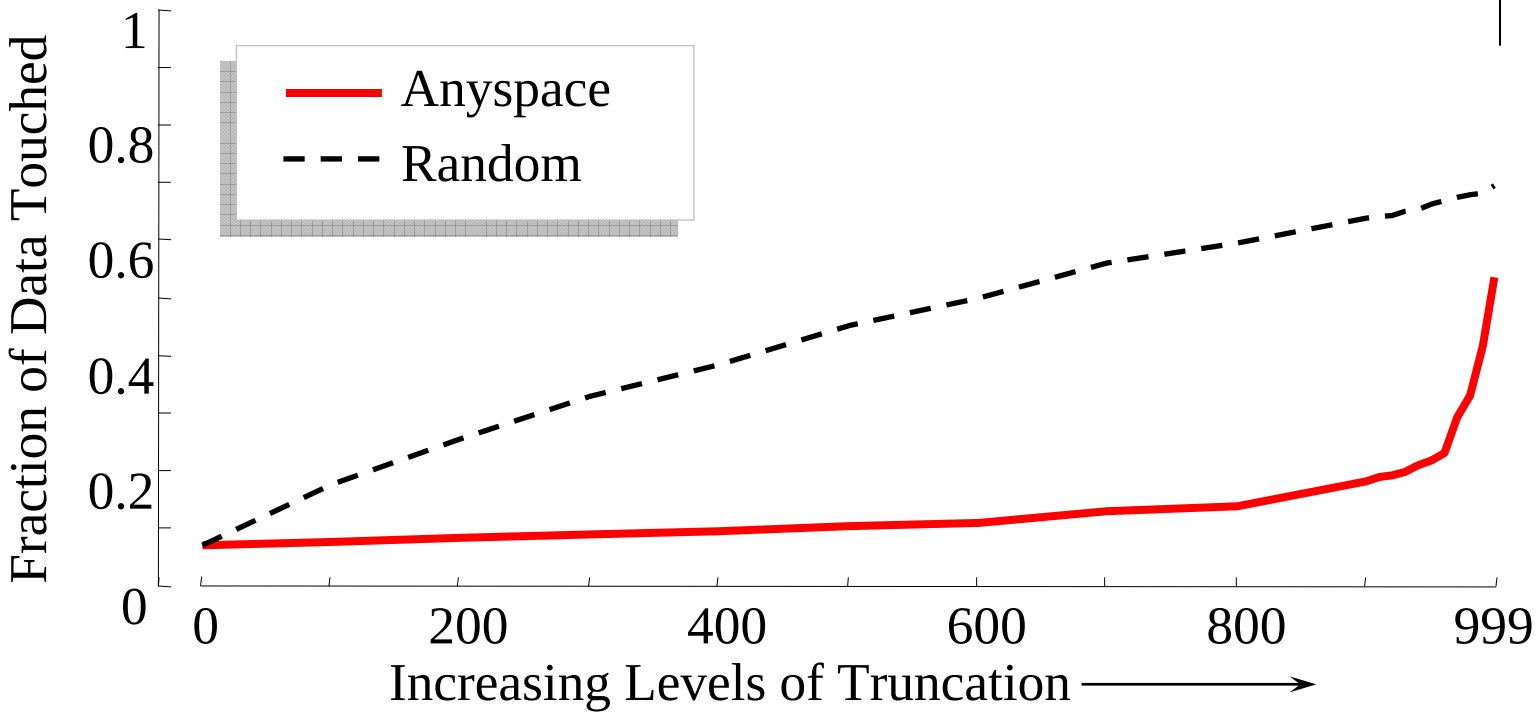
- 2D Gaussian Distribution
  - Index 5,000
  - Test 50,000


Fraction of Data Touched
1
0.8
0.6
0.4
0.2
0
Anyspace
Random
0
1000
2000
3000
4000
4999
Increasing Levels of Truncation

Experiment (2/2)



- Traffic Sensor Data:
  - Index 1,000
  - Test 50,400

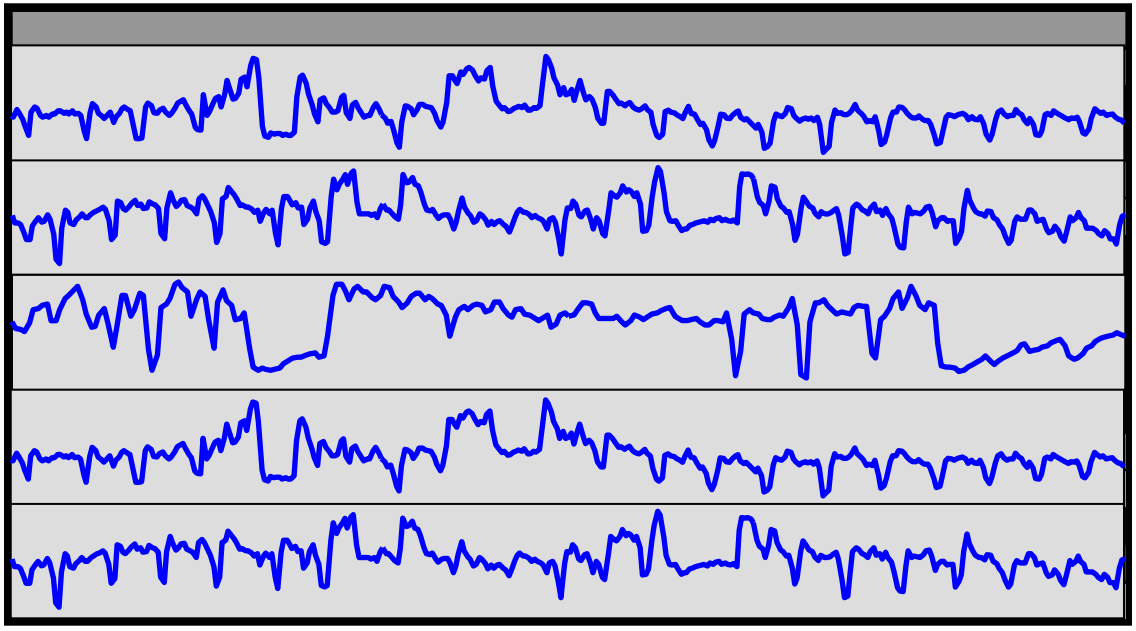

Fraction of Data Touched
1
0.8
0.6
0.4
0.2
0
Anyspace
Random
0
200
400
600
800
999
Increasing Levels of Truncation

Insect Monitoring

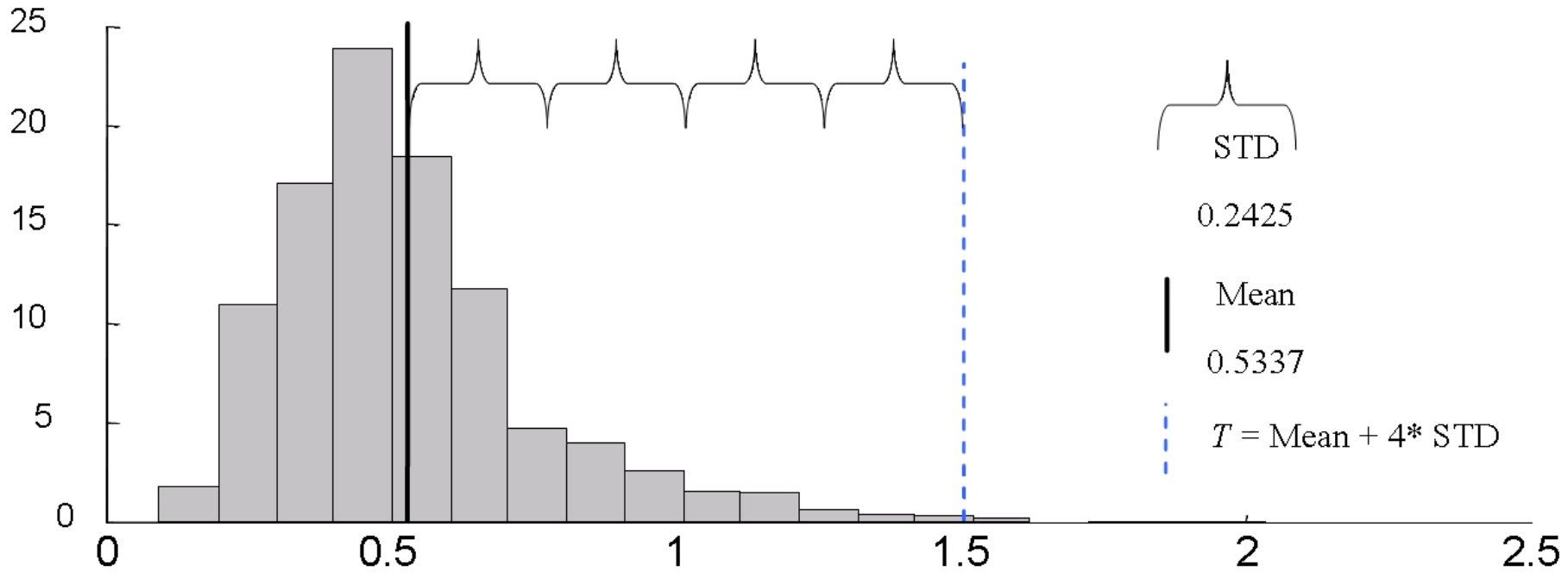
- “Smart-trap” system: design to trap mosquito
  - Classify the sex of mosquito
  - Record sound snippets of outliers
- Limitation
  - Low power – distance calculations
  - Low memory – index and outliers

Auto-cannibalistic Algorithm

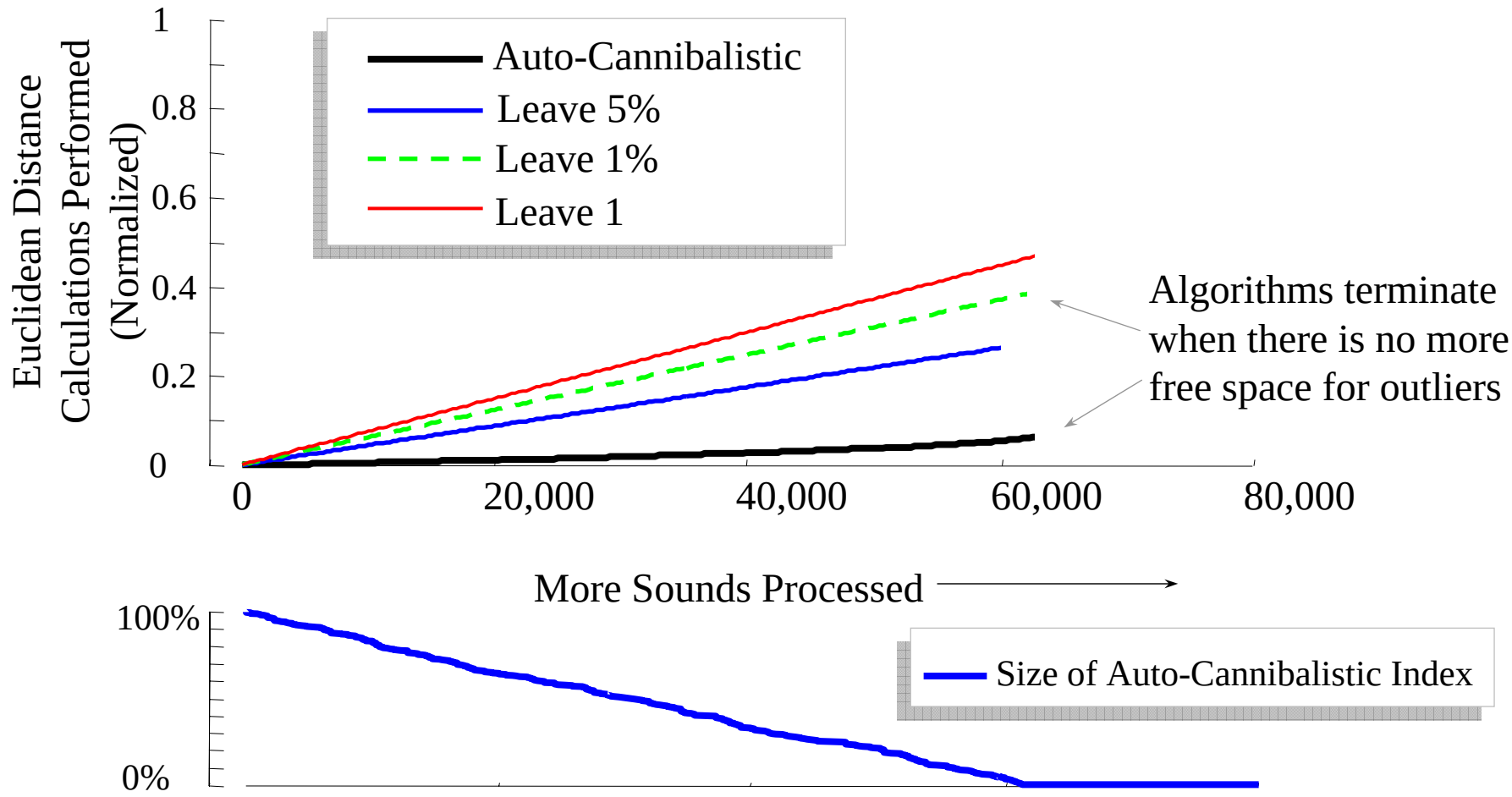
- Initially store the entire index in memory
- Dynamically truncate parts of the index to store outliers as needed



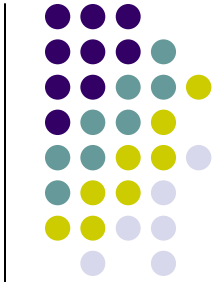
How to define outlier



Result

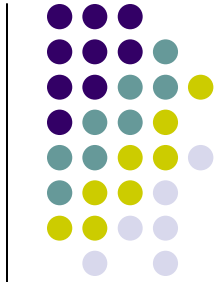

Euclidean Distance Calculations Performed (Normalized)
1
0.8
0.6
0.4
0.2
0
Auto-Cannibalistic
Leave 5%
Leave 1%
Leave 1
0
20,000
40,000
60,000
80,000
Algorithms terminate when there is no more free space for outliers
More Sounds Processed
100%
0%
Size of Auto-Cannibalistic Index

Conclusion

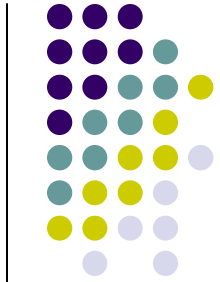


- Cast Orchard’s algorithm into anyspace framework
- first example of auto-cannibalistic algorithm

- Future work: auto-cannibalism for other applications



Questions?



Item	1st NN {dist}	2nd NN {dist}	3rd NN {dist}	4th NN {dist}	5th NN {dist}	6th NN {dist}
a_1	**6** {5.0}	**4** {7.1}	**2** {8.0}	**7** {8.0}	**3** {8.1}	**5** {8.2}
a_2	**3** {1.0}	**4** {5.8}	**1** {8.0}	**6** {9.4}	**5** {10.0}	**7** {11.3}
a_3	**2** {1.0}	**4** {5.0}	**1** {8.1}	**6** {8.9}	**5** {9.2}	**7** {10.6}
a_4	**5** {4.2}	**3** {5.0}	**6** {5.0}	**2** {5.8}	**7** {5.8}	**1** {7.1}
a_5	**7** {2.0}	**6** {3.6}	**4** {4.2}	**1** {8.2}	**3** {9.2}	**2** {10.0}
a_6	**7** {3.0}	**5** {3.6}	**1** {5.0}	**4** {5.0}	**3** {8.9}	**2** {9.4}
a_7	**5** {2.0}	**6** {3.0}	**4** {5.8}	**1** {8.0}	**3** {10.6}	**2** {11.3}