Data Warehousing and Data Mining

Lecture 12 Efficient Cube Computation and Unit Review

CITS3401
CITS5504

Zeyi Wen

Computer Science and Software Engineering

School of Maths, Physics and Computing

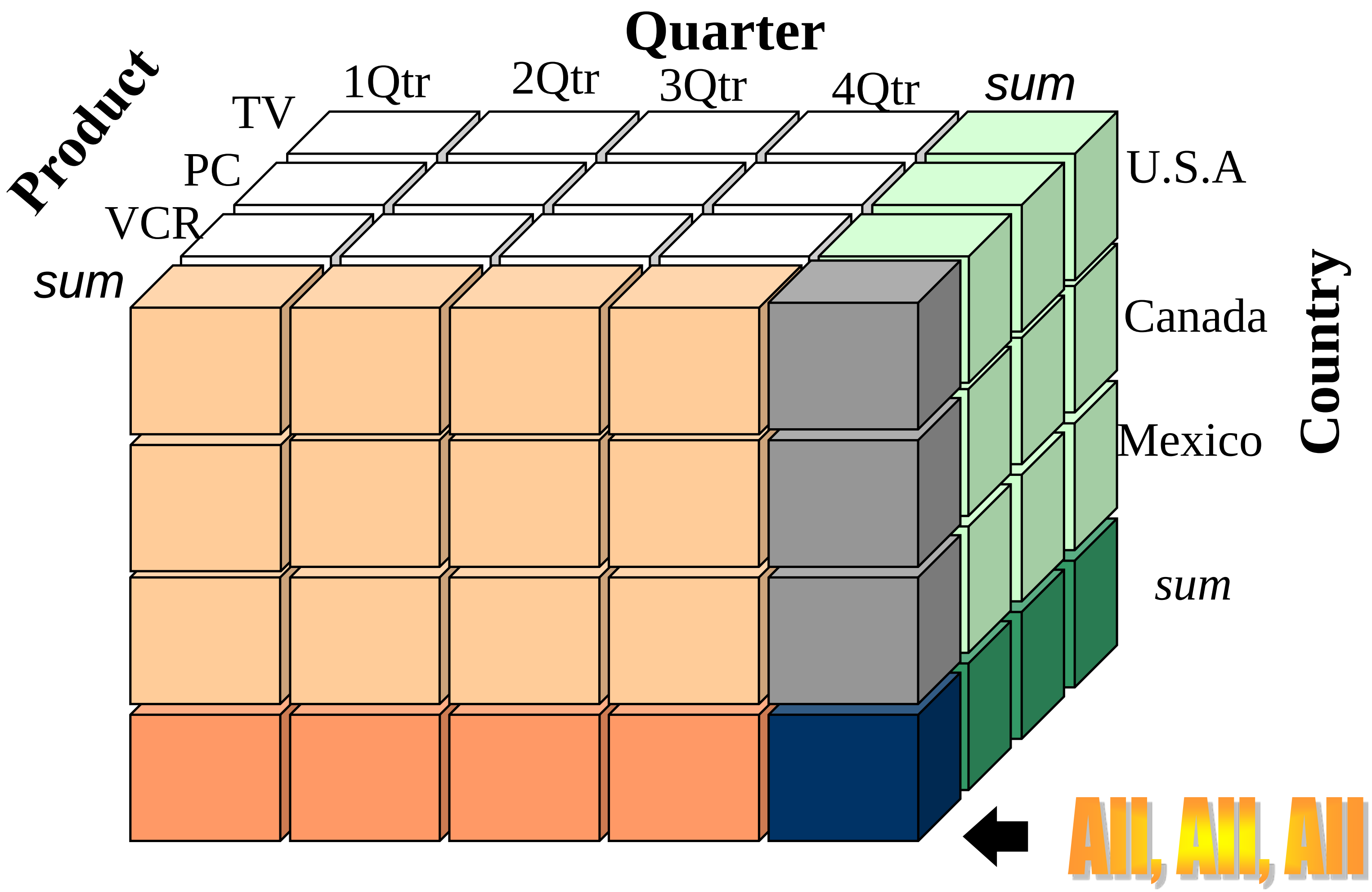
Acknowledgement: The lecture slides are based on online sources.

Lecture Outline

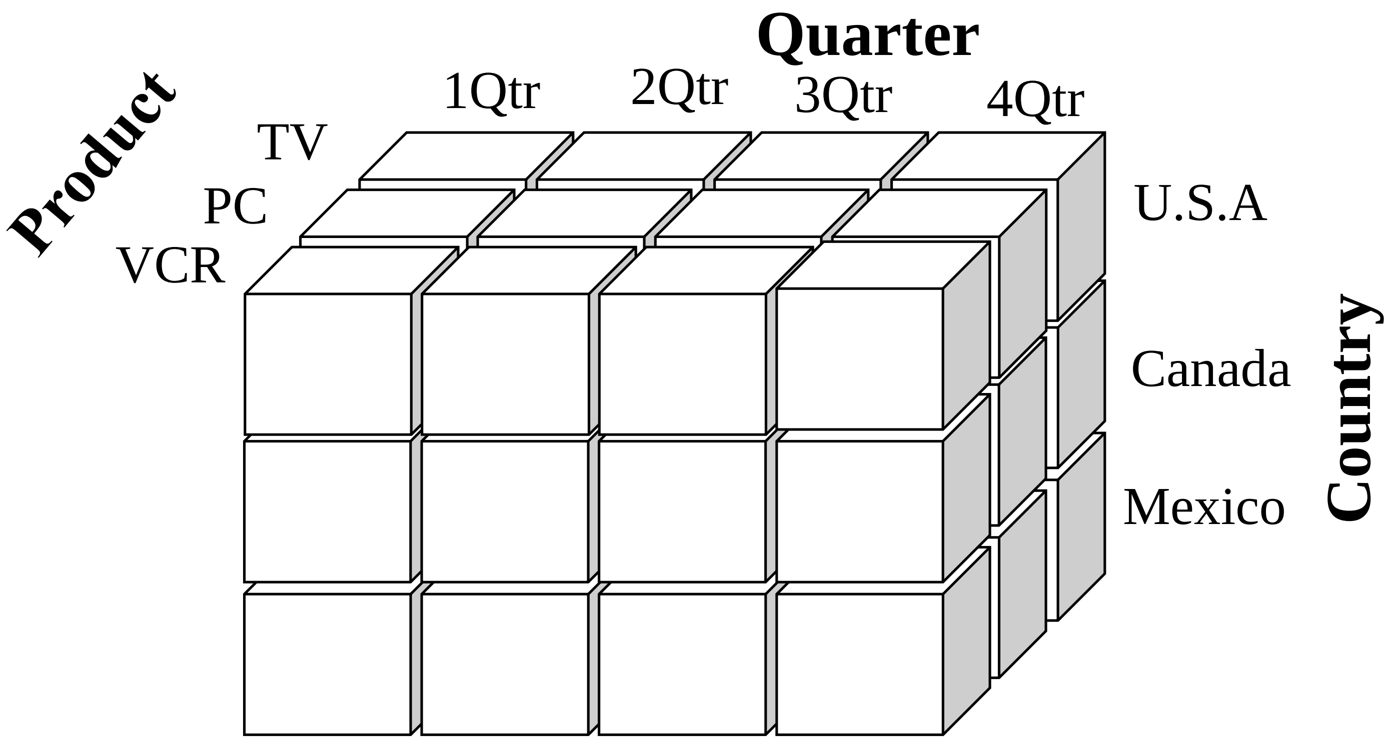
- **Efficient Cube Computation**
  - Multi-Way Array Computation
  - Bottom Up Computation

- **Unit Review and Final Exam**
  - Exam Structure
  - Review

Sample Data Cube



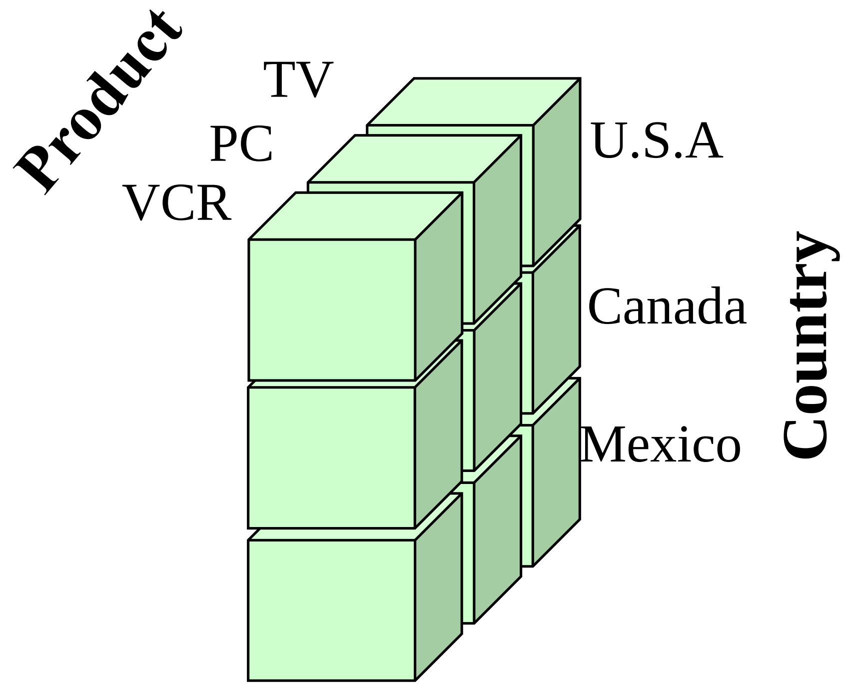
Sample Data Cube: the base cuboid

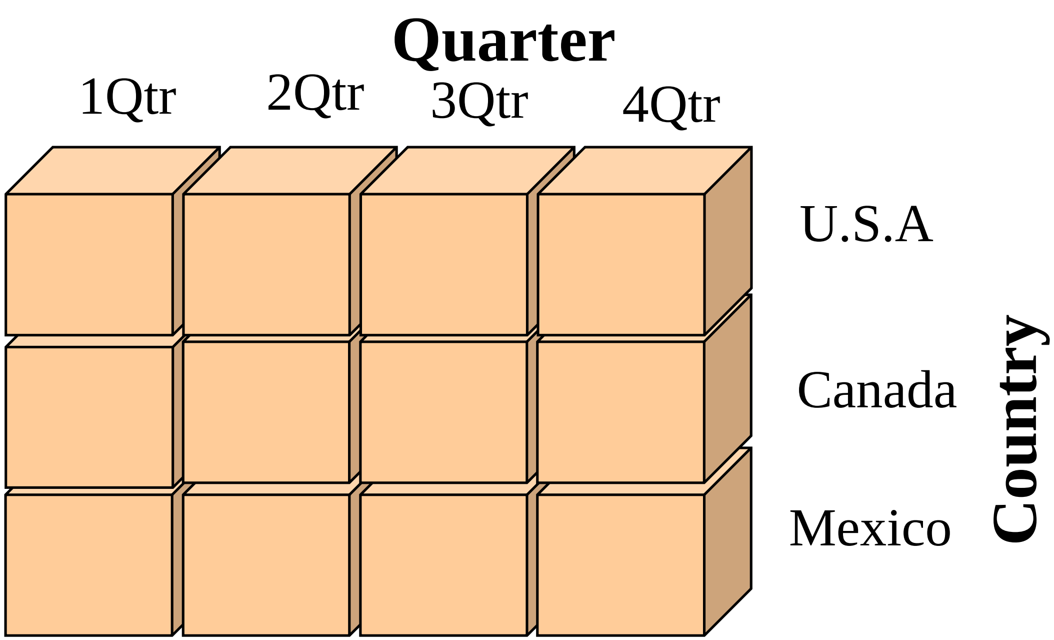


Sample Data Cube: 2-D cubiods

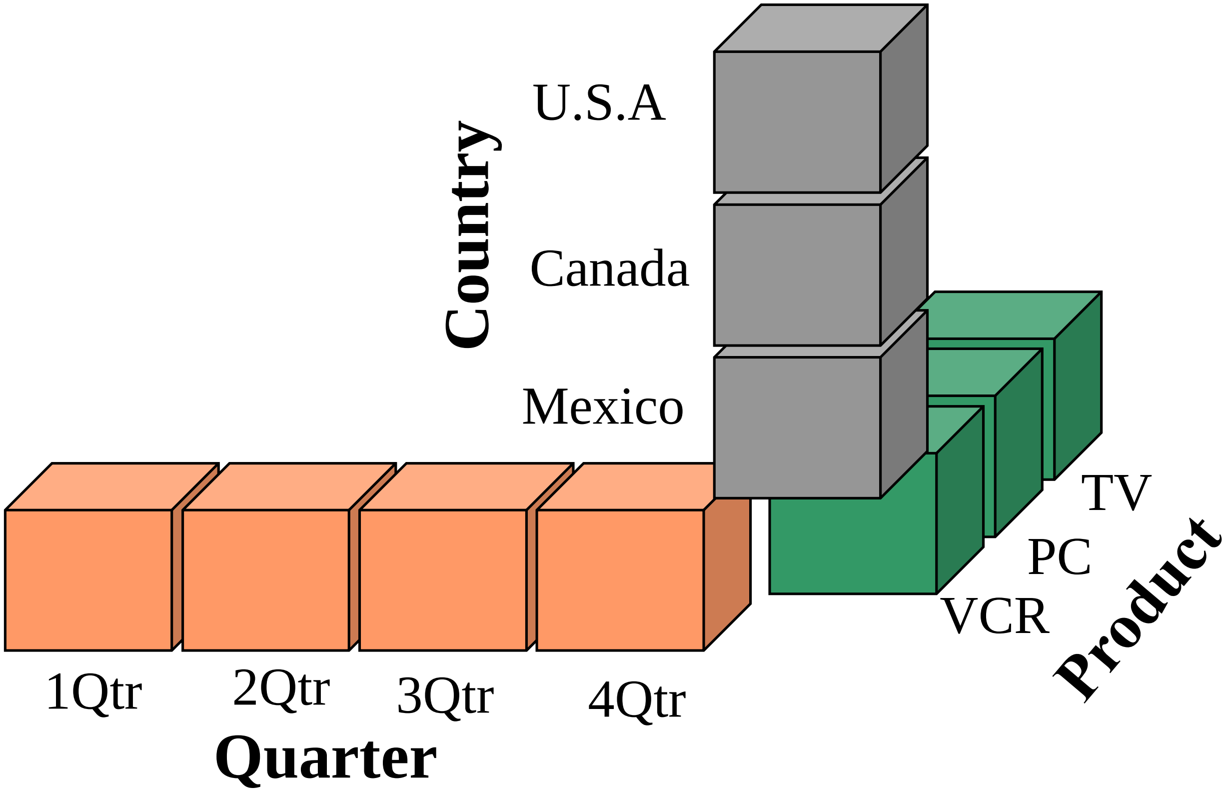
Quarter

Product

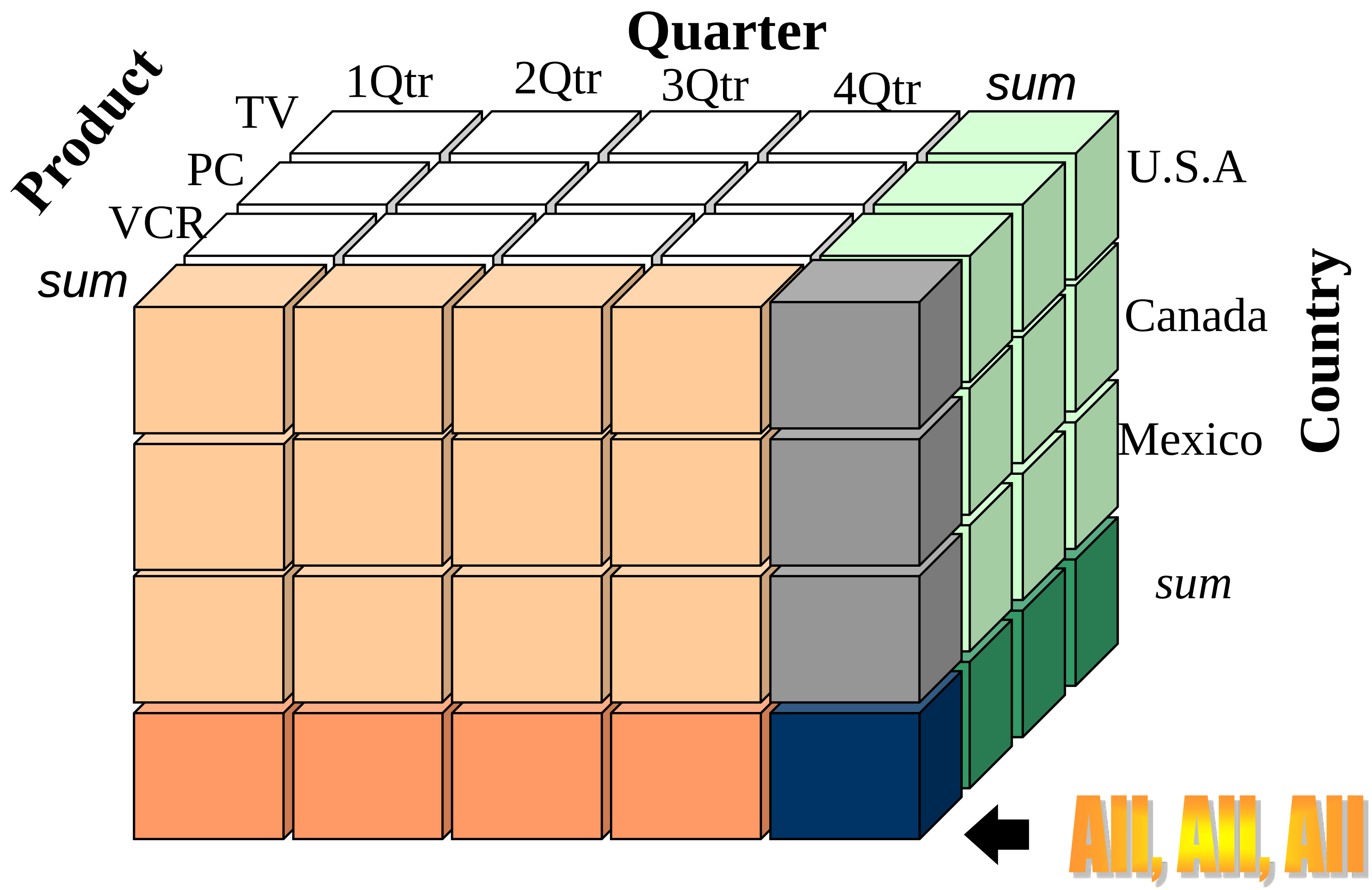




Sample Data Cube: 1-D cuboids



Sample Data Cube



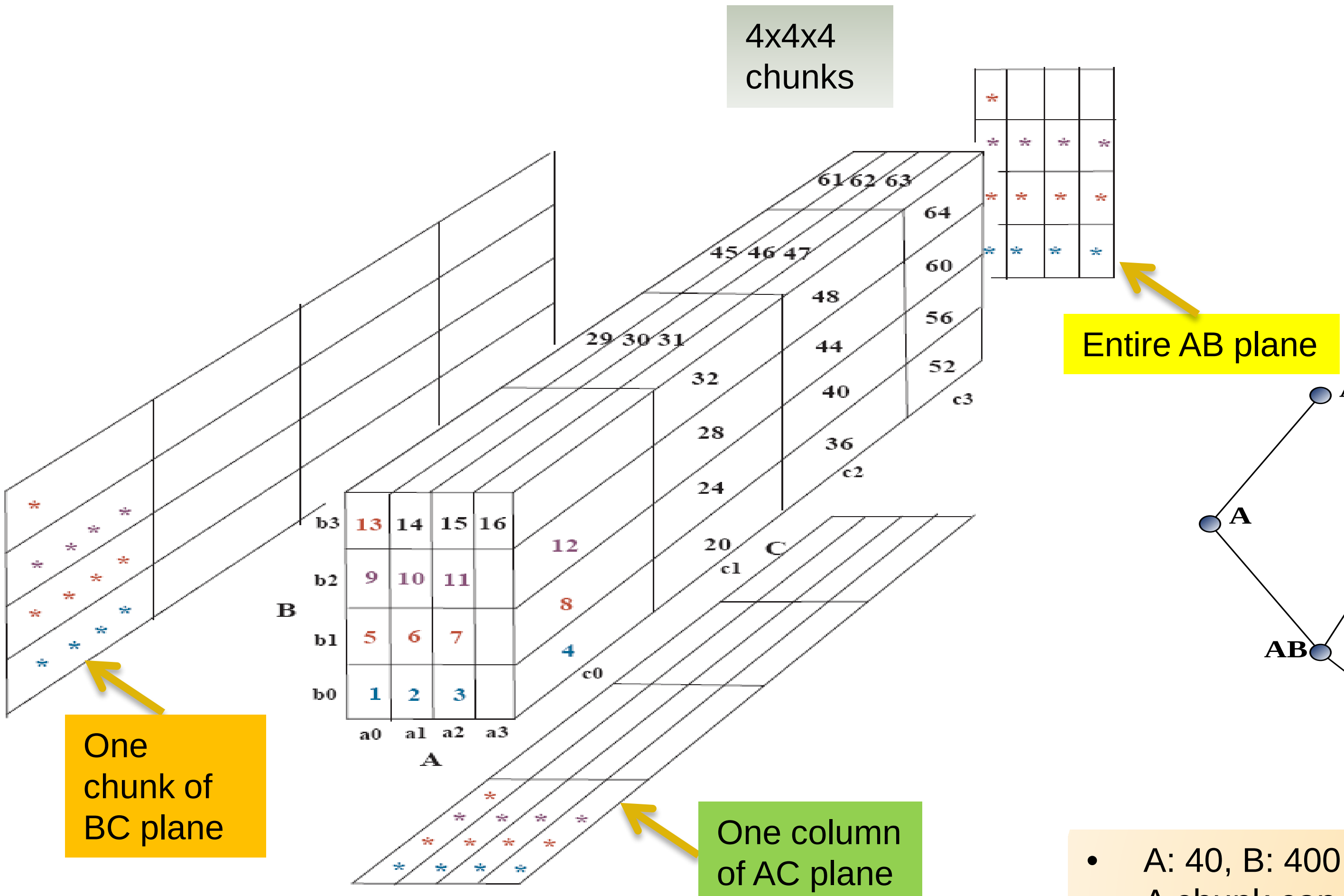
Multi-way Array Aggregation

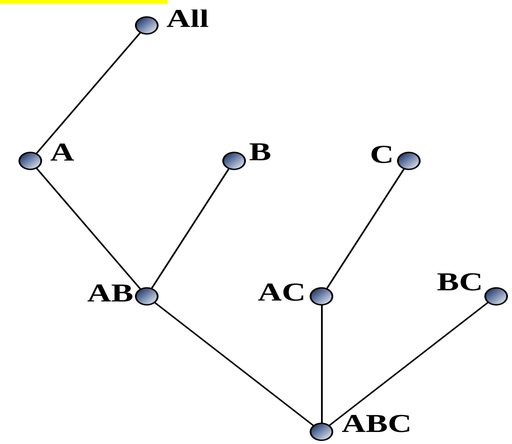
- Used for MOLAP and full cube computation
- Array-based “bottom-up” algorithm
- Using multi-dimensional chunks
  - Direct array addressing
- Simultaneous aggregation on multiple dimensions
- Intermediate aggregate values are re-used for computing ancestor cuboids
- Cannot do *Apriori* pruning: No iceberg optimisation

MOLAP (Lecture 3)

- MOLAP = Multidimensional OLAP
- Store data cube as multidimensional array
- (Usually) pre-compute all aggregates
- Advantages:
  - Very efficient data access $\longrightarrow$ fast answers
- Disadvantages:
  - Doesn't scale to large numbers of dimensions
  - Requires special-purpose data store

Multi-way Array Aggregation





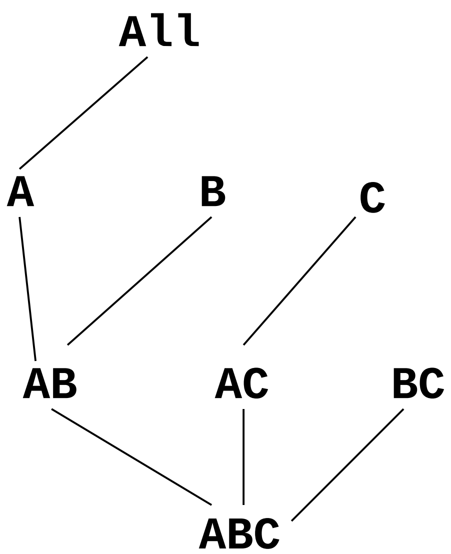
- A: 40, B: 400, C: 4000
- A chunk can only be read once.

Memory Consumption

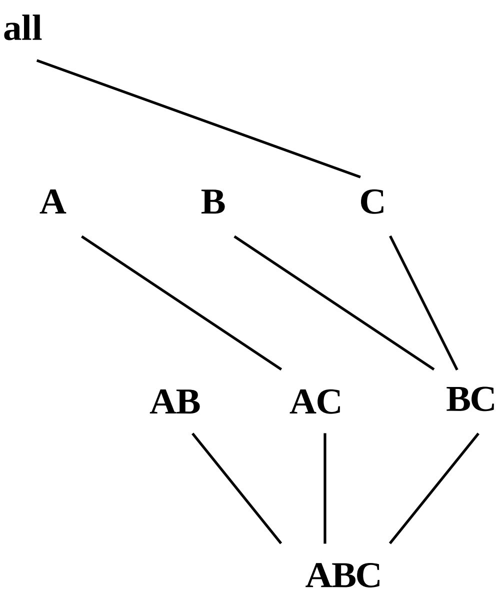
THE UNIVERSITY OF WESTERN AUSTRALIA

- Assume the sizes of dimension, A, B, and C are 40, 400, 4000 respectively.
- Therefore AB is the smallest and BC is the largest 2-D planes
- If chunks are scanned as 1, 2, 3, … then 156,000 memory units are needed (40*400+40*1000+100*1000)
- If chunks are scanned as 1, 17, 33, 49, 5, 21,37 …then 1,641,000 memory units are needed (aggregation ordering AB-AC-BC). Chunk memory units needed are (400*4000 (the whole BC) +10*4000 (one row AC) +10*100 (one chunk of AB))

What is the best traversing order?

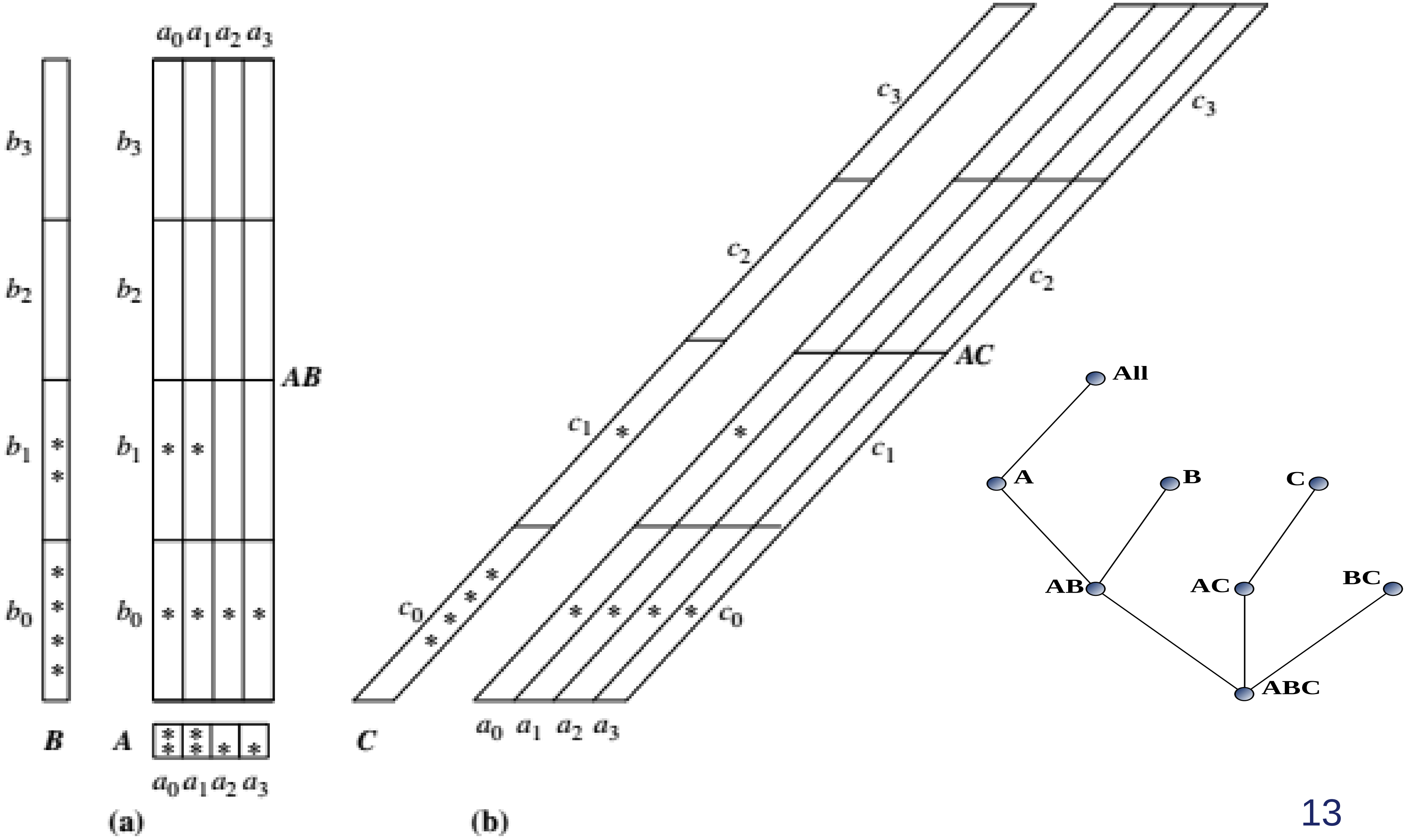


Needs 156,000
Memory units



Needs 1,641,000
Memory units

Example – Multi-way Array Aggregation



Example – cuboids to be computed

- **The base cuboid,**
  - denoted by ABC (from which all the other cuboids are directly or indirectly computed).
  - This cube is already computed and corresponds to the given 3-D array.
- **The 2-D cuboids,**
  - AB, AC, and BC, which respectively correspond to the group-by's AB, AC, and BC.
  - These cuboids must be computed.
- **The 1-D cuboids,**
  - A, B, and C, which respectively correspond to the group-by's A, B, and C.
  - These cuboids must be computed.
- **The 0-D (apex) cuboid,**
  - denoted by all, which corresponds to the group-by ();
  - That is, there is no group-by here.
  - This cuboid must be computed.
  - It consists of only one value.
  - If, say, the data cube measure is count, then the value to be computed is simply the total count of all the tuples in ABC.

Multi-way Array Aggregation

- **Method: the planes should be sorted and computed according to their size in ascending order**
  - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane
- **Limitation of the method: computing well only for a small number of dimensions**
  - If there are a large number of dimensions, “top-down” computation and iceberg cube computation methods can be explored

Lecture Outline

THE UNIVERSITY OF WESTERN AUSTRALIA

- **Efficient Cube Computation**
  - Multi-Way Array Computation
  - **Bottom Up Computation**

- **Unit Review and Final Exam**
  - Exam Structure
  - Review

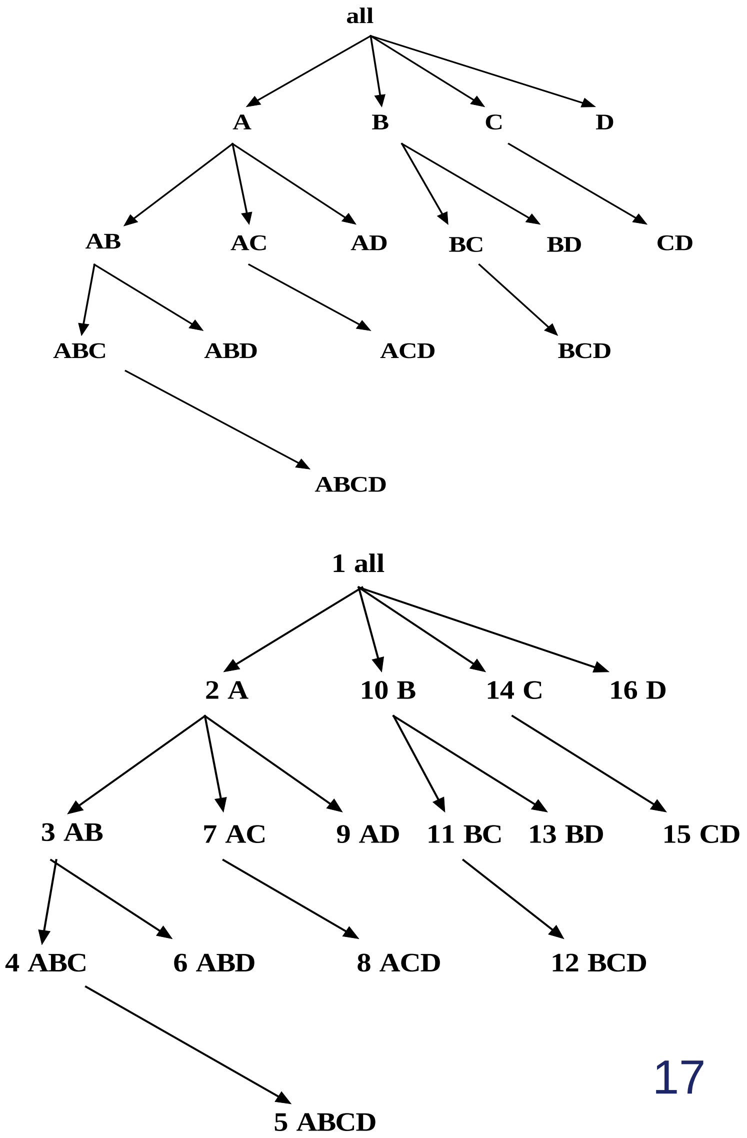
Bottom-Up Computation (BUC)

- **Bottom-up cube computation**

  (Note: top-down in our view!)

- **Divides dimensions into partitions and facilitates iceberg pruning**
  - If a partition does not satisfy min_sup, its descendants can be pruned
  - If $minsup = 1 \Rightarrow$ compute full CUBE!

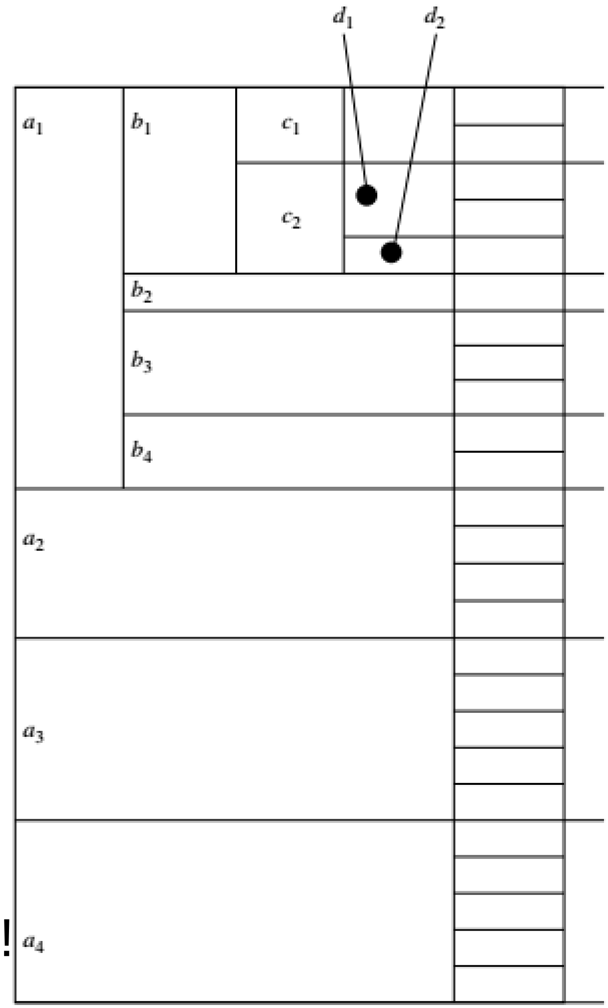
- **No simultaneous aggregation**



BUC Partationing

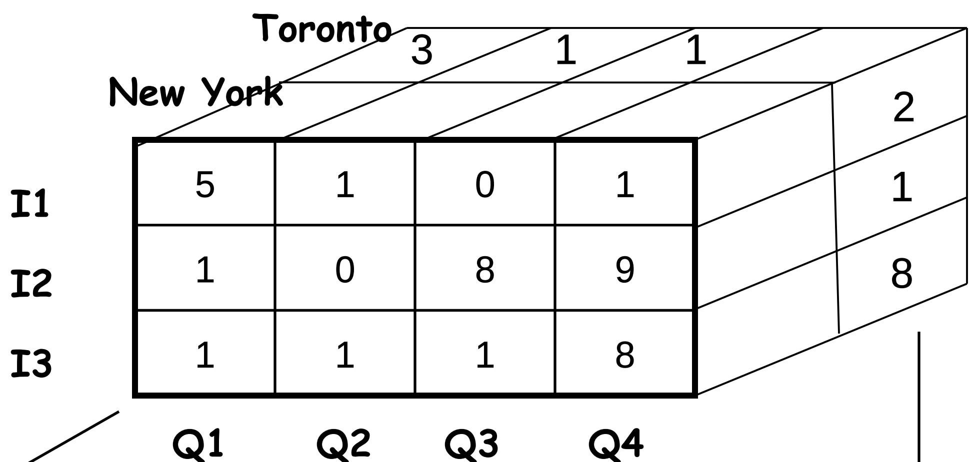
THE UNIVERSITY OF WESTERN AUSTRALIA

- Usually, entire data set can't fit in main memory
- Sort ***distinct*** values
  - partition into blocks that fit
- Continue processing
- Optimisations
  - Partitioning
    - External Sorting, Hashing, Counting Sort
  - Ordering dimensions to encourage pruning
    - Cardinality, Skew, Correlation
    - Higher the cardinality-smaller the partitions-greater pruning opportunity
  - Collapsing duplicates
    - Can't do holistic aggregates anymore!



Ideally the dimension with most discriminative, higher cardinality and having less skew is processed first.

BUC: Example ($having\ count(*) > 5$)



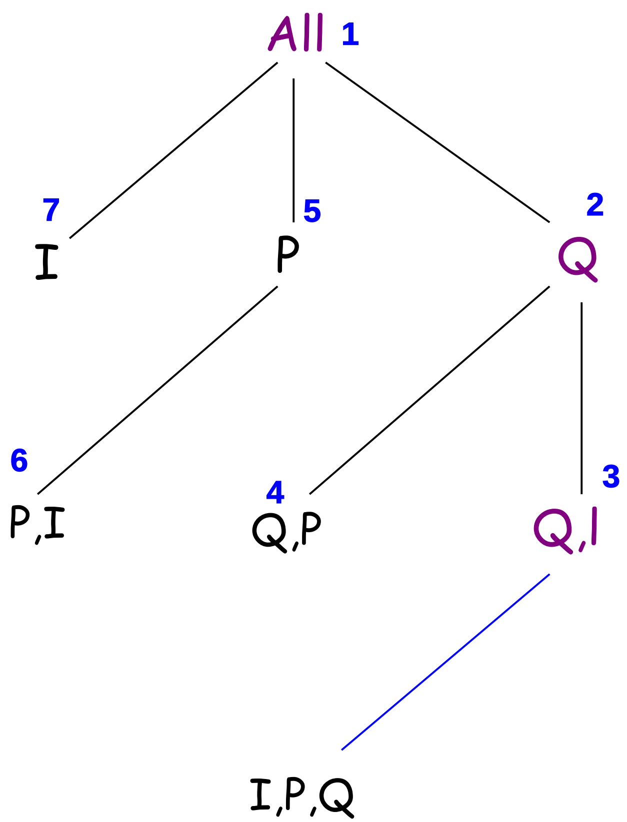
New-York

	Q1	Q2	Q3	Q4
I1	5	1	0	1
I2	1	0	8	9
I3	1	1	1	8

Toronto

	Q1	Q2	Q3	Q4
I1	3	1	1	2
I2	8	1	2	1
I3	2	1	11	8

BUC: Example ($having\ count(*) > 5$)



All

77

Q

Q1	Q2	Q3	Q4
20	5	23	29

Q,I

	Q1	Q2	Q3	Q4
I1	8	2	1	3
I2	9	1	10	10
I3	3	2	12	16

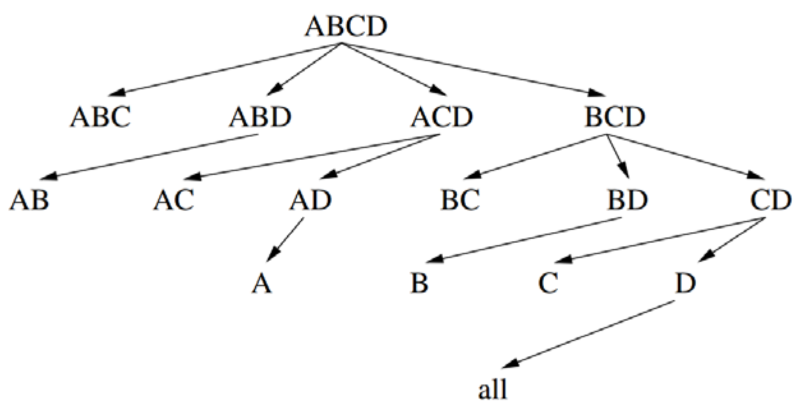
Till Now

Multi-way array aggregation

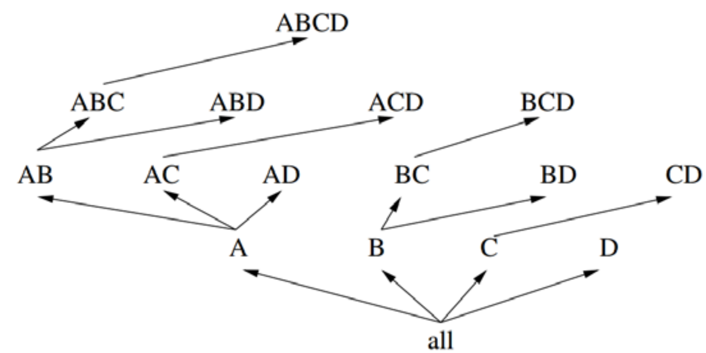
- Aggregates simultaneously on multiple dimensions.
- Multiple cuboids can be computed simultaneously in one pass.

Bottom-up computation

- Facilitates apriori pruning.
- During partitioning, each partition's count is compared with min_sup.
- The recursion stops if the count does not satisfy min_sup.



Top-Down Computation



Bottom-Up Computation

Reference

- Han et al.'s book
  - Chapter 5.
- Readings
  - Iceberg Cube example
  - Star-Cubing algorithm

Lecture Outline

- **Efficient Cube Computation**
  - Multi-Way Array Computation
  - Bottom Up Computation

- **Unit Review and Final Exam**
  - Exam Structure
  - Review

Examination Structure

- **A total of 8 questions (total 60 marks)**
  - Q1: Fact and dimension tables (8 marks)
  - Q2: Data Warehouse Design (5 marks)
  - Q3: Metadata, Data Marts and Data Integration (7 marks)
  - Q4: ETL and Data Quality (10 marks)
  - Q5: Data Warehouses and OLTP Systems (6 marks)
  - Q6: Data Cube and OLAP (7 marks)
  - Q7: Frequent Pattern Mining and Classification (7 marks)
  - Q8: Clustering, Data Reduction and Enhancement (10 marks)

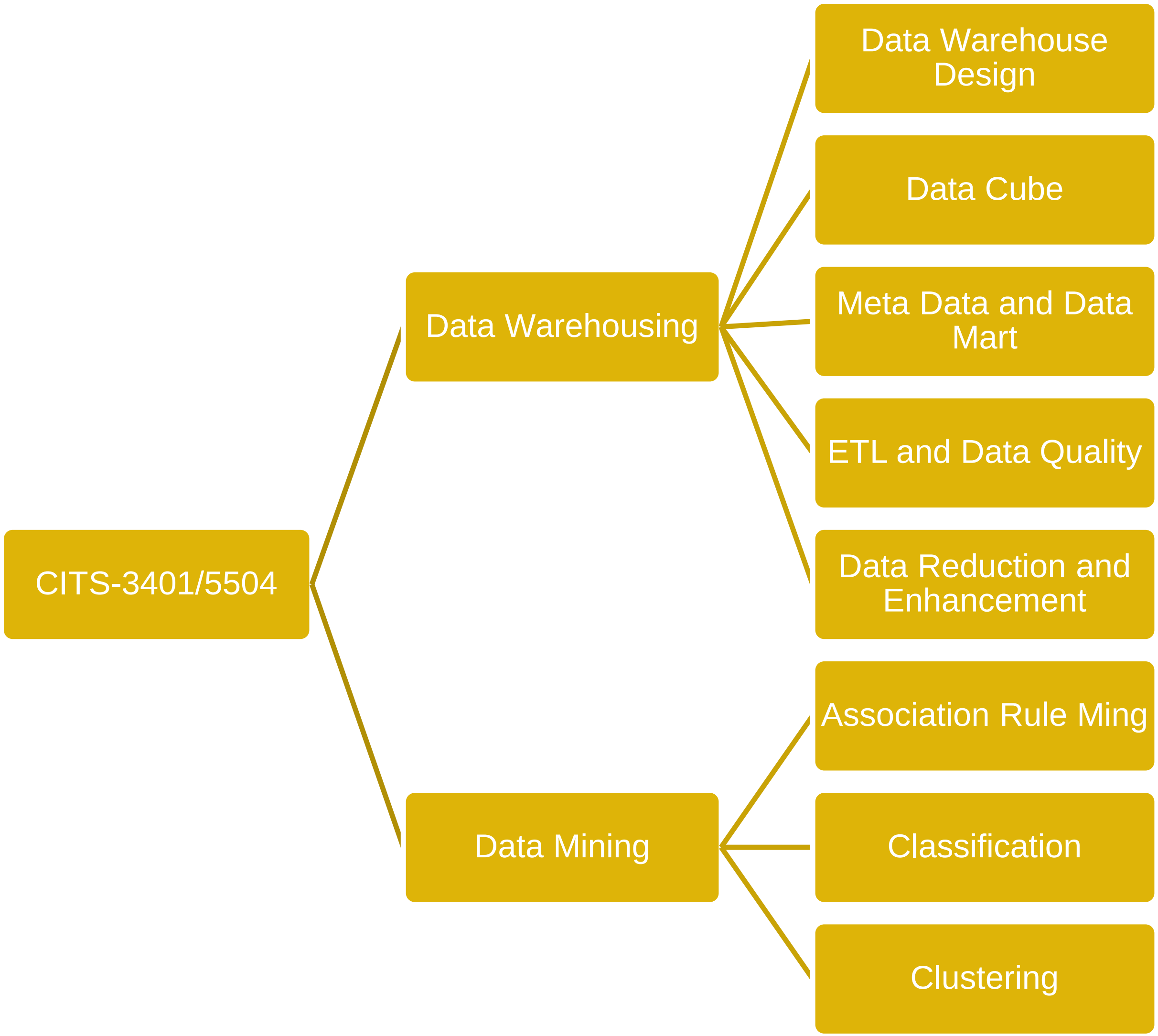
More sub-questions

Examination Structure (Continued)

- **A total of 8 questions (total 60 marks)**
  - **No need calculations** or only need very simple calculation
  - Key Coverage (~80% of the exam content; the other 20% is drawn from Lecture 1-11)
    - OLTP and OLAP, fact tables, dimension tables, business queries, data warehouse schemas, slowly changing dimensions, types of cells and cubes.
    - ETL, storage of data cube, data mart, meta data, data quality.
    - Different types of patterns, association rules, Apriori algorithm, Measures (support, lift, confidence).
    - Information Gain, Gain Ratio, Gini Index, Impurity Reduction, Bayesian Classification, SVMs, Similarity and Dissimilarity, Distances, K-Nearest Neighbours K-Means, K-Medoids, DBScan.

Lecture Overview

- **Basic concepts and overview of the unit**



- **Why Data Warehouse and Data Mining?**
  - **Explosive Growth of Data: from terabytes to petabytes**
  - **We are drowning in data, but starving for knowledge.**
- **What is Data Warehouse and Data Mining?**
  - A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile, collection of data in support of management's decision-making.
  - Extraction of interesting (**non-trivial**, **implicit**, **previously unknown** and **potentially useful**) patterns or knowledge from huge amount of data.
- **OLAP and OLTP**
  - OLTP: Major task of traditional relational DBMS
  - OLAP: Major task of data warehouse system

Lecture 2—Data Modelling

- Storing Data in Data Warehouse
- Fact Tables and Dimension Tables
- Schema of a Data Warehouse
  - Star, Snowflakes, Fact Constellations
- OLAP Operations
  - Roll up, Drill down, Slice & Dice, Pivot

Lecture 3 Data Cube Technologies

- **Data Cube**
  - ➢ Cuboids
  - ➢ Types of Cells
  - ➢ Types of Cubes

- **Answering Queries with Data Cube**

- **Storage of Data Cube**
  - ➢ MOLAP and ROLAP
  - ➢ Cube Materialisation
  - ➢ Indexing Data to Support OLAP

- ETL Overview
- Data Staging
- Data Extraction and Transformation
- Loading Dimension and Fact Tables
- Handling Data Changes

Lecture 5: Dimension Modelling

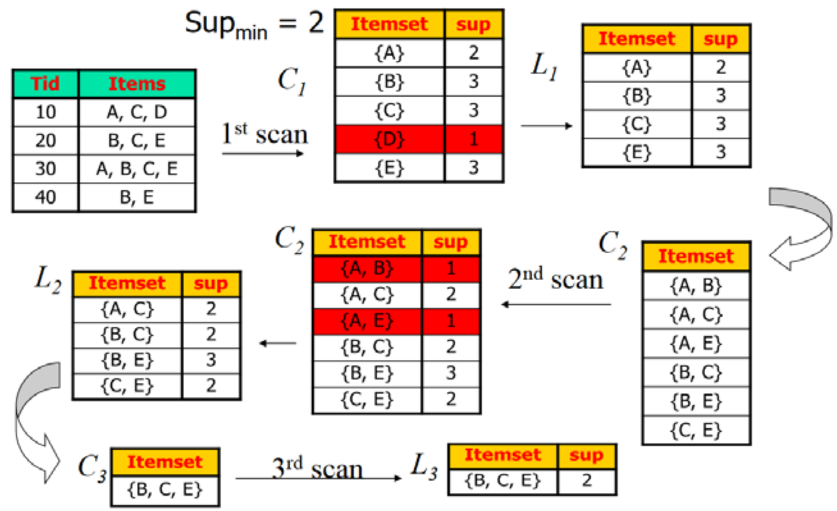
- **Dimension Topics**
  - How many dimensions?
  - Date/Time Dimensions
  - Surrogate Keys
- **Fact Topics**
  - Semi-additive facts
  - “Factless” fact tables
- **Slowly Changing Dimensions**
  - Overwrite history, preserve history, hybrid schemes
- **More dimension topics**
  - Dimension roles
  - Junk dimension
- **More fact topics**
  - Multiple currencies
  - Master/Detail facts and fact allocation

Lecture 6-7: Meta Data, Data Mart and Data Quality

- Meta Data, Why Data Mart
- Data Quality
  - ✓ Data Quality Overview and Examples
  - ✓ Cleansing and Matching
  - ✓ Data Quality Validation
- Data Profiling
- Attribute Types and Similarity Measure

Lecture 8: Association Rule Mining

- **Concepts**
  - Support
  - Confidence
  - Lift



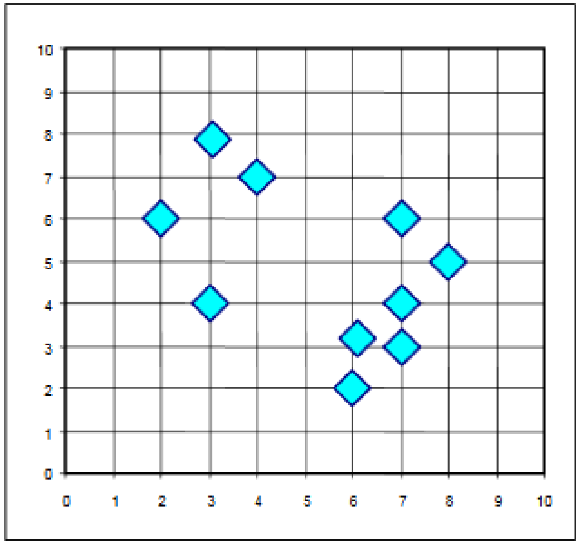
- **Frequent itemsets and association rules.**
  - Frequent Patterns, Closed Patterns and Max-Patterns
  - The Apriori Algorithm
  - How to generate the association rules

Lecture 9: Classification

age	income	student	credit_rating	buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_age	high	no	fair	yes
senior	medium	no	fair	no
senior	low	yes	fair	no
senior	low	yes	excellent	yes
middle_age	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_age	medium	no	excellent	yes
middle_age	high	yes	fair	yes
senior	medium	no	excellent	yes

- **Ranking attributes**
  - Information Gain
  - Gain Ratio
  - Gini Index
- **Advanced classification methods**
  - Bayesian classification, k-nearest neighbours
  - SVMs and neural networks
- **Evaluation of classification models**

Lecture 10 – Clustering Algorithms



- **Conceptual Understanding**
  - Partition based clustering
    - K-means
    - K-medoids
  - Density-based clustering
    - DBScan
  - Distance measures for different types of attributes

- **Understanding the advantage and disadvantage of the clustering algorithms.**

Lecture 11—Data Reduction & Enhancement

- **Data reduction**
  - Attributes (Feature Reduction)
    - Discrete Wavelet Transform (Haar Wavelet Transform example)
    - Principal Component Analysis (high level understanding)
    - Attribute Subset Selection
  - Instances (Numerosity Reduction)
    - Parametric methods
      - A model (regression or log-linear models) is used to estimate the data
      - Only model parameters are stored
    - Non-parametric methods
      - Histogram, Clustering, Sampling, Data Cube Aggregation
    - Data Compression
- **Data Enhancement: Augmentation and Oversampling**

Lecture 12 – Data Cube Computation

- **Efficient Computation of Data Cubes (not examinable)**
  - Multiway Array Aggregation
  - BUC
- **Exam Structure and Unit Review**

Good luck for final projects and exams!

- **Survey: This [link] to the SURF and SPOT survey**
  - Welcome positive comments and constructive criticism.
- **CITS3402/5507 High-Performance Computing**
  - Looking for lab facilitators
- **Data Mining and Machine Learning research projects**
  - Looking for students and Research Assistants
  - The candidates should have good programing skills, and be comfortable of solving technical problems.
- **More information about my research**
  - https://zeyiwen.github.io/

My Recent Research Work (1)

- Solving a text mining problem (i.e. customer review sentiment analysis); techniques used:
  - SVMs, k-means, data split, feature selection, model evaluation

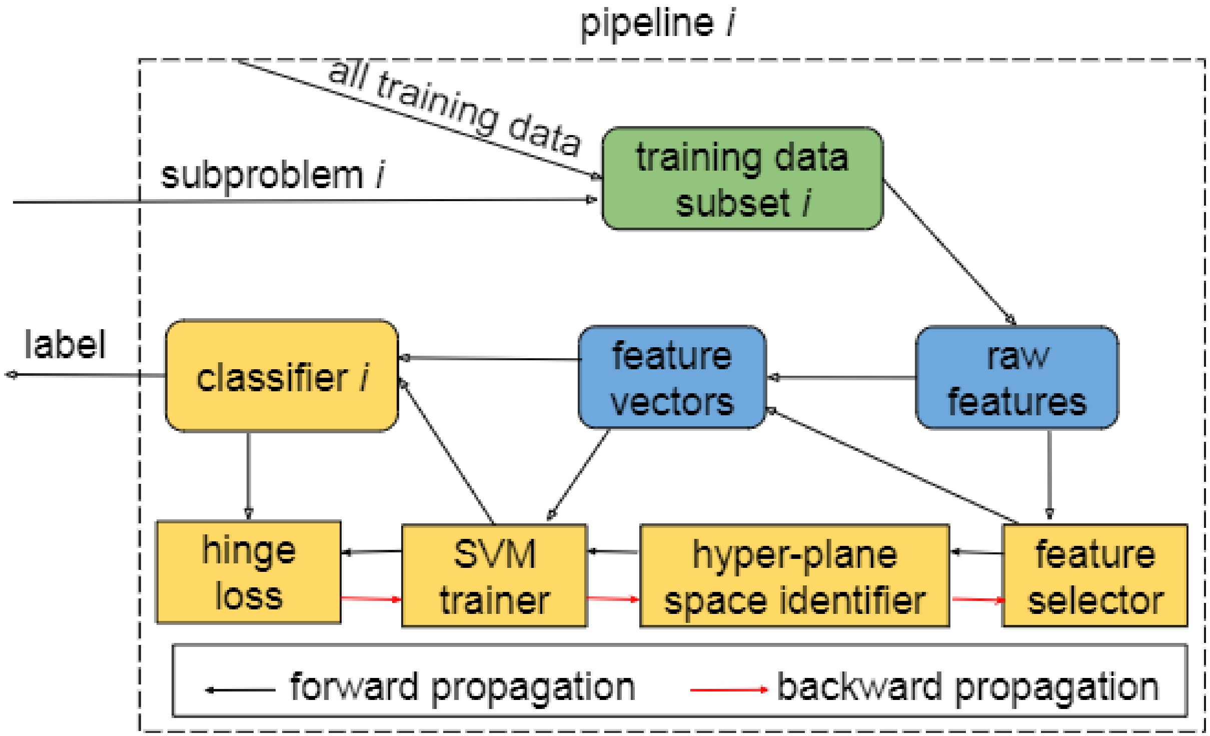


Figure 1: The pipeline of SVM training for a subproblem

Results (1)

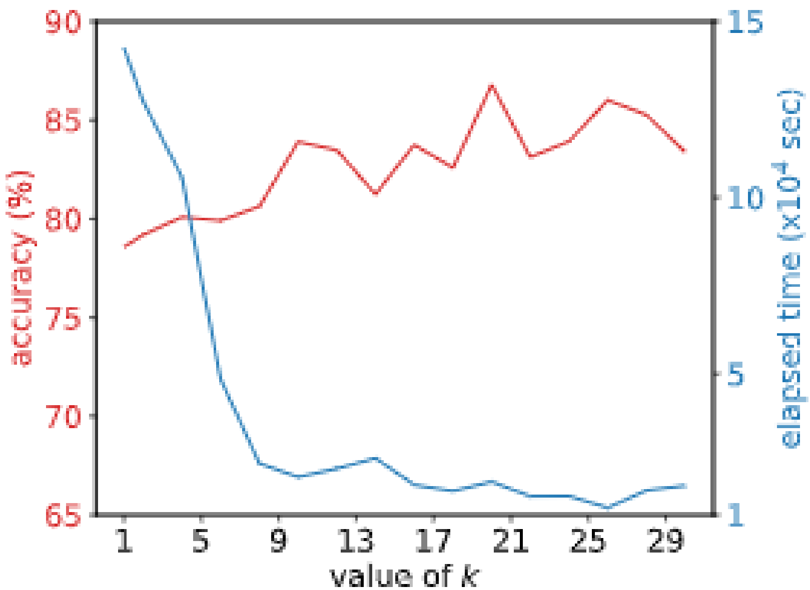
Table 5: Accuracy and Macro-F_1 comparison

Models		Restaurant Acc	Restaurant Macro-F_1	Laptop Acc	Laptop Macro-F_1
BERT based Model	LCF-ATEPC [30] **(need extra labeled data)**	**90.18**	**85.88**	82.29	**79.84**
	LCF-BERT [31]	87.14	81.74	**82.45**	79.59
	BERT-SPC [23]	*84.46*	*76.98*	*78.99*	*75.03*
	SDGCN-BERT [32]	*83.57*	*76.47*	*81.35*	*78.34*
	AEN-BERT [23]	*83.12*	*73.76*	*79.93*	*76.31*
	BERT-PT [29]	*84.95*	*76.96*	*78.07*	*75.08*
Neural Model	HAPN [13]	82.23	-	**77.27**	-
	IMN [7]	**83.89**	**75.66**	75.36	**72.02**
	BILSTM-ATT-G [4]	81.11	72.19	75.44	70.52
	RAM [3]	80.23	70.80	74.49	71.35
	LSTM+SynATT+TarRep [6]	80.63	71.32	71.94	69.23
	PF-CNN [8]	79.20	-	70.06	-
SVM-based Model	existing SVM approach [11]	82.23	73.75	72.27	65.60
	ours (single SVM)	78.57	63.78	72.26	67.61
	ours (multiple SVMs)	**86.79**	**78.81**	**80.25**	**77.07**
Replaced SVMs with BERT	ours (multiple BERTs)	75.98	61.0	62.69	61.0

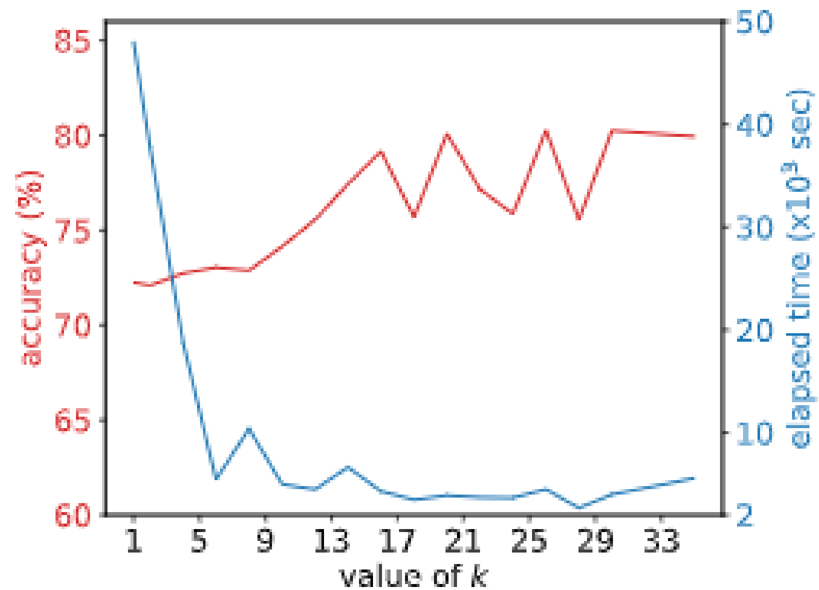
Wen, Zeyi, et al. Enhancing SVMs with Problem Context Aware Pipeline. The 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2021.

Effect of *k* in *k*-means (1)

- **“Restaurant” and “Laptop” are two data set names.**



(a) ***Restaurant***



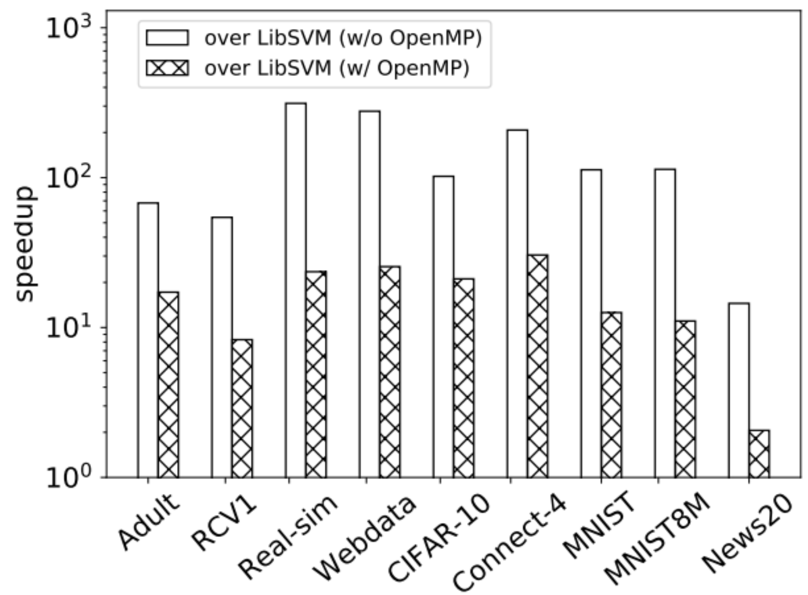
(b) ***Laptop***

My Research Work (2)

- **Aim: making machine learning algorithms run faster**
  - Techniques: parallel computing, algorithm optimisation, hardware
- **Two libraries**
  - ThunderSVM: A Fast SVM Library on CPUs and GPUs
    - https://github.com/zeyiwen/thundersvm
  - ThunderGBM: Fast Gradient Boosting Decision Trees on GPUs
    - https://github.com/zeyiwen/thundergbm

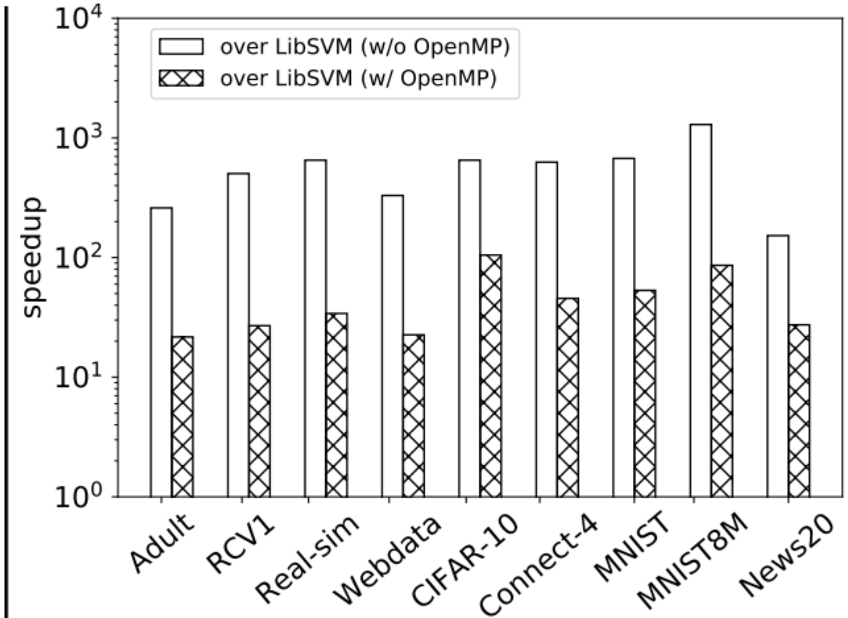
Results (2)

ThunderSVM vs. LibSVM



Training

~10 to 100 times faster



Prediction

~100 times faster

the models are the same as LibSVM

Outcomes of ThunderSVM (2)

THE UNIVERSITY OF WESTERN AUSTRALIA

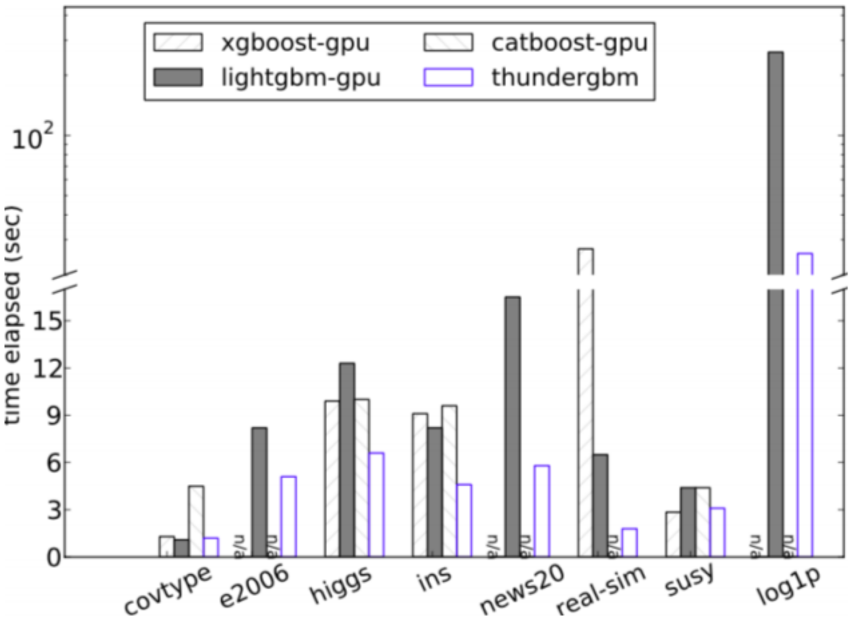
- ThunderSVM: A Fast SVM Library on GPUs and CPUs
  - https://github.com/zeyiwen/thundersvm
  - 1300+ stars, 170+ forks
  - Publications: JMLR’18 [1] and TKDE’18 [2]

[1] Wen, Zeyi, et al. "ThunderSVM: A fast SVM library on GPUs and CPUs." Journal of Machine Learning Research (JMLR), 19.1: 797-801, 2018.

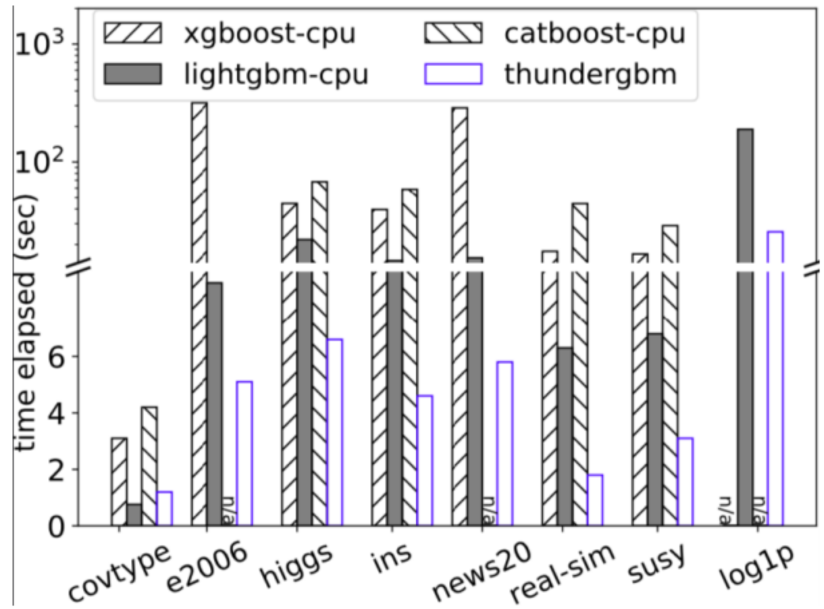
[2] Wen, Zeyi, et al. "Efficient Multi-Class Probabilistic SVMs on GPUs." IEEE Transactions on Knowledge and Data Engineering (TKDE), 2018.

Results (3)

Training is faster and scalable



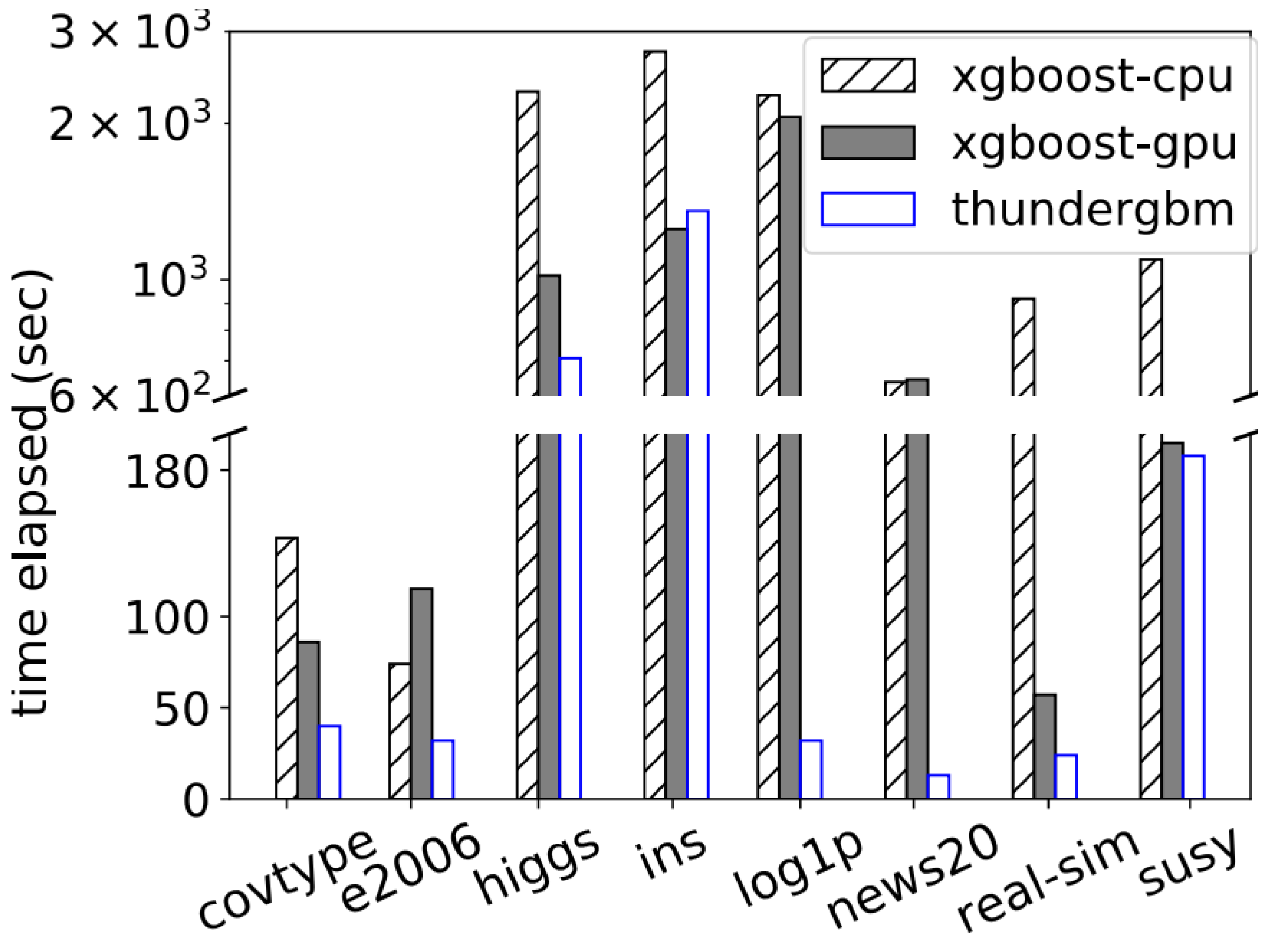
faster and more scalable



~10 times faster

The models are the same as XGBoost

Prediction is also faster



Outcomes of ThunderGBM

- ThunderGBM: Fast Gradient Boost Decision Trees on GPUs
  – https://github.com/zeyiwen/thundergbm
  – 590+ stars, 70+ forks
  – Publications: IPDPS'18 [3], TPDS'19 [4] and JMLR'20 [5]

[3] Wen, Zeyi, et al. "Efficient Gradient Boosted Decision Tree training on GPUs." IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2018.
[4] Wen, Zeyi, et al. "Exploiting GPUs for efficient Gradient Boosting Decision Tree training." IEEE Transactions on Parallel and Distributed Systems (TPDS), 2019.
[5] Wen, Zeyi, et al. "ThunderGBM: Fast GBDTs and Random Forests on GPUs." Journal of Machine Learning Research (JMLR), 2020.

Copyright Notice

Copyright Notice