GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMIA Y COMPETITIVIDAD

UNIÓN EUROPEA
Fondo Europeo de Desarrollo Regional
"Una manera de hacer Europa"

S.I.M.D.
Sistemas Inteligentes y Minería de Datos

JACINTO ARIAS - UCLM

LARGE SCALE BAYESIAN NETWORKS ON HIGHLY DISTRIBUTED COMPUTING FRAMEWORKS

Granada - Febrero 2016

STATE-OF-THE-ART OF HIGH PERFORMANCE AND CLOUD COMPUTING

Traditional Clusters, MapReduce, Apache Spark, Cloud Platforms and Functional Programming

"TRADITIONAL SYSTEMS"

- Hardware constrains:
  - Required efficient languages.
  - Required specialised compilers.
  - Required further optimisation of the code.

C FORTRAN … MPI OPENMP …

ADVANTAGES OF CLOUD COMPUTING

- New Hardware advantages:
  - It's Cheap
  - It's Transparent
  - It's Elastic/Scalable
  - Independent Data Storage

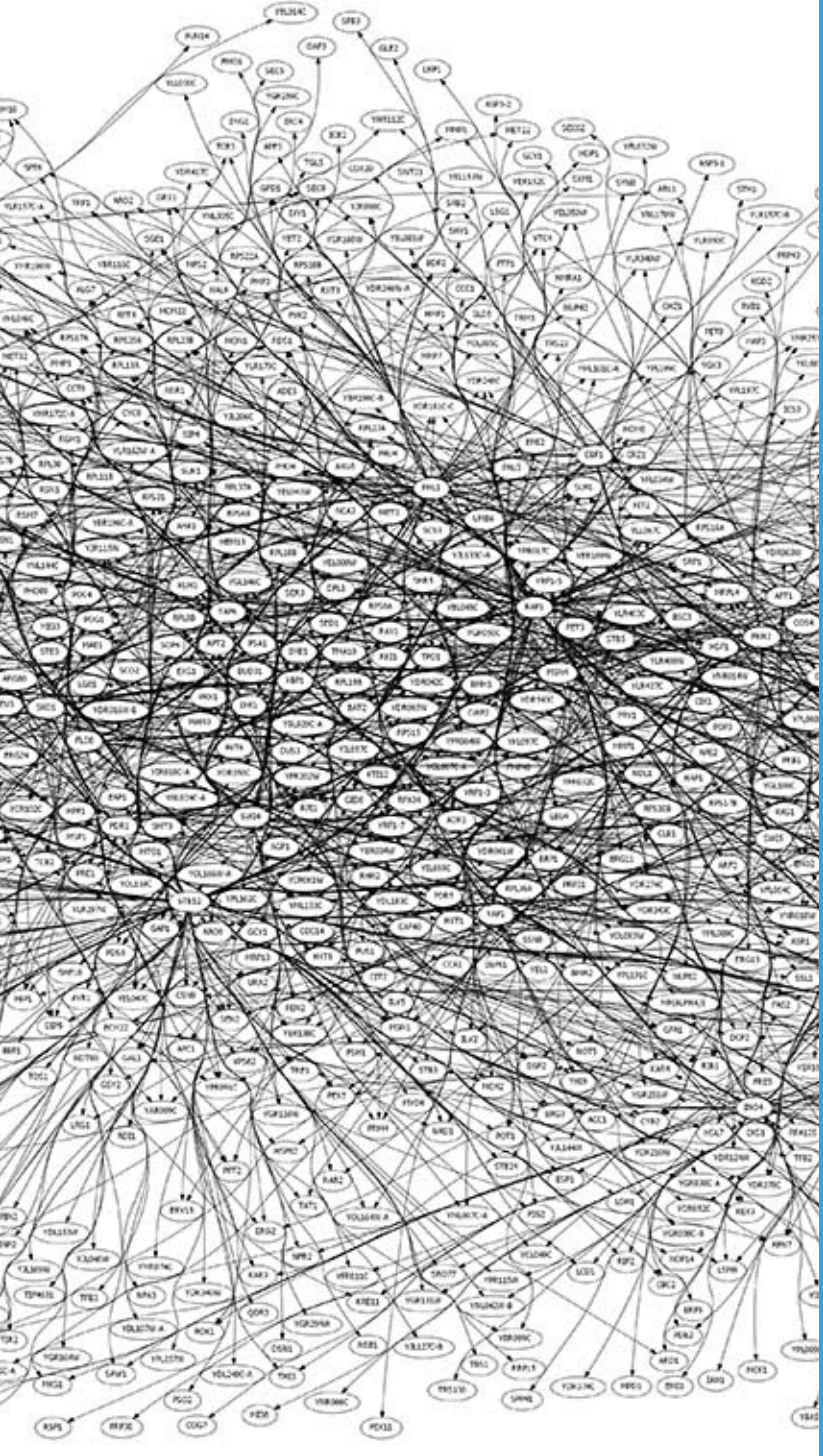
Sistemas Inteligentes y Minería de Datos

DISTRIBUTED (PARALLEL) COMPUTING ECOSYSTEM

Traditional HPC and GPUs

"Classic" MapReduce

Memory based Distributed computing & "modern" language interfaces



"DISTRIBUTED DISCRETE BAYESIAN NETWORK CLASSIFIERS UNDER MAPREDUCE WITH APACHE SPARK"

IEEE BigData Software and Engineering

(Helsinki, Aug 2015)

CAEPIA'15

(Albacete, Nov 2015)

Knowledge-Based Systems (Target submission)

(Special Issue on Volume, Variety and Velocity of Data Sciences)

MOTIVATION AND IMPACT

Java

- Logistic Regression
- Decision Trees
- Neural Network
- Ensembles
  - Bagging
  - AdaBoost
  - Random Forests
  - Rotation Forests
  - Random Subspaces
- kNN
- SVM
- **Bayes**
  - Naive Bayes
  - TAN
  - AODE / A2DE
  - BayesNet
  - KDB
- (...)

Python

- Logistic Regression
- **Naive Bayes**
- Perceptron
- Ensembles
  - Bagging
  - AdaBoost
  - Random Forests
- Decision Trees
- kNN
- SVM
- SGD
- (...)

~~Hadoop~~ Spark

- Logistic Regression
- **Naive Bayes**
- Multilayer Perceptron
- Random Forests
- Hidden Markov Models

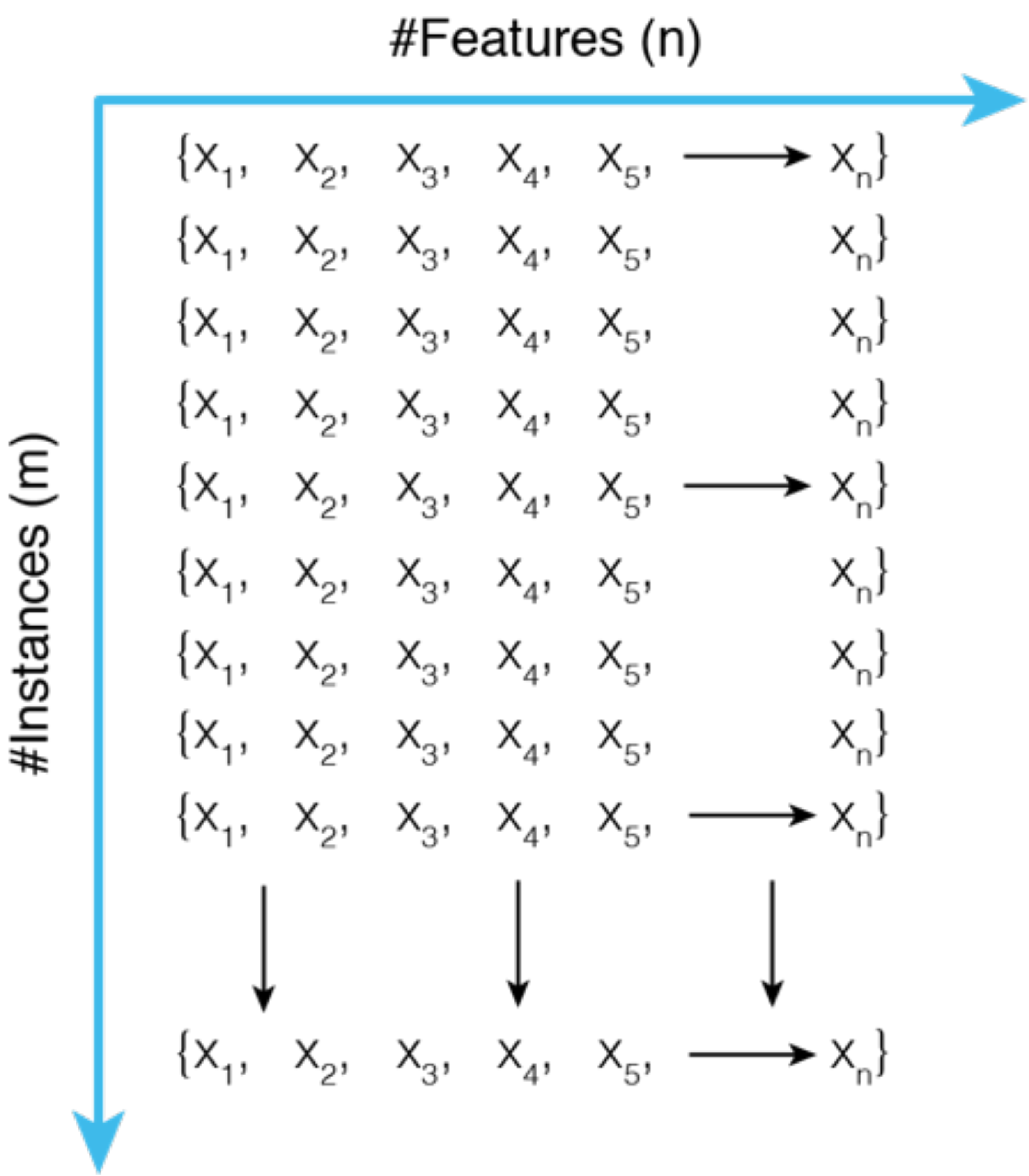
(Apache Spark)

- Linear SVM
- Logistic Regression
- Decision Trees
- **Naive Bayes**
- Random Forests

Java, R, Python

- GLM
- GBM
- Decision Trees
- **Naive Bayes**
- Random Forests
- Deep Learning

SUPERVISED CLASSIFICATION SCALABILITY



Big Data (>>m)

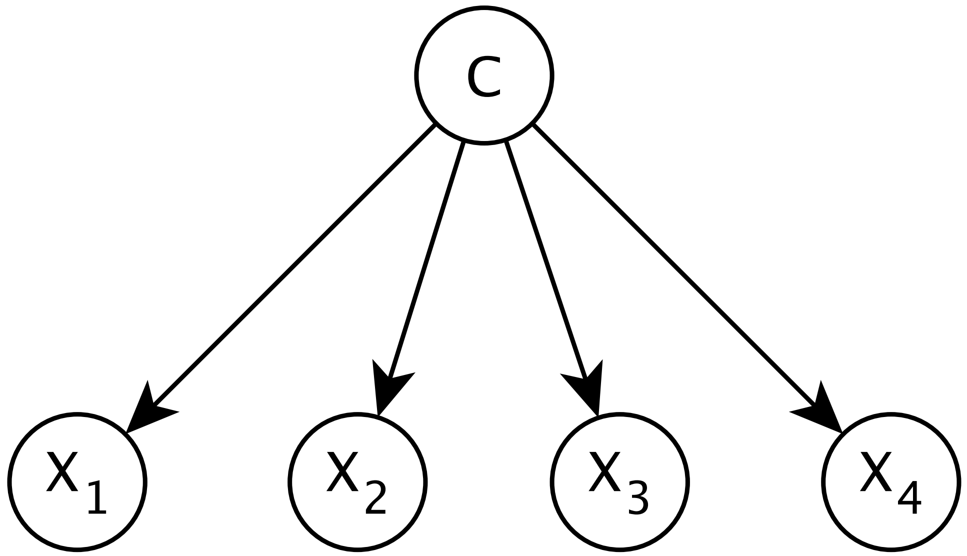
- High storage demands
- Does not fit in main memory
- Long execution times due to intensive disk reading.

High Dimensional (>>n)

- Increase complexity of models
- Increase size of models

AUGMENTED NAIVE BAYESIAN CLASSIFIERS

NAIVE BAYES



Complexity

Time: O(n·m)

Space: O(n·c·v)

Model: O(n·c·v)

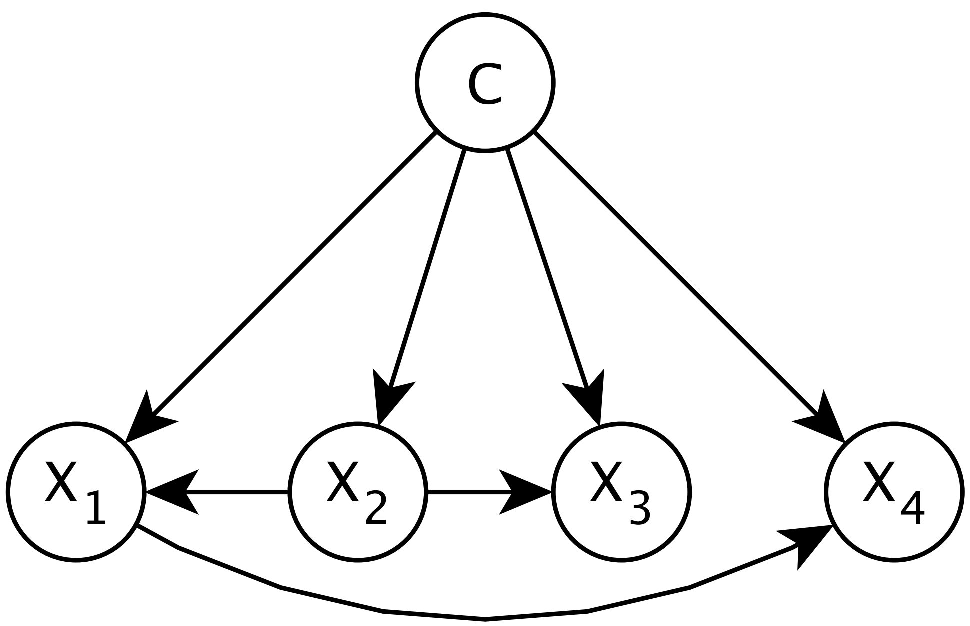
Passes: 1 data pass

Learning

- Fixed Structure
- Only parameter estimation

AUGMENTED NAIVE BAYESIAN CLASSIFIERS

TAN



Complexity

Time: $O(n^2 \cdot m + n^2 \log n + n)$

Space: $O(n^2 \cdot c \cdot v^2)$

Model: $O(n \cdot c \cdot v^2)$

Passes: 1 data pass

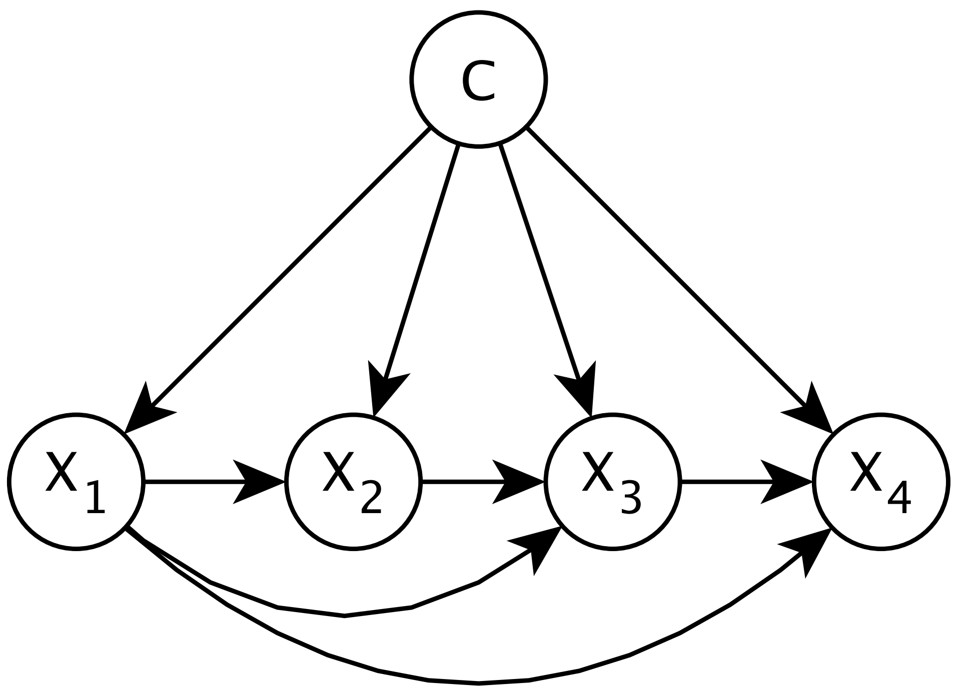
Learning

- Estimate $MI(X_i \mid X_j, C)$ for all pairs
- Build MST with Chow-Liu's, select root, add Class
- Estimate parameters (reuse counts)

S.I.M.D.
Sistemas Inteligentes y Minería de Datos

AUGMENTED NAIVE BAYESIAN CLASSIFIERS

KDB



Complexity

Time: $O(n^2 \cdot m + n \cdot \log n + n \cdot m)$

Space: $O(n^2 \cdot c \cdot v^2)$

Model: $O(n \cdot c \cdot v^k)$

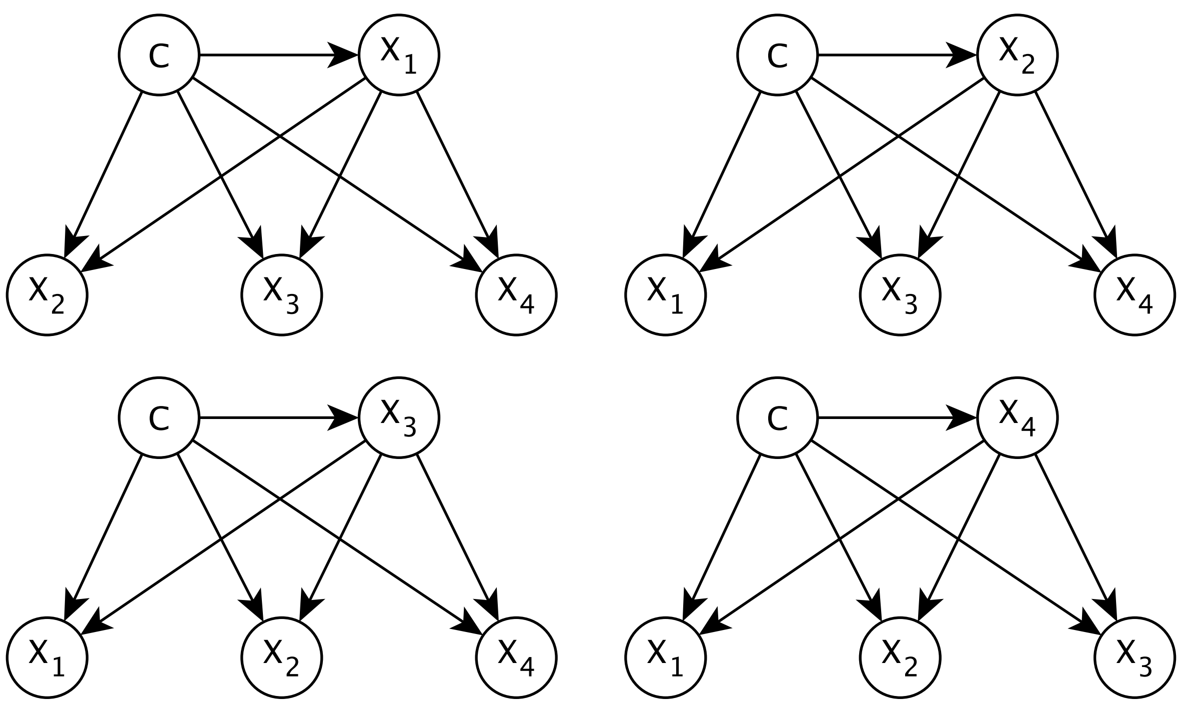
Passes: 2 data passes

Learning

- Estimate $MI(X_i \mid C)$ for all attributes and order ascending
- For each X_i get best k previous vars with max $MI(X_i \mid X_j, C)$
- Estimate parameters (cannot reuse counts for $k>1$)

AUGMENTED NAIVE BAYESIAN CLASSIFIERS

A1DE



Complexity

Time: $O(n^2 \cdot m)$

Space: $O(n^2 \cdot c \cdot v^2)$

Model: $O(n^2 \cdot c \cdot v^2)$

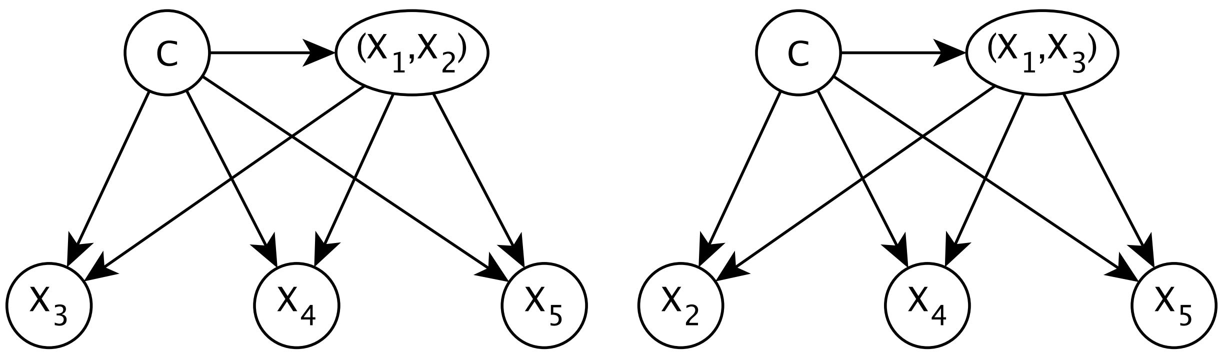
Passes: 1 data pass

Learning

- Fixed structure
- Estimate the parameters for all models in the ensemble

AUGMENTED NAIVE BAYESIAN CLASSIFIERS

A2DE



...

Complexity

Time: $O(n^3 \cdot m)$

Space: $O(n^3 \cdot c \cdot v^3)$

Model: $O(n^3 \cdot c \cdot v^3)$

Passes: 1 data pass

Learning

- Fixed structure
- Estimate the parameters for all models in the ensemble

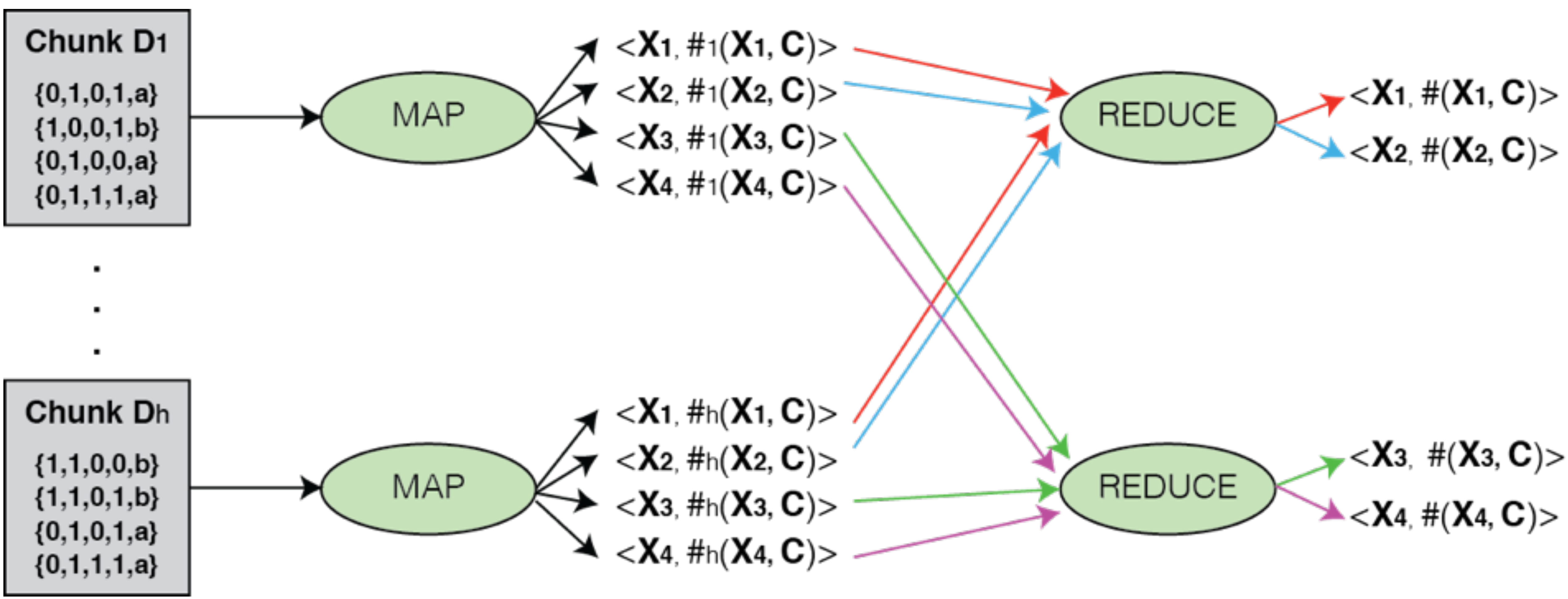
PARALLELIZATION STRATEGY

- Either for structural or parametric learning the main bottleneck is estimating the Joint Frequency Distributions.
- This builds a (k+1)-dimensional contingency table involving k attributes and the class which in general involves:

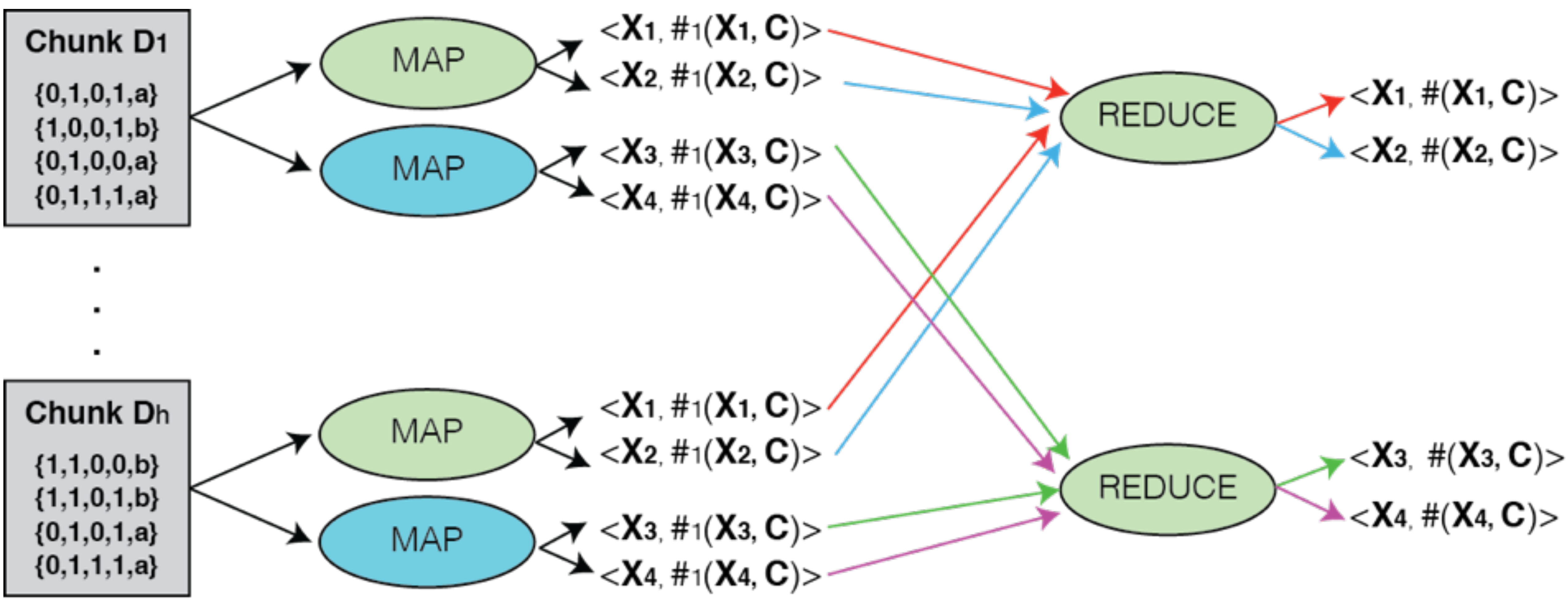
$$O(n^k \cdot m)$$

- Horizontal Parallelism: Distribute the counts (m is large)
- Vertical Parallelism: Distribute attribute combinations (n is large)

HORIZONTAL PARALLELISM (NAIVE BAYES)

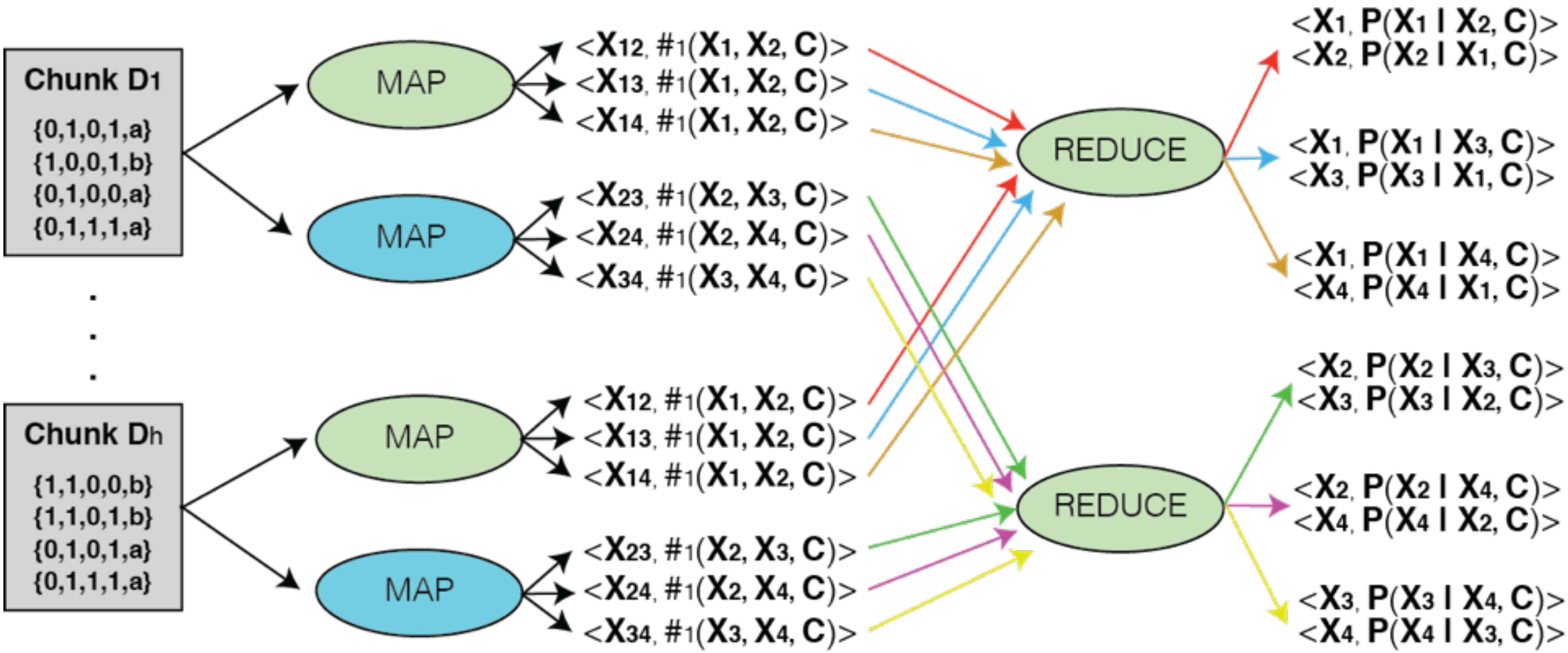


HORIZONTAL+VERTICAL PARALLELISM (NAIVE BAYES)



S.I.M.D.
Sistemas Inteligentes y Minería de Datos

HORIZONTAL+VERTICAL PARALLELISM (AODE)



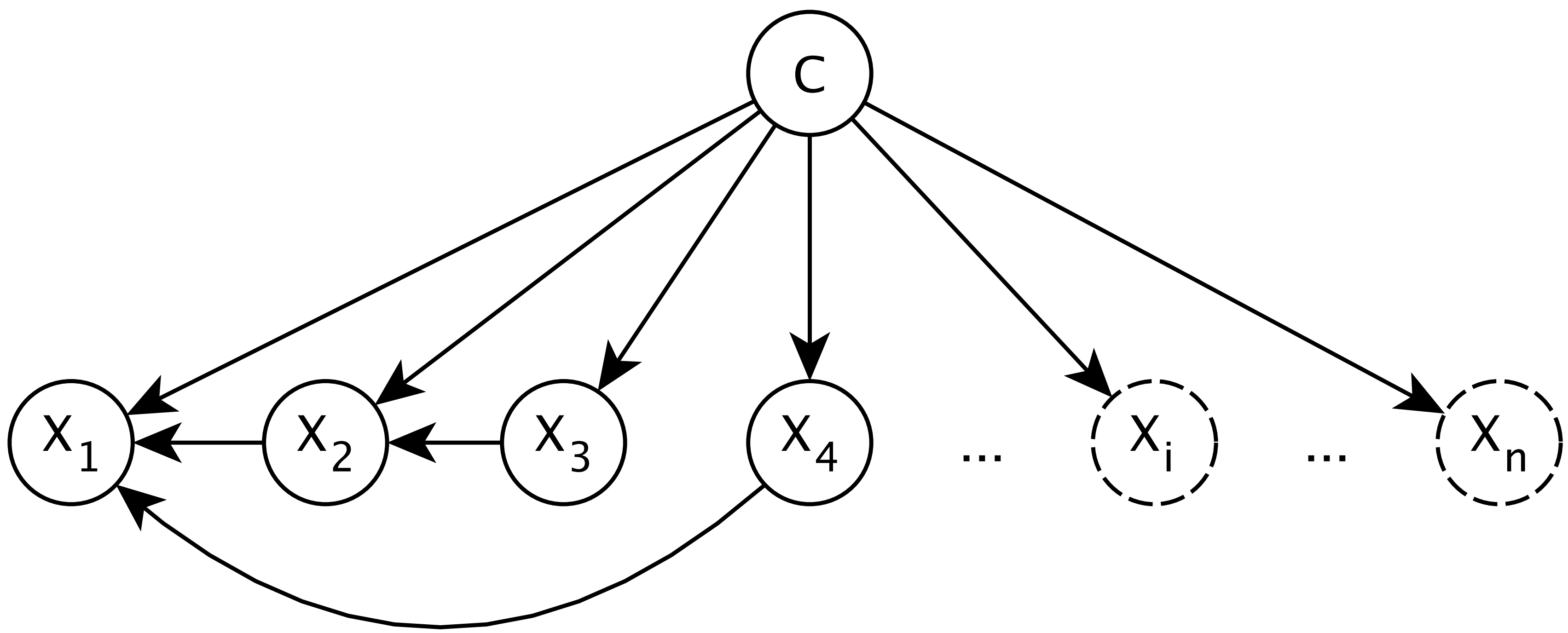
S.I.M.D.
Sistemas Inteligentes y Minería de Datos

COMPUTING ENVIRONMENT

- Our "BigSimd" cluster:
  - 7 nodes (1 master + 6 slaves)
  - Dual Intel Xeon E5-2609v3 1.90GHz hexacore processors (72 cores)
  - 64GB Main Memory
  - 4x1TB Disks
  - Apache Spark 1.6 + Apache Hadoop 2.6 (Cloudera cdh5.5)

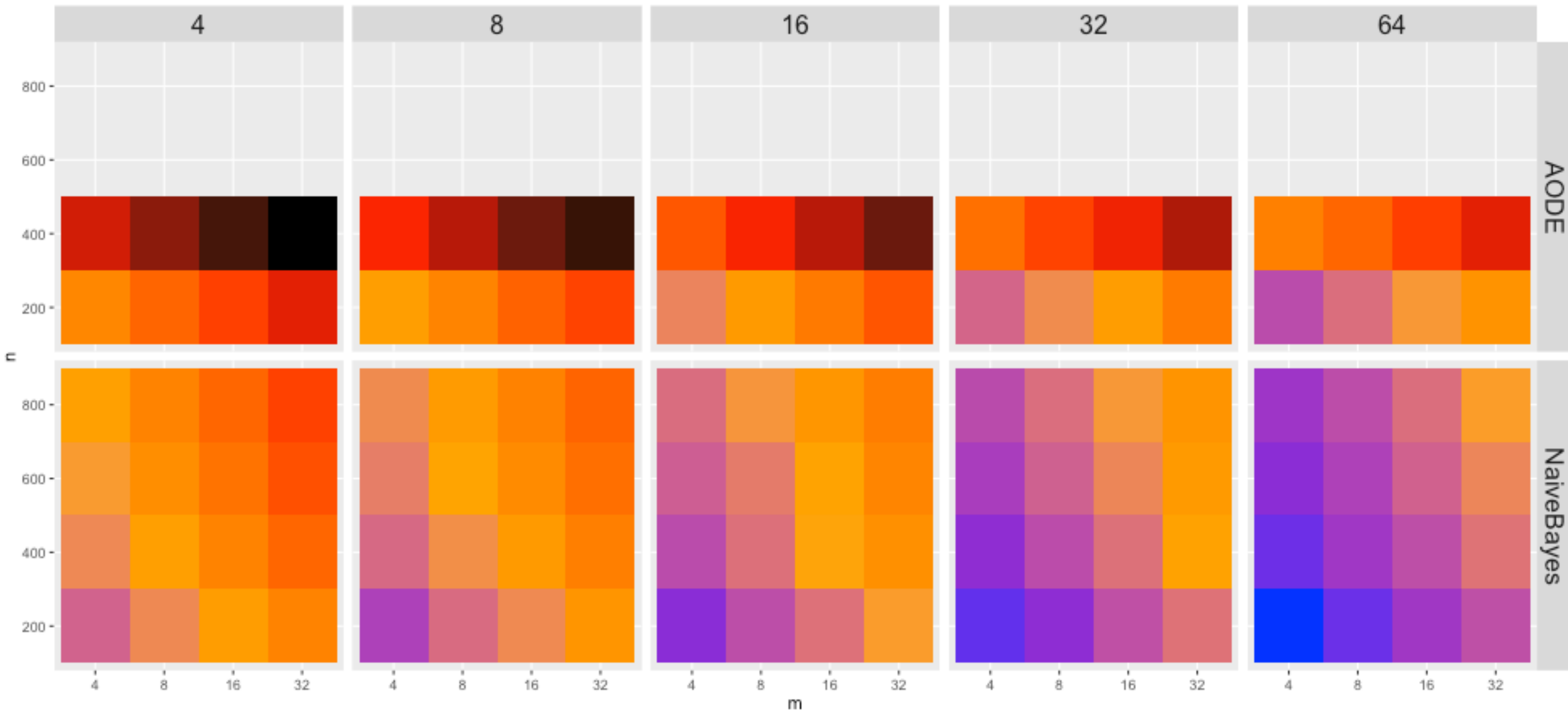
S.I.M.D.
Sistemas Inteligentes y Minería de Datos

ONGOING EXPERIMENTS ON SYNTHETIC DATA

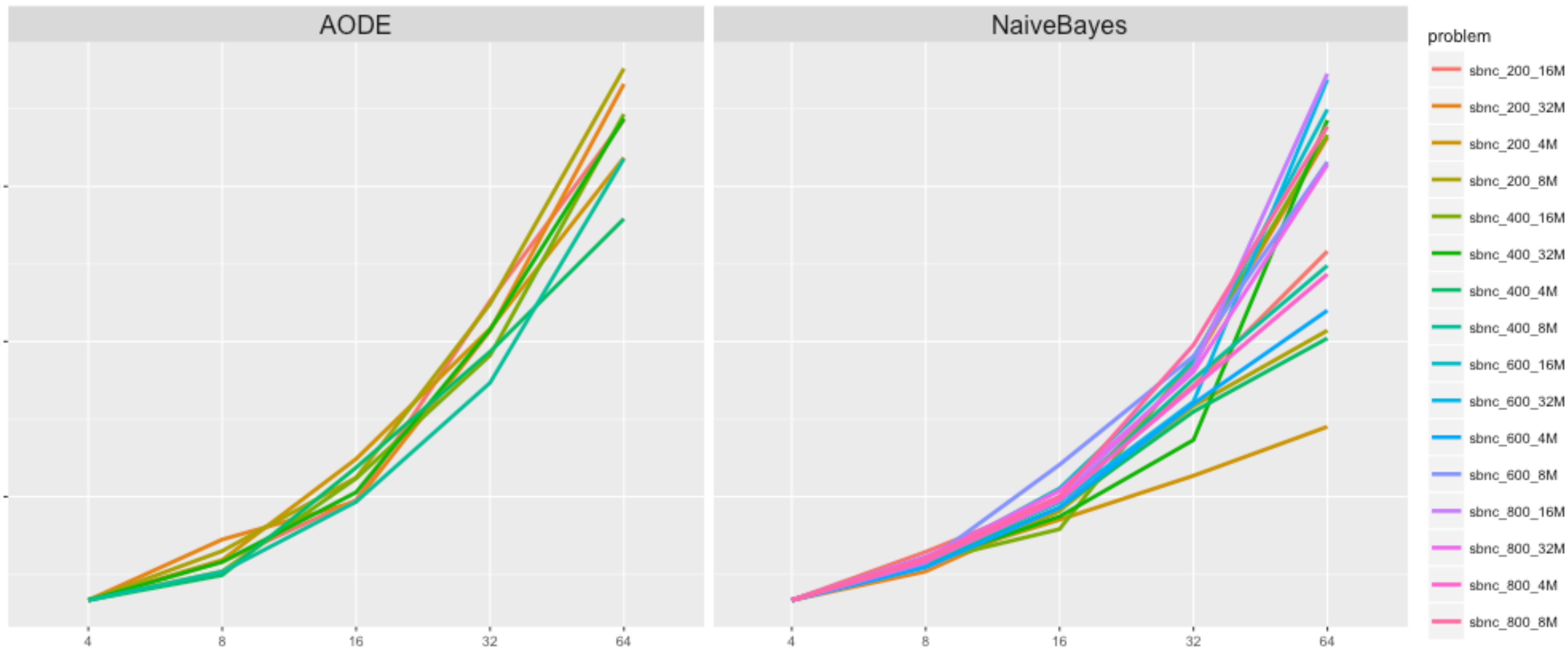


n \ m	4M	8M	16M	32M
200	1.6GB	3.2GB	6.4GB	12.8GB
400	3.2GB	6.4GB	12.8GB	26.6GB
600	4.8GB	9.6GB	19.2GB	38.4GB
800	6.4GB	12.8GB	25.6GB	51.2GB

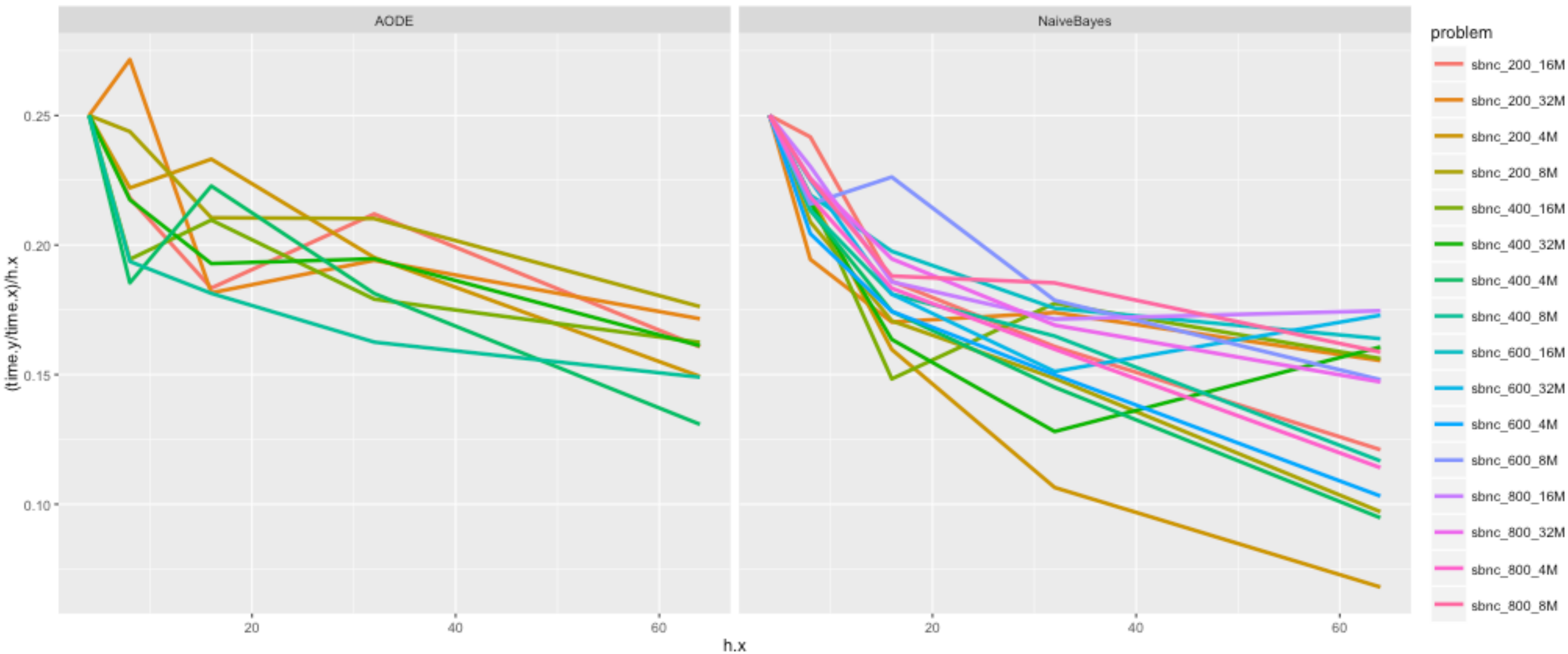
EXECUTION TIME (LOG SCALE)



SPEED-UP (AGAINST 4 TASKS)



EFFICIENCY (AGAINST 4 TASKS)



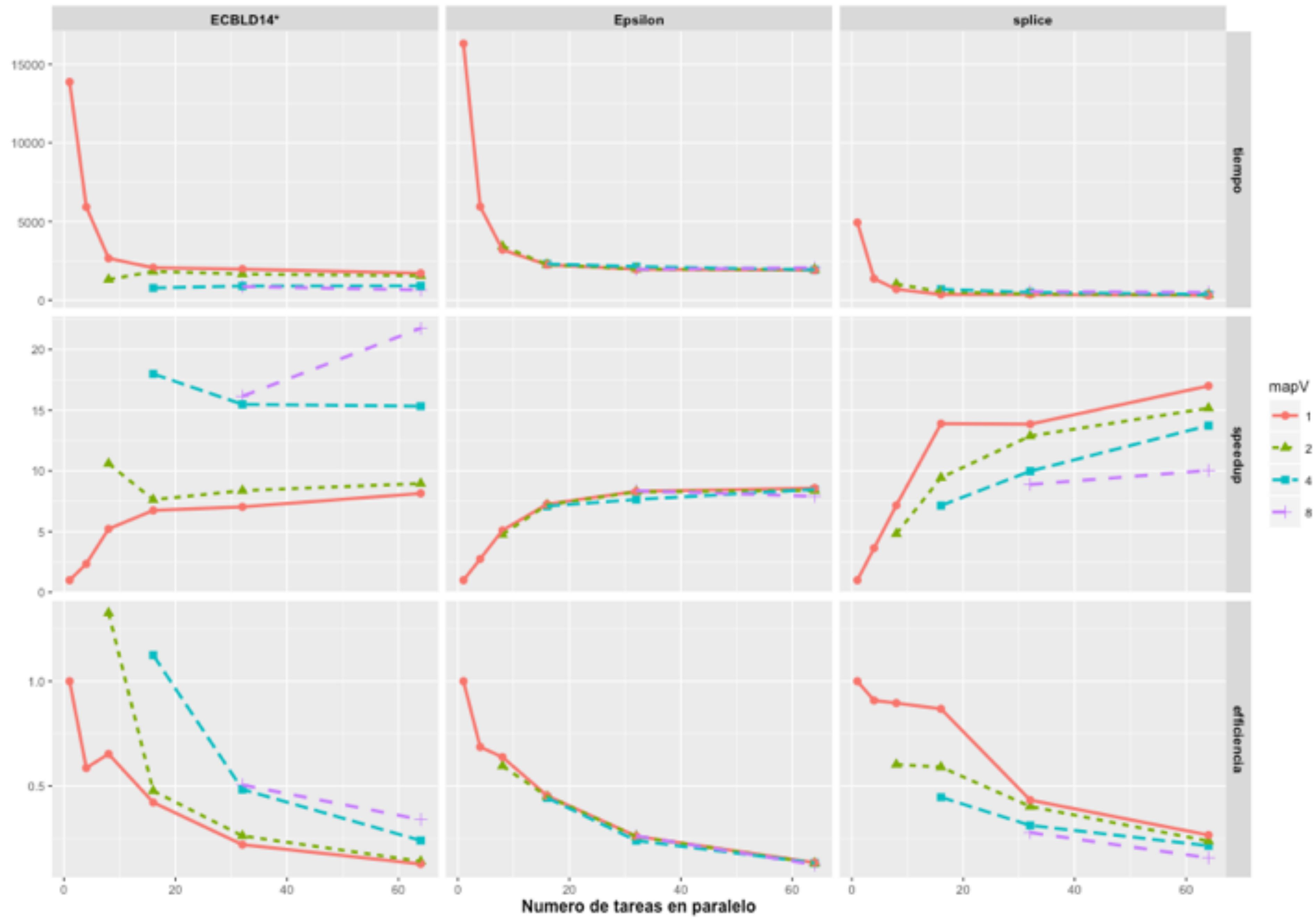
PAST EXPERIMENTS ON REAL DATA

	#Attributes	#Instances	Size
SPLICE	141	50M	14GB
ECBLD'14	630	4.3M	5GB
EPSILON	2000	500k	1.9GB

- Horizontal: 4, 8, 32, 64
- Vertical: 1, 2, 4, 8
- Sequential: Optimized Weka+Moa

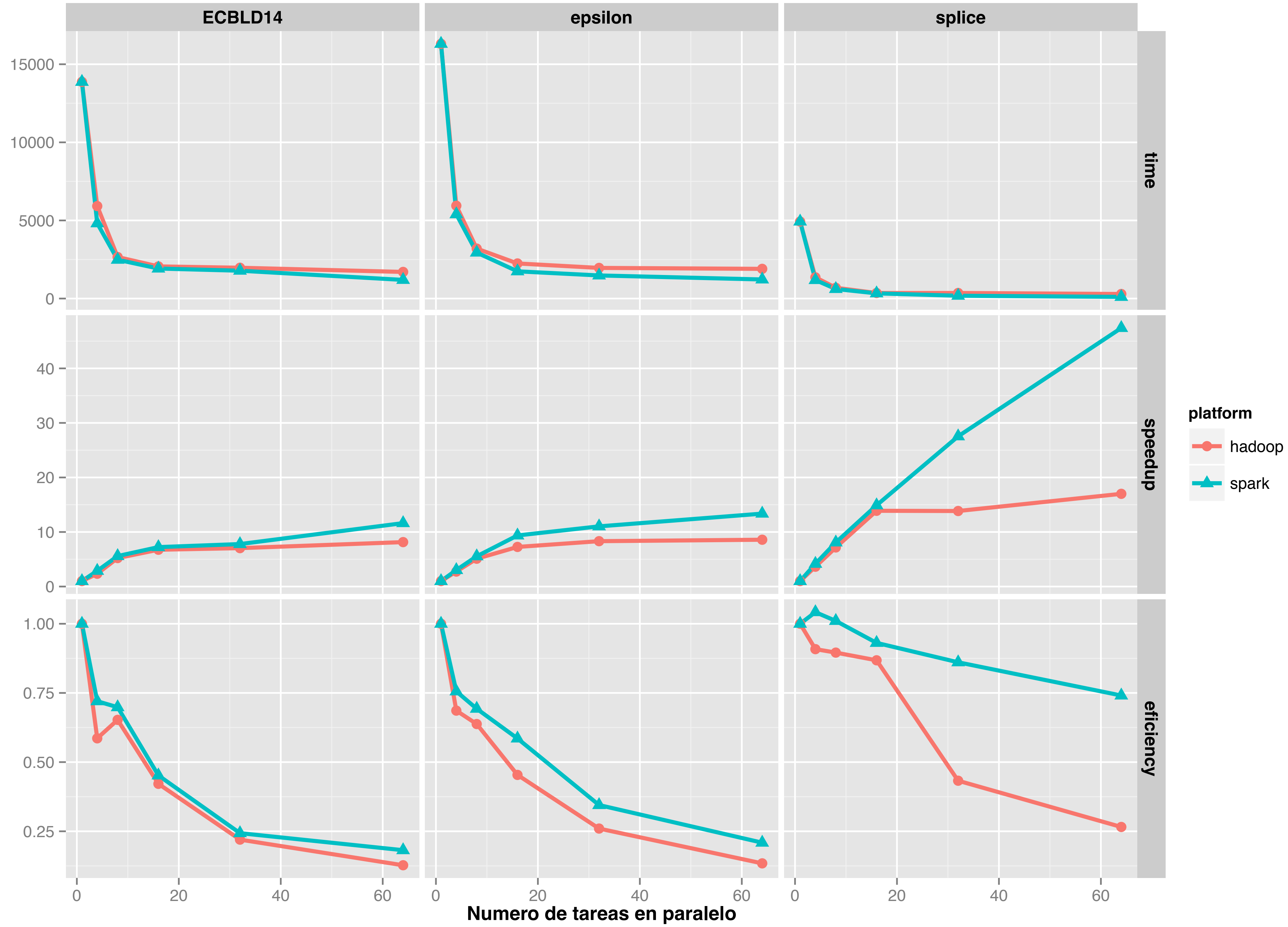
S.I.M.D.
Sistemas Inteligentes y Minería de Datos

AODE UNDER HADOOP ON REAL DATA



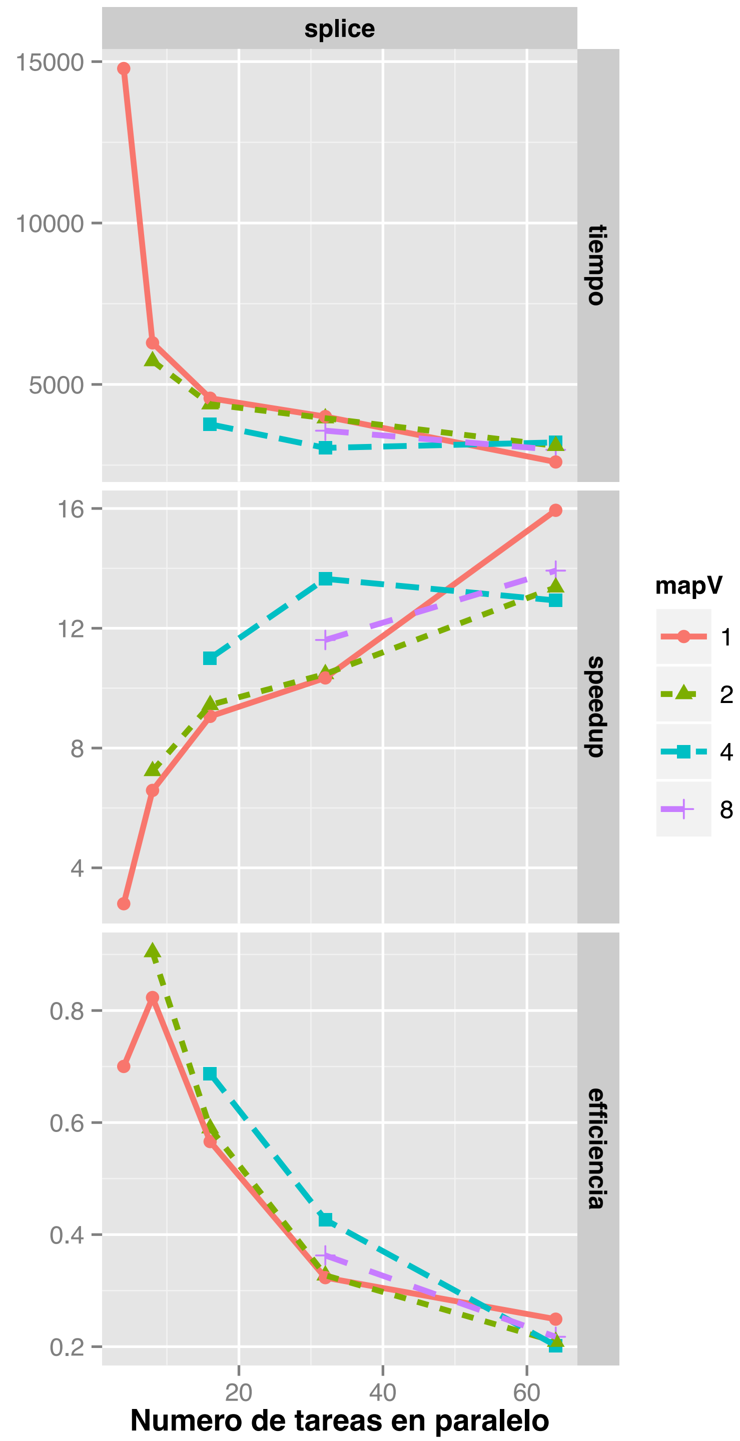
S.I.M.D.
Sistemas Inteligentes y Minería de Datos

AODE UNDER SPARK ON REAL DATA



S.I.M.D.
Sistemas Inteligentes y Minería de Datos

A2DE (FAIL) UNDER HADOOP



Size of the resulting model

	Splice
A1DE	7.7M
A2DE	919.4M

	ECBLD14
A1DE	75.2M
A2DE	70.7G

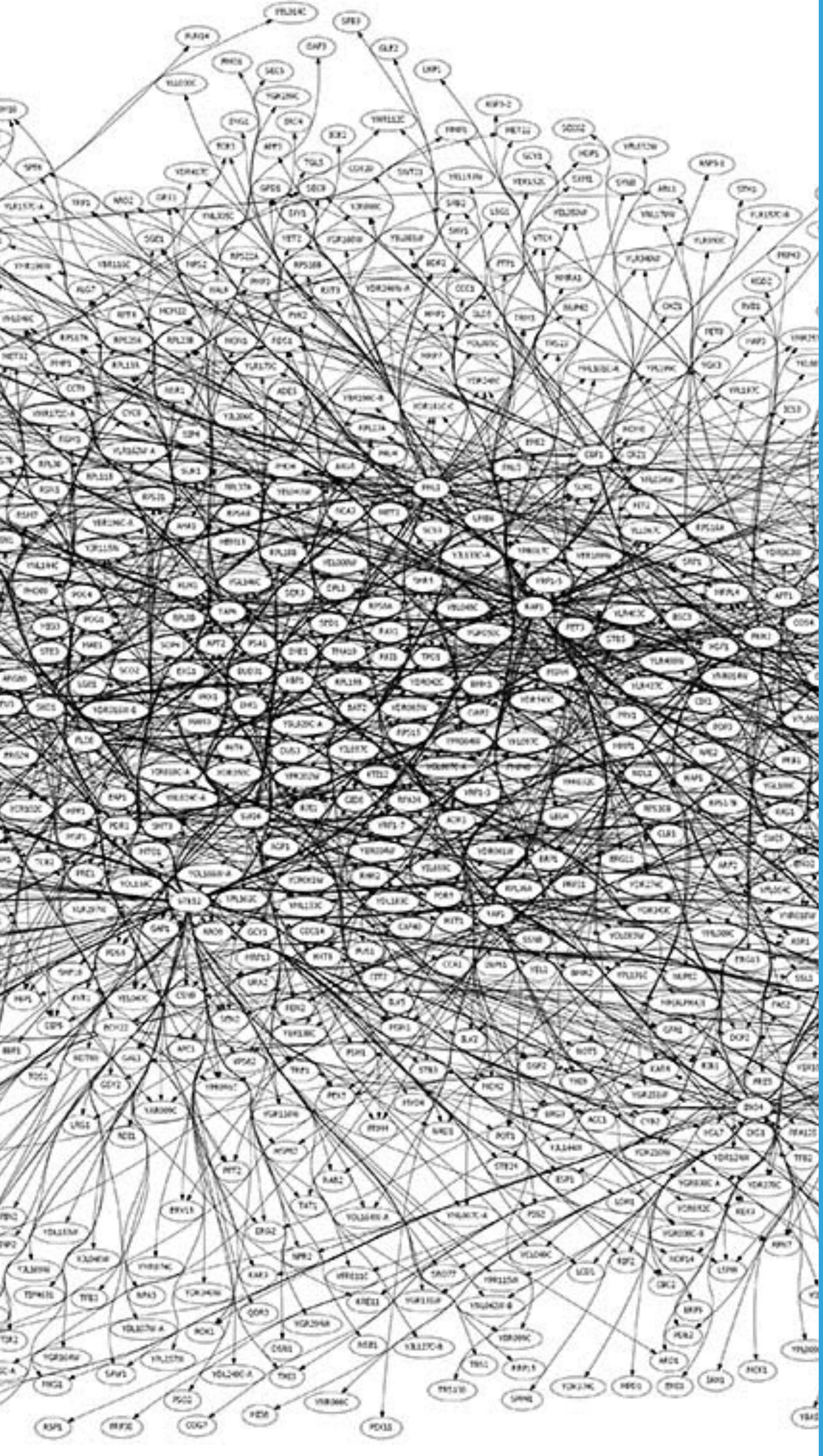
	Epsilon
A1DE	634.8M
A2DE	700G

CONCLUSIONS

- Scalability is obtained as well as elasticity with our design.
- Memory can be managed using vertical partitions, efficiency may be improved by using optimal partitions of the data.
- Higher order attribute combinations should be carefully managed in order to scale up with the data (A2DE).

FUTURE WORK:

- Testing heuristic or exact partitioning for vertical parallel classifiers.
- Design of new classifiers based on A2DE and inspired in random subspaces.



LEARNING GENERAL BAYESIAN NETWORKS FROM LARGE SCALE DATA IN DISTRIBUTED FRAMEWORKS

Next steps in the task (and my thesis). Brief ideas and key concepts to move on from supervised classification.

SAME APPROACH: EVALUATING CANONICAL ALGORITHMS

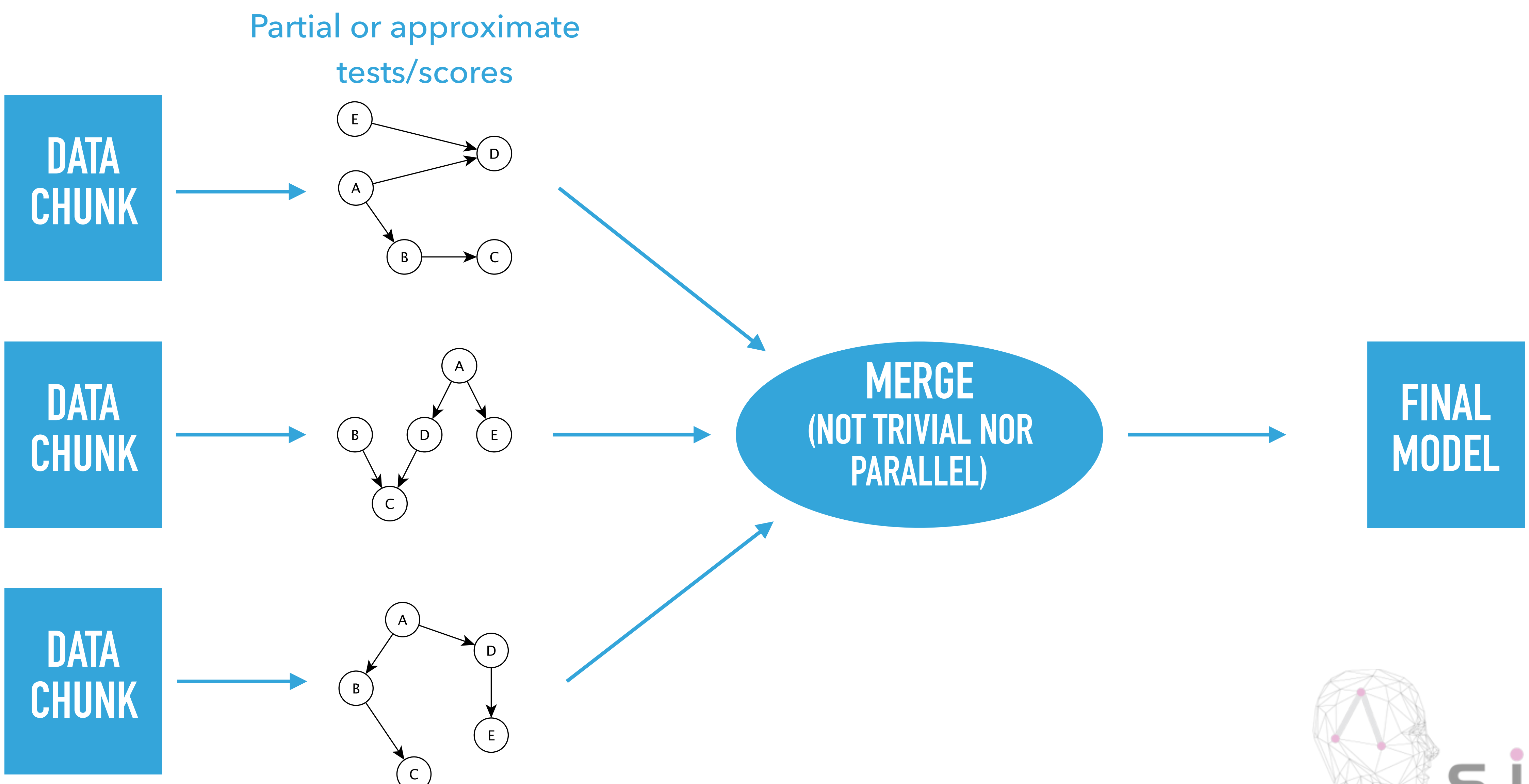
- The PC algorithm is the best candidate. Recent work (Madsen et al. 2015) shows vertical distribution of independence test similar to TAN.

MAIN CONCERNS

- Problems come up with the requirement of iterations (alleviated by Spark). Score+Search approaches difficult to adapt directly.
- Independence test robustness over BigData…
- Availability of big problems to be solved and tested…

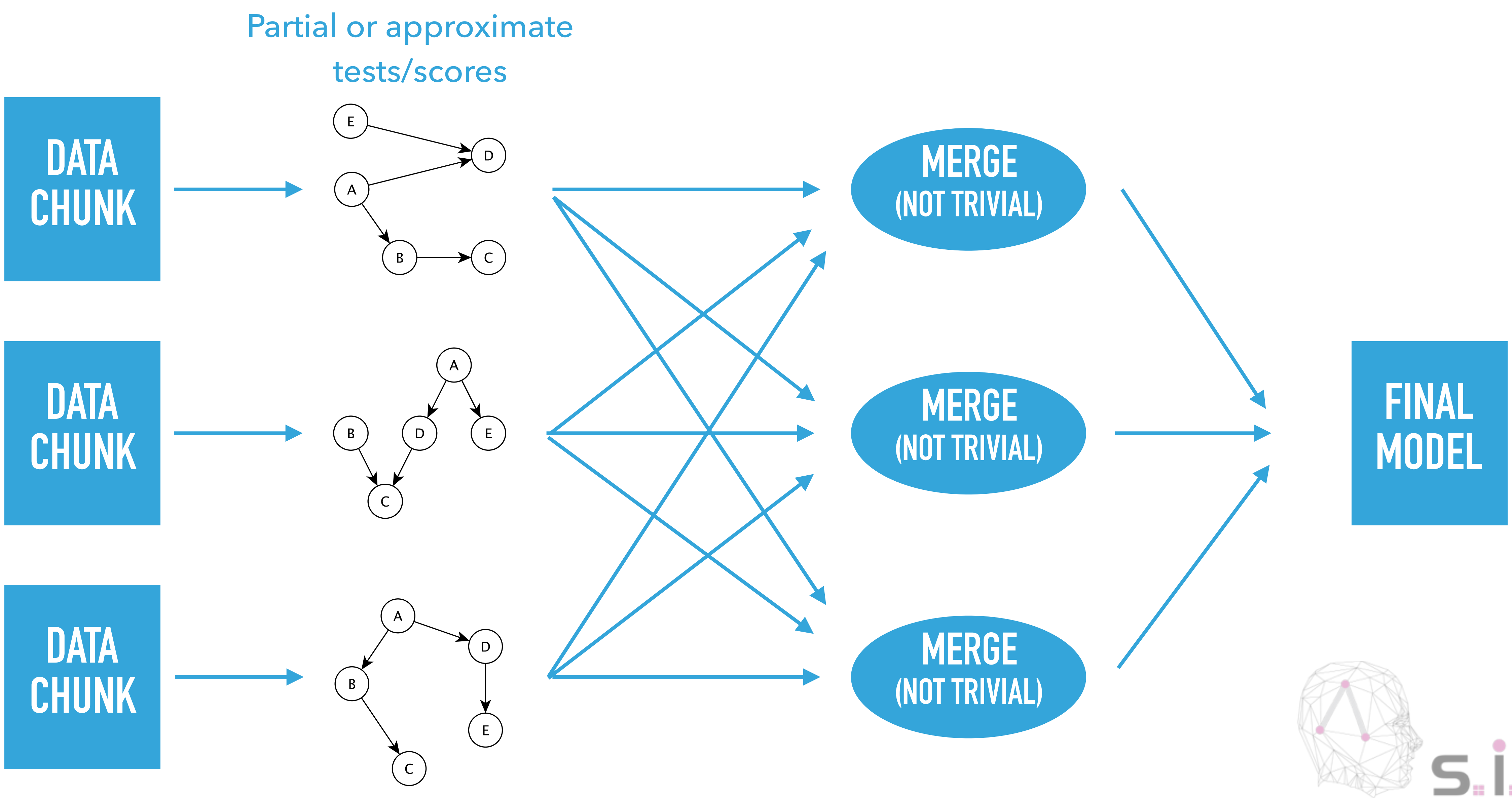
NEW APPROACHES: SPLIT, APPROXIMATE AND MERGE

- A common approach for big data is to split up the data (horizontally) and learn sub-models to be merged (ensembles in classification)



NEW APPROACHES: SPLIT, APPROXIMATE AND MERGE

- A common approach for big data is to split up the data (horizontally) and learn sub-models to be merged (ensembles in classification)



CHALLENGES

- Find an experimental environment, with suitable data and metrics.
- Define proper horizontal partitioners for unsupervised data (random).
- Determine if the statistical tests/scores are valid or have a boundary regarding m.
- Evaluate the ultimate usefulness of such a "big" model, inference schemes, visualization, etc…

JACINTO ARIAS - UCLM

LARGE SCALE BAYESIAN NETWORKS ON HIGHLY DISTRIBUTED COMPUTING FRAMEWORKS

Granada - Febrero 2016