Distributed Data Mining Models as Services on the Grid

Eugenio Cesario
ICAR-CNR, Italy

Domenico Talia
DEIS - University of Calabria, Italy

HPDM 2008 – Pisa, December 15th 2008
10th International Workshop on High Performance Data Mining
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
Distributed Data Mining and the Grid

- Distributed Data Mining (DDM) is a fast growing area that deals with the problem of finding data patterns in scenarios with distributed data and computation.

- Two main reasons:
  - Processing large data requires very high computational cost
  - Geographical distribution of data repositories
Distributed Data Mining and the Grid

- The Grid is a global distributed computing platform through which users gain ubiquitous access to a range of services, computing and data resources
  - Implement distributed high-performance applications
  - Support to the implementation and use of data mining and knowledge discovery systems
    - OGSA (Open Grid Services Architecture)
    - WSRF (Web Service Resource Framework)
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
DDM exploiting the Grid

- Our goal is to design a service-oriented architectural model that can be exploited for different distributed data mining algorithms, deployed as WSRF-compliant Grid services, for the analysis of dispersed data sources.
  - Implementation of mining services by exploiting the Grid infrastructure.

- In order to validate our model, we also present the implementation of two clustering algorithms on such an architecture, and evaluate their performance.
DDM Algorithm

- DDM: execution of data mining processes in a distributed environment
  - At local sites: execution of distinct data mining processes on different distributed data subsets
  - At a central site: combination of the local results at a centralized site

- The whole process of Knowledge Discovery could speeded up
  - Particularly suitable for applications typically dealing with very large amount of data

- Crucial aspect: trade-off between computational and communication cost
DDM Algorithm General Schema

1. Analysis of local datasets at each site

2. Local models (or Sufficient Statistics) sent to a merger site

3. Integration of local models

4. Algorithm termination?
   - Yes: Global Model obtained
   - No: further elaborations for the local sites (go to 1.)
A DDM Service-based Model

- The overall architecture resembles a DDM architectural model
- It is composed of two Grid Services:
  - GlobalMiner-WS, acting as a coordinator on a central site
  - LocalMiner-WS, acting as a miner on local sites
- A resource is associated to each service, used to store the service status (computed models)
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
Implementation

- Two examples of distributed clustering algorithms exploiting the proposed model
  - Distributed K-Means
  - Distributed Expectation Maximization

- Implemented and deployed as WSRF Services by using Globus Toolkit 4.0.x
Distributed K-Means

- **Input parameters:**
  - $K$, number of clusters
  - $S$, seed
  - $\text{maxIterations}$
  - $\text{datasetLocations}$

- **GlobalModel Resource**
  - Centroids $\mu_k$, $k=1,...,K$
  - ClusterMembership
  - CostFunction $\text{Perf}_{KM}$
Distributed K-Means

- Local Model Resource on the $i^{th}$ node is composed by the Sufficient Statistics computed on that node, $SS_k^{(i)} = \{n_k^{(i)}, \Sigma_k^{(i)}, s_k^{(i)}\}$

$$n_k^{(i)} = \left| C_k^{(i)} \right|$$

$$\Sigma_k^{(i)} = \sum_{x \in C_k^{(i)}} x$$

$$s_k^{(i)} = \sum_{x \in C_k^{(i)}} dist(x, \mu_k)^2$$

Number of data points

Linear sum of data points

Square sum of the distance (point, centroid)
Distributed K-Means

1. The coordinator initializes the $K$ centroids, $\{\mu_1, ..., \mu_K\}$ and sends them to each local site

2. The $i^{th}$ local site
   1. assigns each $x$ in $D_i$ to the closest centroid
   2. computes the local Sufficient Statistics
      $$SS_k^{(i)} = \left\{ n_k^{(i)}, \Sigma_k^{(i)}, s_k^{(i)} \right\}, \text{for each cluster } k$$
   3. collects all the $SS_k^{(i)}$ and sends them to the coordinator
Distributed K-Means

3. On the central site, the coordinator

1. adds up all the SS\(^{(i)}\) received from each local site, to get the global sufficient statistics
   \[ SS_k = \left\{ n_k, \sum_k, s_k \right\} \]
   for each cluster k, by the formula
   \[ n_k = \sum_{i=1}^{N} n_k^{(i)}, \quad \sum_k = \sum_{i=1}^{N} \sum_k^{(i)} \]
   \[ s_k = \sum_{i=1}^{N} s_k^{(i)} \]

2. computes the new centroids, \( \{\mu_1, \ldots, \mu_K\} \), and updates the Performance Function
   \[ \mu_k = \frac{\sum_k}{n_k} \]
   \[ Perf_{KM} = \sum_{k=1}^{K} s_k \]

4. If the algorithm converges, stop; else, a new iteration re-starts (go to the step 2)
Distributed EM

- Input parameters:
  - $K$, number of clusters
  - $S$, seed
  - $\text{maxIterations}$
  - $\varepsilon$
  - $\text{datasetLocations}$

- **GlobalModel Resource**
  - Centers $m_k$, $k=1,...,K$
  - Covariance Matrices $\Sigma_k$, $k=1,...,K$
  - Mixing Probabilities $p(m_k)$, $k=1,...,K$
  - CostFunction $\text{Perf}_{EM}$
Distributed EM

1. The coordinator initializes
   - the centers \( m_k \)
   - the covariance matrices \( \Sigma_k \) (for each \( k=1,\ldots,K \))
   - mixing probabilities \( p(m_k) \)
   and sends them to each local site

2. The \( i^{th} \) local site
   1. computes the membership probabilities \( p(m_k|x) \)
   2. computes the local Sufficient Statistics
      \[
      SS^{(i)} = \left\{ s1^{(i)}_k, s2^{(i)}_k, s3^{(i)}_k, f^{(i)} \right\}, \ k=1,\ldots,K
      \]
   3. collects all the \( SS^{(i)} \) and sends them to the coordinator
Distributed EM

3. On the central site, the coordinator
   1. adds up all the $SS^{(i)}$ received from each local site, to get the $m_k$, $\Sigma_k$, $p(m_k)$, $k=1,...,K$ and $\text{Perf}_{EM}$ by the formula:

   $$m_k = \frac{\sum_{i=1}^{N} s2_k^{(i)}}{\sum_{i=1}^{N} s1_k^{(i)}}, \quad \Sigma_k = \frac{\sum_{i=1}^{N} s3_k^{(i)}}{\sum_{i=1}^{N} s1_k^{(i)}}, \quad p(m_k) = \frac{\sum_{i=1}^{N} s1_k^{(i)}}{|D|}, \quad \text{Perf}_{EM} = \sum_{k=1}^{K} f^{(i)}$$

4. If the algorithm converges, stop; else, a new iteration re-starts (go to the step 2)
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
Experimental Evaluation

- Number of nodes: \( n=1,2,4,8 \) (in a LAN)

- Dataset: CoverType (from the UCI archive)
  - 581012 tuples (72 MB)
  - 54 numeric attributes

- Dataset size on each node: \(|D|/n\)
  - We are supposing to have our data just splitted and each partition stored on a given node
Experimental Evaluation – K-Means

- Scalability wrt number of nodes

- Execution time:
  - N=1 -> 3418 s
  - N=8 -> 535 s
Experimental Evaluation – K-Means

- Execution speedup

- Speedup:
  - $N=2 \rightarrow 1.77$
  - $N=8 \rightarrow 6.38$
Experimental Evaluation – EM

- Scalability wrt number of nodes

- Execution time:
  - N=1 -> 5108 s
  - N=8 -> 2283 s
Experimental Evaluation – EM

- Execution speedup

- Speedup
  - N=2 -> 1.94
  - N=8 -> 6.29
Experimental Evaluation

- The distributed version of K-Means and EM build the same model of sequential (centralized) algorithms
  - They do not produce approximated models
    - errorRate_{K-Means} = 0.43
    - errorRate_{EM} = 0.49
Summary

- Distributed Data Mining and the Grid
- DDM exploiting the Grid: A Proposed Architectural Model
- Two case studies: K-Means and EM
- Preliminary Experimental Results
- Concluding Remarks and Future Works
Concluding Remarks

- **DDM and Grid:** Distributed Data Mining models implemented as mining Grid services

- We have defined a general distributed architectural model that can be exploited for distributed algorithms deployed as Grid Services

- **Two implementations:**
  - K-Means
  - Expectation Maximization
Future Works

- More complete experimental evaluation:
  - compute the WSRF overhead vs total execution time
  - total execution time wrt other parameters (#clusters, dimensionality, data set size, etc.)

- Develop and deploy other mining algorithms

- Add a data splitting functionality
Final

- Questions?

Thanks