Unsupervised Data Mining (Clustering)

Javier Béjar

KEMLG

December 2012
Clustering in KDD

- One of the main tasks in the KDD process is the analysis of data when we do not know its structure.
- This task is very different from the task of prediction in which we know the true answer and we try to approximate it.
- A large number of KDD projects involve unsupervised problems (KDNuggets poll, 2-3 most frequent task).
- Problems: Scalability, Arbitrary cluster shapes, New types of data, Parameters fitting, ...
All kinds of data

Data

- Structured Data
- Unstructured Data
- Data Streams
Unstructured data

- Only one table of observations
- Each example represents an instance of the problem
- Each instance is represented by a set of attributes (Discrete, continuous)

| A  | B    | C  | ...
|----|------|----|-----
| 1  | 3.1  | a  | ... |
| 1  | 5.7  | b  | ... |
| 0  | -2.2 | b  | ... |
| 1  | -9.0 | c  | ... |
| 0  | 0.3  | d  | ... |
| 1  | 2.1  | a  | ... |
|    |      |    |     |
Structured data

- One sequential relation among instances (Time, Strings)
  - Several instances with internal structure (e.g., sequences of events)
  - Subsequences of unstructured instances (e.g., sequences of complex transactions)
  - One big instance (e.g., time series)

- Several relations among instances (graphs, trees)
  - Several instances with internal structure (e.g., XML documents)
  - One big instance (e.g., social network)
Data Streams

- Endless sequence of data (e.g., sensor data)
  - Several streams synchronized
  - Unstructured instances
  - Structured instances
- Static/Dynamic model
I am in Hyperspace
The Curse of dimensionality

- Two problems arise from the dimensionality of a dataset
  - The computational cost of processing the data (scalability of the algorithms)
  - The quality of the data (more probability of bad data)
- Two elements define the dimensionality of a dataset
  - The number of examples
  - The number of attributes
- Having too many examples can sometimes be solved by sampling
- Attribute reduction is a more complex problem
Reducing attributes

• Usually the number of attributes of the dataset has an impact on the performance of the algorithms:
  • Because their poor scalability (cost is a function of the number of attributes)
  • Because the inability to cope with irrelevant/noisy/redundant attributes
  • Because the low ratio instances/dimensions (sparsity of space)

• Two main methodologies to reduce attributes from a dataset
  • Transforming the data to a space of less dimensions preserving the structure of the data (dimensionality reduction - feature extraction)
  • Eliminating the attributes that are not relevant for the goal task (feature subset selection)
Many techniques have been developed for this purpose

- Projection to a space that preserve the statistical model of the data (Linear: PCA, ICA; Non linear: Kernel-PCA)
- Projection to a space that preserves distances among the data (Linear: Multidimensional scaling, random projection, SVD, NMF; Non linear: ISOMAP, LLE, t-SNE)

Not all this techniques are scalable per se \(O(N^2)\) to \(O(N^3)\) time complexity), but some can be approximated.
Clustering in Data Mining

Dimensionality Reduction

Original (60 Dim)  PCA

Kernel PCA  ISOMAP
Unsupervised Attribute Selection

- The goal is to eliminate from the dataset all the redundant or irrelevant attributes.
- The original attributes are preserved.
- The methods for Unsupervised Attribute Selection are less developed than in Supervised Attribute Selection.
- The problem is that an attribute can be relevant or not depending on the goal of the discovery process.
- There are mainly two techniques for attribute selection: **Wrapping** and **Filtering**.
All kinds of algorithms

- Density Based
- Grid Based
- Agglomerative
- Model Based
- Other
Model/Center Based Algorithms

- The algorithm is usually biased by a preferred model (hyperspheres, hyper ellipsoids, ...)
- Clustering defined as an optimization problem
- An objective function is optimized (Distorsion, loglikelihood, ...)
- Based on gradient descent-like algorithms
  - k-means and variants
  - Probabilistic mixture models/EM algorithms (e.g., Gaussian Mixture)
- Scalability depends on the number of parameters, size of the dataset and iterations until convergence
Model Based Algorithms
Agglomerative Algorithms

- Based on the hierarchical aggregation of examples
- From spherical to arbitrary shaped clusters
- Defined over the affinity/distance matrix
- Algorithms based on graphs/matrix algebra
- Several criteria for the aggregation
- Scalability is cubic on the number of examples ($O(n^2 \log(n))$ using heaps, $O(n^2)$ in special cases)
Agglomerative Algorithms
Density Based Algorithms

- Based on the discovery of areas of larger density
- Arbitrary shaped clusters
- Noise/outliers tolerant
- Algorithms based on:
  - Topology (DBSCAN-like: core points, reachability)
  - Kernel density estimation (DENCLUE, mean shift)
- Scalability depends on the size of the dataset and the computation of the density (KDE, k-nearest neighbours) (best complexity $O(n \log(n))$)
Density Based Algorithms - DBSCAN
Density Based Algorithms - DENCLUE
Grid Based Algorithms

- Based on the discovery of areas of larger density by space partitioning
- Arbitrary shaped clusters
- Noise/outliers tolerant
- Algorithms based on:
  - Fixed/adaptative grid partitioning of features
  - Histogram mode discovery and dimension merging
- Scalability depends on the number of features, grid granularity and number of examples
Grid Based Algorithms
Other Algorithms

- Based on graph theory
  - K-way graph partitioning
  - Max-flow/min-flow algorithms
  - Spectral clustering
- Message passing/belief propagation (affinity clustering)
- Evolutionary algorithms
- SVM clustering/Maximum margin margin clustering
- ...
Other Clustering Related Areas

- **Consensus clustering**
  - Clustering by consensuating several partitions

- **Semisupervised clustering/clustering with constraints**
  - Clustering with additional information (must and cannot links)

- **Subspace clustering**
  - Clusters with reduced dimensionality
Size matters :-)
Strategies for cluster scalability

- **One-pass**
  - Process data as a stream

- **Summarization/data compression**
  - Compress examples to fit more data in memory

- **Sampling/Batch algorithms**
  - Process a subset of the data and maintain/compute a global model

- **Approximation**
  - Avoid expensive computations by approximate estimation

- **Parallelization/Distribution**
  - Divide the task in several parts and merge models
One pass

- This strategy is based on incremental clustering algorithms
- They are cheap but order of processing affects greatly their quality
- Although can be used as a preprocessing step
- Two steps algorithms
  1. A large number of clusters is generated using the one-pass algorithm
  2. A more accurate algorithm clusters the preprocessed data
Data Compression/Summarization

- Discard sets of examples and summarize by:
  - Sufficient statistics
  - Density approximations
- Discard data irrelevant for the model (do not affect the result)
Approximation

- Not using all the information available to make decisions
  - Using K-neighbours (data structures for computing k-neighbours)
- Preprocessing the data using a cheaper algorithm
  - Generate batches using approximate distances (e.g., canopy clustering)
- Use approximate data structures
  - Use of hashing or approximate counts for distances and frequency computation
Batches/Sampling

- Process only data that fits in memory
- Obtain from the data set:
  - Samples (process only a subset of the dataset)
    - Determine the size of the sample so all the clusters are represented
  - Batches (process all the dataset)
Paralelization/Distribution/Divide&Conquer

- Paralelization of clustering usually depends on the specific algorithm
- Some are difficult to parallelize (eg: hierarchical clustering)
- Some have specific parts that can be solved in parallel or by Divide&Conquer
  - Distance computations in k-means
  - Parameter estimation in EM algorithms
  - Grid density estimations
  - Space partitioning
- Batches and sampling are more general approaches
  - The problem is how to merge all the different partitions
Examples - One pass + Summarization

- One pass + hierarchical single link
  - One pass summarization using Leader algorithm (many clusters)
  - Single link hierarchical clustering of the summaries
  - Theoretical results about equivalence to SL at top levels (from $O(N^2)$ to $O(c^2)$)
One pass + Single Link

1st Phase
Leader Algorithm

2nd Phase
Hierarchical Clustering
Examples - One pass + Summarization

- One pass + model based algorithms
  - BIRCH algorithm
  - One-pass Leader algorithm using an adaptive hierarchical data structure (CF-tree)
  - Postprocess for noise filtering
  - Clustering of cluster centers from CF-tree
One pass + CFTREE (BIRCH)

1st Phase - CFTree

2nd Phase - Kmeans
Examples - Batch + (Divide&Conquer / One-pass)

- Batch + K-means-like
  - SaveSpace algorithm (hierarchy of centers)
  - Clustering using a randomized algorithm based on k-facilities problem (LSEARCH algorithm)
  - Data is divided on $M$ batches
  - Each batch is clustered in $k$ weighted centers
  - Recursively $M \cdot k$ centers are divided on new batches for the next level
  - Last level is clustered in $k$ centers
Batch + Divide & Conquer

- M Batches - K Clusters each
- M*K new examples
Examples - Sampling + Summarization

- Random Samples + Data Compression
  - Scalable K-means
  - Retrieve a sample from the database that fits in memory
  - Cluster with the current model
  - Discard data close to the centers
  - Compress high density areas using sufficient statistics
Sampling + Summarization

Original Data

Sampling

Updating model

Compression

More data is added

Updating model
Examples - Sampling + Distribution

- **CURE**
  - Draws a random sample from the dataset
  - Partitions the sample in \( p \) groups
  - Executes the clustering algorithm on each partition
  - Deletes outliers
  - Runs the clustering algorithm on the union of all groups until it obtains \( k \) groups
Sampling + Distribution

DATA

Sampling + Partition

Clustering partition 1

Clustering partition 2

Join partitions

Labelling data
Examples - Approximation + Distribution

- Canopy Clustering
  - Approximate dense areas by probing the space of examples
  - Batches are formed by extracting spherical volumes (canopies) using a cheap distance
    - Inner radius/Outer radius
  - Each canopy is clustered separately (disjoint clusters)
  - Only distances inside the canopy are computed
  - Several strategies for classical algorithms
Approximation + Distribution

1st Canopy

2nd Canopy

3rd Canopy
Examples - Divide & Conquer

- **K-means + indexing**
  - Build a kd-tree structure for the examples
  - Determine the example-center assignment in parallel
  - Each branch of the kd-tree only keeps a subset of centers
  - If only one candidate in a branch $\rightarrow$ no distances computed
  - Distances only are computed at leaves
  - Effective for low dimensionality
Divide & Conquer - K-means + KD-tree

KD-TREE
Examples - Divide & Conquer

- **Optigrid**
  - Find projections of the data in lower dimensionality
  - Determine the best cuts of the projections
  - Build a grid with the cuts
  - Recurse for dense cells (parallelization)
Divide & Conquer - Grid Clustering
Examples - Approximation

- Quantization Cluster (k-means)
  - Quantize the space of examples and keep dense cells
  - Approximate cluster assignments using cells as examples
  - Recompute centers with examples in the cells
Divide & Conquer + approximation
THANKS

(Questions, Comments, ...)