16th International Conference on Information Quality, 2011

A Hierarchical Clustering Approach to Support the Data Verification Process in Master Data Management

Frank Beer, Kai-Uwe Baryga
SYDECON GmbH
Wolfratshausen, Germany
{frank.beer, kai-uwe.baryga}@sydecon.de

Abstract: Over the past decades, quality on core data has become a major factor on daily business activities. Therefore enterprises are exerted to develop data management strategies in order to ensure a smooth execution of business transactions. As a result, software portals are introduced to either conduct business on maintaining and analyzing master data. In this work we are outlining a clustering approach that is useful for business-drivers in identifying and verifying hidden pattern in that discipline. In addition our method can be applied to bring content-based support to purely business-oriented master data portals.

16th International Conference on Information Quality, 2011

Table of Content

Verification on Maintaining MD Entities

- The quality of **master data** (MD) can rely on many specific criteria (e.g. poor data definitions, processes, expiration). Additionally a major factor is **data acquisition** [2]
- Software **portals** supporting data acquisition (e.g. create, update, block) are covering formal business requirements, but often its pure and ideal business orientation let data-driver struggle because of **poor content-based** or **data-driven support**

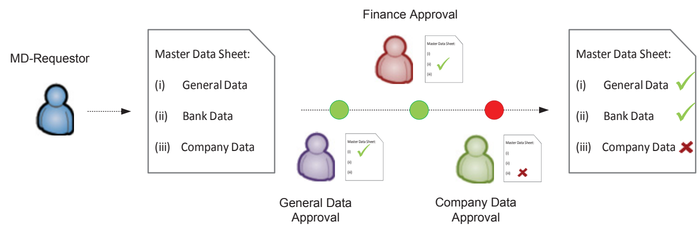


Figure: Typical data aquisition portal system

Verification Process on Cleansing

- Once data is approved and running, **still problems** occure **within daily business** transactions:
  - **Erroneous** delivery of **invoices or purchasing** due to wrongly maintained MD entities
  - **Anomalies** (e.g. duplicated data) confuse on settling invoices
    - Pay a bill multiple times to same vendor
    - Breaking payment terms
  - Order fulfillment process collapses, because material (e.g. spare parts, configuration kits) have **incomplete configuration**
- Typically MD analysis tools provide support to cover these daily affaires based on **query languages**. Often this is not sufficient, because of **limited support** to adaptive or fuzzy considerations:
  - Identifying similarities
  - Find hidden taxonomies

Advanced Techniques for the Verification Process

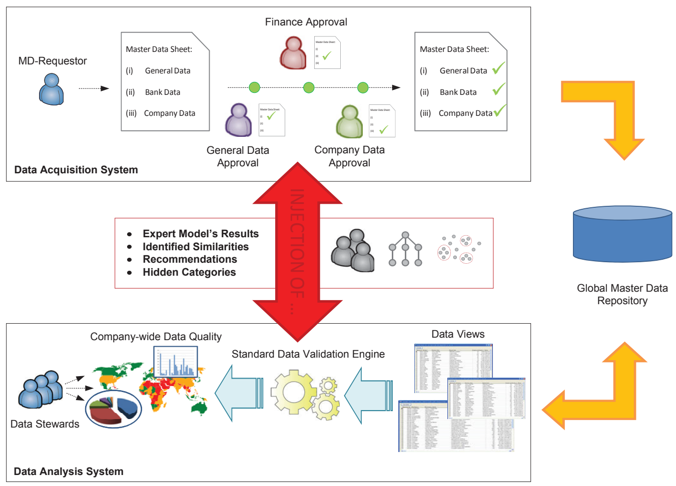


Figure: Advanced data analysis to support the verification process

Pattern Identification to improve Data Quality

- In this work, we introduce a **semi-unsupervised learning approach** supporting business to understand MD in a transparent format:
  - Extracting **groups of similar MD** entities
  - **Identify hidden taxonomies** by the data itself
  - Providing **content-based support** for complex structures
- Unfortunately no perfect model exists tackling all those affairs
- **But** initially a hierachical clustering approach would come close accepting some drawbacks [13]:
  - Computational complexity on huge data sets
  - Cluster identification process does not support distinct groups

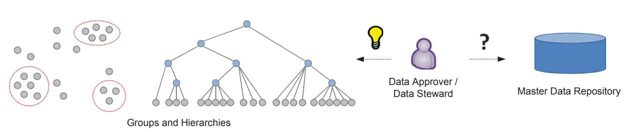


Figure: Pattern recognition and technical limitations

Table of Content

Data Clustering, Properties and its Techniques

- Cluster Analysis is a well-known task in the field of Data Mining. Its goal is to **discover interesting data distributions** represented by **groups, classes or clusters of similar data entities** [9]
- The **characteristic of** such **clusters** should fulfill the following 2 properties [3]:
  - **Cluster member** should **share** some kind of **similarity**
  - **Different clusters** should be **dissimilar** to each other
- A **wide range of clustering algorithms** have been introduced over the past 4 decades. In accordance with [5, 7], most of these approaches can be led back to 2 categories:
  - Relocation or partitioning (e.g. *k*-means, EM-Algorithm)
  - Hierarchical clustering approaches (e.g. CURE, BIRCH)

Hierarchical Clustering

- Ways to identify hierarchies (agglomerative vs. divisive)
- Linkage strategies (i.e. complete, single and average)
- Iterations for Agglomerative Hierarchical Clustering (HAC) and its visualization

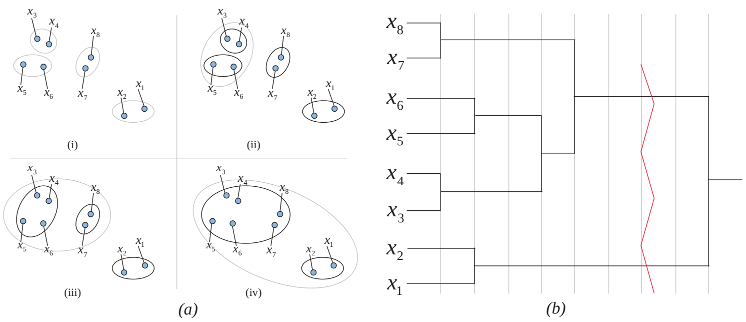
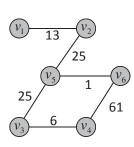


Figure: Agglomerative iterations and its resulting dendrogram

16th International Conference on Information Quality, 2011

Graphs and Information Representation

- A graph can be expressed within a triple $G = \langle V, E, \beta \rangle$:
  - $V = \{v_1, ..., v_n\}, n \in \mathbb{N}$ is the set of vertices
  - $E = \{e_1, ..., e_m\}, m \in \mathbb{N}$ is the set of edges with $\langle x, y \rangle \in E, x, y \in V$
  - $\beta : E \to \mathbb{N}$ is the length mapping
- Data representation through an Information System [11] $\mathcal{A} = \langle \mathbb{U}, A \rangle$:
  - $\mathbb{U} = \{x_1, ..., x_p\}, p \in \mathbb{N}$ is the universe containing all objects
  - $A = \{a_1, ..., a_q\}, q \in \mathbb{N}$ is the attribute set such that $a : \mathbb{U} \to V_a, \forall a \in A$



(a)

–	*color*	*shape*
x_1	*red*	circle
x_2	*blue*	square
x_3	*yellow*	circle

(b)

Figure: A graph and an information system

Table of Content

Challenges in Clustering Master Data

- The organization of MD varies, but typically enterprises build on stable relational database systems in order to store **huge data volumes**
- **Format of data:** Complete MD information is spread through multiple customized relations
- **Capturing** the varity of **mixed data types** (generally vendor and customer MD rely on categorical data but material related data consists of versatile types, e.g. price unit, status, description)
- Finding **adequate methods to measure data (dis)similarity** which is transparent and accepted by data-drivers
- Analysis of hierarchies to identify data clusters

Dissimilarity and Distance of Objects

- Commutative **dissimilarity** operator **mapping** $\alpha : A \rightarrow C$ with $C = \{\neq, \texttt{Levenshtein}, \texttt{SOUNDEX}, \ldots\}$ and $A = \{a_1, a_2, \ldots\}$, the feature set. We write $\neq_b$ to indicate $\alpha(b) = c, b \in A, c \in C$
- **Dissimilarity** of 2 MD entities x, y:

$$d_{xy} = |\{a \mid a(x) \neq_a a(y), \forall a \in A\}| \tag{1}$$

- **Penalty mapping** $\omega : A \rightarrow \mathbb{N}_0$
- **Weighted distance metric**:

$$d_{xy}(\omega) = \sum_{\forall a \in A} \omega(a) \cdot \chi_{xy}(a) = \sum_{\forall a(x) \neq_a a(y)} \omega(a) \tag{2}$$

$$\chi_{xy}(a) = \begin{cases} 1 & , a(x) \neq_a a(y) \\ 0 & , \text{else} \end{cases} \tag{3}$$

16th International Conference on Information Quality, 2011

Tackling the Data Volume in MD Environments (I)

- Creating the distance matrix within a HAC is a tough and crucial job even on modern machines, because ...
  - All object combinations must be determined
  - All object combinations has to be stored in order to find the minimal distance in each iteration
- Considering a graph $G = \langle V, E, \beta \rangle$, the initial size of the distance matrix can be expressed through:

$$|E| = \binom{|V|}{2} = \frac{|V| \cdot (|V| - 1)}{2} \tag{4}$$

- To overcome the complexity, a varity of simplifications exist:
  - Random sampling
  - Build hierarchies based on representatives
  - ...

The MD Clustering Approach

Tackling the Data Volume in MD Environments (II)

- Our approach is based on the ideas in [10, 14], i.e. **growing a minimum-spanning tree** (MST)
- The motivation comes from the fact, that we are able to **reduce the numbers of edges to the size of vertices** within a complete graph
- As a result: **Only nearest neighborhood analysis** can be performed (i.e. single linkage)
- Our MST construction relies on the efficient Prim-Jarnik algorithm [8] which makes use of the MST property as shown below:

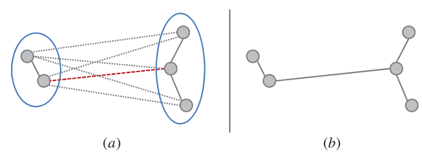


Figure: MST Property - Shortest edge between a partition of vertices

HAC Algorithm - Important Steps

Compute MST $G = \langle V, E, \beta \rangle$
Transform G to $DM \subseteq \mathcal{P}(V) \times \mathcal{P}(V) \times \mathbb{N}_0, \forall \langle A, B, n \rangle : A \neq B$
while $|DM| > 0$ **do**
    $g \leftarrow \langle A, B, n \rangle \in DM :=$ Find min distance in DM
    $C \leftarrow A \cup B$ `// build new cluster` C
    $DM \leftarrow DM - \{g\}$ `// remove` g `from` DM
    foreach $h \leftarrow \langle X, Y, m \rangle \in DM$ **do**
        if $X - A = \emptyset \vee X - B = \emptyset$ **then** $h \leftarrow \langle C, Y, m \rangle$
        else if $Y - A = \emptyset \vee Y - B = \emptyset$ **then** $h \leftarrow \langle X, C, m \rangle$
    end
    `// store all relevant info into cluster protocol`
end
`// return cluster protocol`

Figure: Pseudocode to construct hierarchical clusters

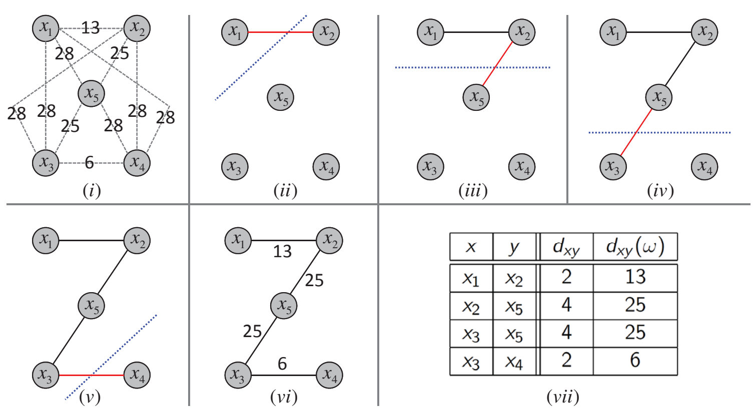
Bringing it all together - An illustration (I)

-	a_1	a_2	a_3	a_4	a_5
$\omega(a)$	10	3	5	3	7
$\neq_a$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$
x_1	zz Software	98632	Long Island	Ocean Drive	US
x_2	Novel Food	98632	Long Island	Laurel Road	US
x_3	ABC AM	63073	Berlin	Flottenstr.	DE
x_4	ABC AM	-	Berlin	-	DE
x_5	Cityprint	63073	London	Laurel Road	GB

x	y	d_{xy}	$d_{xy}(\omega)$
x_1	x_2	2	13
x_3	x_4	2	6
x_3	x_5	4	25

Figure: Input data set and important object distances

Bringing it all together - An illustration (II)



x	y	d_{xy}	$d_{xy}(\omega)$
x_1	x_2	2	13
x_2	x_5	4	25
x_3	x_5	4	25
x_3	x_4	2	6

Figure: Construction of the MST and the distance matrix

Bringing it all together - An illustration (III)

- After clustering the objects with minimal distance in each iteration, the resulting dendrogram can be reviewed as follows:

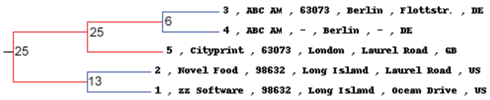


Figure: Graphical result of the hierarchies

- For bigger data sets, the protocol is stored in a relation:

	it	ClusterID	ChildCluster1	ChildCluster2	Cluster1IsInit	Cluster2IsInit	Distance
1	0	1	-1	-1	0	0	-1
2	0	2	-1	-1	0	0	-1
3	0	3	-1	-1	0	0	-1
4	0	4	-1	-1	0	0	-1
5	0	5	-1	-1	0	0	-1
6	1	761b38b2-76b0-45fe-855f-791a9479ea48	3	4	1	1	6
7	2	34c158ee-778a-4f60-8b89-eb4aa091cada	2	1	1	1	13
8	3	125e6032-2f7f-4191-8115-c585752bc6f1	761b38b2-76b0-45fe-855f-791a9479ea48	5	0	1	25
9	4	6700a44c-b157-4e5a-abe7-c3529adc1cc2	125e6032-2f7f-4191-8115-c585752bc6f1	34c158ee-778a-4f60-8b89-eb4aa091cada	0	0	25

Figure: Relational representation of the cluster protocol

Table of Content

Clustering based on single Word Similarity

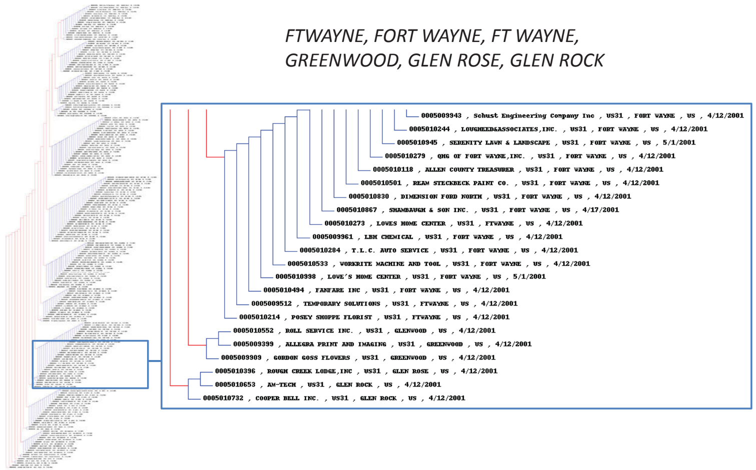


Figure: Address grouping by fuzzy city comparison

Experimental Results

16th International Conference on Information Quality, 2011

Clustering based on multiple Attributes

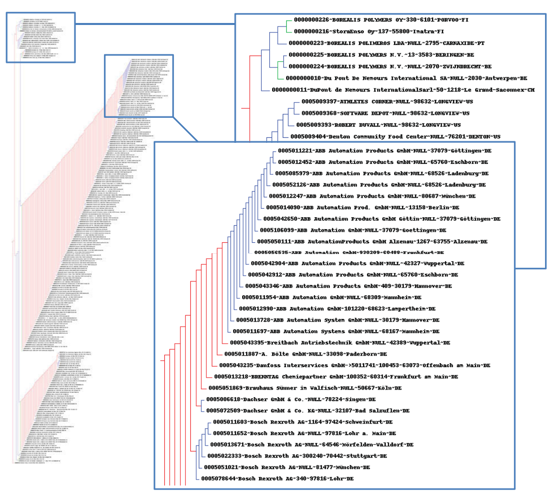


Figure: Vendor clustering by name and country

Experimental Results

Empirical Time Analysis

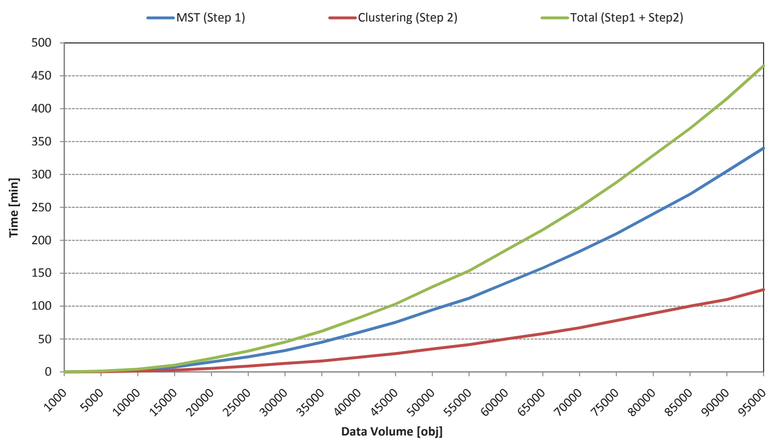


Figure: Time consumption on 6 attributes

Table of Content

Conclusion and Future Work (I)

- Presentation of **adaptive agglomerative clustering approach** which:
  - Is applicable in the field of Master Data Management
  - Comes without preprocessing techniques, e.g. discretization
  - Produces valueable and previously unknown pattern, i.e. knowledge
- Our model makes use of a **MST construction** based on the ideas in [10, 14] in order to reduce the massive amount of considerable object distances
- As a result we only can make use of nearest neighborhood considerations, what often is **not competitive** in comparison to complete linkages (review discussed chaining effect in [9])

Conclusion and Future Work (II)

- Increase performance by utilizing **parallel computation** as proposed by the authors in [1]
- **Experiencing further techniques** from Information Theory and further statistical methodology to handle other linkage strategies in a scalable way
- **Annotate cluster model** for complete automated deployment
  - Transforms our master data taxonomies into a conceptual clustering such as [4, 6]
  - Introduction of insert and update operations to the clustering tree decreasing running time
- Extend approach to **bag-oriented object similarity** measures that analyze relational data in a native fashion [12]

Table of Content

[1] Micah Adler, Wolfgang Dittrich, Ben Juurlink, Miroslaw Kutylowski, and Ingo Rieping. Communication-optimal parallel minimum spanning tree algorithms. In *Proceedings of the 10th annual ACM symposium on Parallel algorithms and architectures*, pages 27–36. ACM, 1998.

[2] Detlef Apel, Wolfgang Behme, Rüdiger Eberlein, and Christian Merighi. *Datenqualität erfolgreich steuern. Praxislösungen für Business-Intelligence-Projekte.* Carl Hanser, 2009.

[3] Johann Bacher, Andreas Pöge, and Knut Wenzig. *Clusteranalyse: Anwendungsorientierte Einführung.* Oldenbourg, 1996.

[4] Gautam Biswas, Jerry B. Weinberg, and Douglas Fisher. Iterate: A conceptual clustering algorithm for data mining. *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS*, 28(2):100–111, 1998.

[5] Daniel Fasulo. An analysis of recent work on clustering algorithms. Technical report, University of Washington.

[6] Douglas Fisher. Improving inference through conceptual clustering. In *AAAI-87 Proceedings*, pages 461–465. The AAAI Press, 1987.

[7] Chris Fraley and Adrian E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[8] Michael T. Goodrich and Roberto Tamassia. *Data Structures and Algorithms in Java*. Wiley, 1998.

[9] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: An efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 73–84. ACM, 1998.

[10] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26:354–359, 1983.

[11] Z. Pawlak. *Rough Sets - Theoretical Aspects of Reasoning about Data*. Kluwer Academic Phublishers, 1991.

[12] Tae-Wan Ryu and Christoph F. Eick. A database clustering methodology and tool. *Inf. Sci. Inf. Comput. Sci.*, 171:29–59, 2005.

[13] Tony Segaran. *Programming Collective Intelligence*. O'Reilly, 2007.

[14] Robert E. Tarjan. *Data Structures and Network Algorithms*. Society for Industrial Mathematics, 1983.

References

Track: IQ Measurement