DATA CLUSTERING: FROM DOCUMENTS TO THE WEB

Dušan Húsek, Jaroslav Pokorný, Hana Řezanková, Václav Snášel

Institute of Computer Science, Academy of the Sciences of the Czech Republic,
Pod vodárenskou věží 2, 182 07 Praha 8, Czech Republic
dusan@cs.cas.cz

Phone: (+420) 603 444 471

Fax: (+420) 28658 5789

Department of Software Engineering, Charles University,
Malostranské náměstí 25, 118 00 Praha 1, Czech Republic
pokorny@ksi.ms.mff.cuni.cz

Phone: (+420) 221 914265

Fax: (+420) 221 914323

Department of Statistics and Probability, University of Economics, Prague,
nám. W. Churchilla 4, 130 67 Praha 3, Czech Republic
rezanka@vse.cz

Phone (+420) 224 095 483

Fax: (+420) 224 095 431

Department of Computer Science, Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
vaclav.snasel@vsb.cz
Phone: (+420) 603 444 471

Fax: (+420) 28658 5789

DATA CLUSTERING: FROM DOCUMENTS TO THE WEB

Abstract

The chapter provides a survey of some clustering methods relevant to the clustering document collections and, in consequence, Web data. We start with classical methods of cluster analysis which seem to be relevant in approaching to cluster Web data. The graph clustering is also described since its methods contribute significantly to clustering Web data. A use of artificial neural networks for clustering has the same motivation. Based on previously presented material, the core of the chapter provides an overview of approaches to clustering in the Web environment. Particularly, we focus on clustering web search results, in which clustering search engines arrange the search results into groups around a common theme. We conclude with some general considerations concerning the justification of so many clustering algorithms and their application in the Web environment.

INTRODUCTION

Document and information retrieval (IR) is an important task for Web communities. In this chapter we introduce some clustering methods with aim to its use for clustering, classification, and retrieval of Web documents.

The aim of clustering is either to create groups of similar objects or create a hierarchy of such groups [53]. Clustering is often confused with classification, but there is some difference between the two techniques. In classification the objects are assigned to predefined classes, whereas in clustering the classes are also to be defined. We focus here mainly on document clustering, e.g. objects are texts, web pages, phrases, etc. Any clustering technique relies on four concepts:

- model of data to be clustered,
- similarity measure,
- cluster model, and
- clustering algorithm that builds the clusters using the data model and the similarity measure.

By a data model we mean the common notion used in IR. For example, in the Boolean model the text is represented by a set of significant terms, in the vector space model documents are modelled by vectors of term weights. A way, how objects are clustered is called a cluster model. This approach is in accordance with [53] (Jain, 1988), where objects are called patterns and the following steps are considered:

- pattern representation (optionally including feature extraction and/or selection),
- definition of a pattern proximity measure appropriate to the data domain,
- clustering or grouping,
- data abstraction (if needed), and
- assessment of output (if needed).

The last two steps concern rather application of clustering. Data abstraction influences a description of clusters, for example labels of folders in clustering with snippets in the Web environment. A difficult task is an assessment of output, i.e. an evaluating the quality of

clustering. Various statistical approaches are used in this context, while in IR we make this with the usual measures such as precision and recall. In the past, clustering has been mainly addressed to exploratory data analysis. In consequence, most of data clustering methods come from statistics. The other application area is fast retrieval of the relevant information from databases, particularly from huge text collections. In this chapter we will present clustering from this perspective. As texts become more and more multimedia oriented, a lot of special clustering techniques can be applied in this context (e.g. image clustering). Consider now the Web or a set of Web search results as a text collection. Web pages are modelled from various points of view. In a Web model we can combine

- textual information,
- hyperlink structure,
- co-citation,
- metadata,
- pictures, and
- HTML or XML structure of Web pages.

We can observe that e.g. hyperlink structure or a combining data and metadata in XML documents extend usual assumptions about texts to be clustered. Consequently, new issues appear.

As different communities use clustering, the associated terminology varies widely. We will freely take up the taxonomy presented in [54] (Jain, 1999).

- *Hierarchical vs. flat*. In the former case, a hierarchy of clusters is found and objects can be assigned to different numbers of clusters. The result of flat clustering is an assignment of objects to the certain number of clusters determined before analysis. These methods are sometimes divided into partitioning methods if classes are mutually exclusive and clumping methods, in which an overlap is allowed.
- *Agglomerative vs. divisive (hierarchical clustering)*. Agglomerative methods start with each object in a separation group, and proceed until all objects are in a single group. Divisive methods start with all objects in a single group and proceed until each object is in a separate group.
- *Monothetic vs. polythetic (hierarchical clustering)*. Monothetic methods use single-feature based assignment into clusters. Polythetic algorithms consider multiple-features based assignment.
- *Hard vs. fuzzy*. In non-fuzzy or hard clustering, objects are divided into crisp clusters, where each object belongs to exactly one cluster. In fuzzy clustering, the object can belong to more than one cluster, and associated with each of the objects are membership grades which indicate the degree to which the objects belong to the different clusters.
- *Deterministic vs. stochastic*. Deterministic clustering methods, given a data set, always arrive at the same clustering. Stochastic clustering methods employ stochastic elements (e.g. random numbers) to find a good clustering.
- *Incremental vs. non-incremental*. Non-incremental clustering methods mainly rely on having the whole data set ready before applying the algorithm. For example, a hierarchical agglomerative clustering belongs to this class. Incremental clustering algorithms work by assigning objects to their respective clusters as they arrive.

Beside of flat and hierarchical methods, some authors (e.g. [40,73]) (Han, 2001; Mercer, 2003). distinguish from three to four further categories. These are density-based approaches,

grid-based approaches, model-based approaches, and also hybrid approaches, which are based on the all three mentioned approaches.

The chapter provides a survey of some clustering methods relevant to the clustering document collections. In Section 2 we start with classical methods of cluster analysis. In general, the choice of methods has been influenced by a progress appearing recently in approaching to cluster Web data. Graph clustering described in Section 3 contributes to this issue. Section 4 about artificial neural networks is built with the same motivation. Based on previously presented material, Section 5 tries to provide an overview of approaches to clustering in the Web environment. Particularly, we focus on clustering web search results, in which clustering search engines arrange the search results into groups around a common theme. Section 6 concludes the chapter.

METHODS OF CLUSTER ANALYSIS

The following terms and notation will be used throughout this chapter.

- An *object* (or pattern or feature vector or Web page) $\mathbf{x}$ is a single data item used by the clustering algorithm. It typically consists of a vector of p components $\mathbf{x} = (x_1,\ldots,x_p)$.
- The individual scalar components x_i of an object $\mathbf{x}$ are called *features* (or attributes or values of variables).
- p is the dimensionality of the objects or of the feature space.
- An object set will be denoted $\mathrm{X} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$. The i^{th} object in X will be denoted $\mathbf{x}_i = (x_{i1},\ldots,x_{ip})$. In many cases an object set to be clustered is viewed as an $n \times p$ object matrix.

Clustering is a division of the objects set into subsets (groups) of similar objects. Each group, called a *cluster*, consists of objects that are similar between themselves and dissimilar to objects of other groups.

Clustering can be realized by means of such techniques as multivariate statistical methods, neural networks, genetic algorithms, and formal concept analysis. In the terminology of machine learning, we can talk about *unsupervised learning*. Statistical methods for clustering can be classified to the groups like cluster analysis, multidimensional scaling ([53,32]) (Jain, 1988; Gordon, 1999), factor analysis and correspondence analysis.

The following tasks should be solved in connection with clustering of documents: clustering of large data sets, clustering in high dimensional spaces, a sparse matrix approach, outlier detection and handling.

We can start with methods for the *vector space model* (VSM [83] (Salton, 1988), which represents a document as a vector of the terms that appear in all the document set. Each feature vector contains term weights of the terms appearing in that document. The term weighting scheme is usually based on $tf \times idf$ method in IR.

A collection of documents can be represented by a term-document matrix. A *similarity* between documents is measured using one of several similarity measures that are based on relations of feature vectors, e.g. cosine of feature vectors or, equivalently, by a distance measure (generally, we will use the term of *proximity measure*). We can consider both documents (Web pages) and terms (topics) as objects of clustering. In the latter case, searching of clusters is very close to reduction of dimensionality. For example, factor analysis

can be used both for reduction of dimensionality and for clustering [42] (Hartigan, 1975).

We can mention the following basic requirements for clustering techniques for large data sets [40] (Han, 2001) scalability (clustering techniques must be scalable, both in terms of computing time and memory requirements), independence of the order of input (i.e. order of objects which enter into analysis) and ability to evaluate the validity produced clusters. The user usually wants to have a robust clustering technique which is robust on the following areas: dimensionality (the distance between two objects must be distinguishable in a high dimensional space), noise and outliers (an algorithm must be able to detect noise and outliers and eliminate their negative effects), statistical distribution, cluster shape, cluster size, cluster density, cluster separation (an algorithm must be able to detect overlapping clusters).

The particular attention is paid to the problem of high dimensional data. Clustering algorithms based on proximity measures work effectively for dimensions below 16. Therefore, Berkhin [6] (Berkhin, 2002)claims that data with more than 16 attributes is high dimensional. Two general techniques are used in the case of high dimensionality: *attributes transformation* and *domain decomposition*.

In the former case, for certain type of data aggregated attributes can be used. If it is impossible, *principal component analysis* can be applied. However, this approach is problematic since it leads to a cluster with poor interpretability. In the area of IR, *singular value decomposition* (SVD) technique is used to reduce dimensionality. As concerns domain decomposition, it divides the data into subsets (canopies) using some inexpensive similarity measure. The dimension stays the same, but the costs are reduced. Some algorithms were designed for *subspace clustering*, for example CLIQUE or MAFIA.

For large data sets, hybrid methods, which combine different techniques, are often suggested. In past ten years, new approaches to clustering large data sets were suggested and some surveys of clustering methods were prepared, for example [6,54,73,80]. (Berkhin, 2002; Jain, 1999; Mercer, 2003; Řezanková, 2004).

Several approaches are used for clustering large data sets by means of traditional methods of cluster analysis. One of them can be characterized by the following way. Only objects of the sample (either random or representative) are clustered to the desired number of clusters. Other objects are assigned to these created clusters. In the second approach, the data set is divided to blocks (their size is determined by capability of used software product) and in each block objects are clustered. As results we obtain *centroids* which characterize created clusters (centroid is a vector of average values of object features computed on the base of objects assigned to the cluster). At the final stage, the centroids are clustered for obtaining desired number of clusters. The centroids can be obtained also by other way, for example by incremental clustering.

For easier searching of document clusters, we can find groups of similar terms (topics). We can repeat clustering of terms and documents for achievement interesting co-occurrences. We can find second order co-occurrences of documents.

In the following text we will focus only on clustering of documents (Web pages) and subspace clustering. When clusters of documents (Web pages) are found, each cluster can be characterized by the certain way, e.g. by centroid or *medoid* (an object of the cluster which was chosen as representative). In the process of IR, we can calculate similarity coefficients between the query and the centroid or medoid and search which clusters of documents best correspond to the query. This way of calculation is less time consuming for searching documents with high similarity than calculation of similarity coefficients between the query and individual documents.

Dissimilarity and similarity measures

A *dissimilarity* (or *distance*) between object $\mathbf{x}$ and $\mathbf{y}$ (or distance measure) is function $d(\mathbf{x},\mathbf{y}) : \mathrm{X} \times \mathrm{X} \to \mathrm{R}$ which satisfied the following conditions:

$$d(\mathbf{x},\mathbf{x}) = 0$$
$$d(\mathbf{x},\mathbf{y}) \geq 0$$
$$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$$

For distance we require triangle inequality to satisfy, i.e. for any objects $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$

$$d(\mathbf{x},\mathbf{z}) = d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z})$$

A *similarity* $s(\mathbf{x},\mathbf{y})$ between object $\mathbf{x}$ and $\mathbf{y}$ is function $s(\mathbf{x},\mathbf{y}) : \mathrm{X} \times \mathrm{X} \to \mathrm{R}$ which satisfied the following conditions:

$$s(\mathbf{x},\mathbf{x}) = 1$$
$$s(\mathbf{x},\mathbf{y}) \geq 0$$
$$s(\mathbf{x},\mathbf{y}) = s(\mathbf{y},\mathbf{x})$$

Both dissimilarity and similarity functions is often defined by a matrix.

Some clustering algorithms operate on a dissimilarity matrix (they are called distance-space methods in [73] (Mercer, 2003)). How the dissimilarity between two objects is computed depends on the type of the original objects.

Here are some most frequently used dissimilarity measures for continuous data.

- *Minkowski* L_q distance (for $1 \geq q$)

$$d(\mathbf{x}_i,\mathbf{x}_j) = \sqrt[q]{\sum_{l=1}^{p} |\, x_{il} - x_{jl} \,|^q}$$

- *City*-block (or Manhattan distance or L_1)

$$d(\mathbf{x}_i,\mathbf{x}_j) = \sum_{l=1}^{p} |\, x_{il} - x_{jl} \,|$$

- *Euclidean distance* (aliases L_2)

$$d(\mathbf{x}_i,\mathbf{x}_j) = \sqrt{\sum_{l=1}^{p} (x_{il} - x_{jl})^2}$$

- *Chebychev distance metric* (or maximum or L_∞)

$$d(\mathbf{x}_i,\mathbf{x}_j) = \max_{l=1,\ldots,p} (|\, x_{il} - x_{jl} \,|)$$

In the case of Chebychev distance the objects with the largest dispersion will have the largest

impact on the clustering. If all objects are considered equally important, the data need to be standardized first. If continuous measurements are on an unknown scale (*continuous ordinal variables*), each value x_{ip} must be replaced by its rank $r_{ip} \in \{1,\ldots,M_i\}$ and the rank scale must be transformed to $[0,1]$. Then dissimilarities as for interval-scaled variables can be used.

A relation between two objects can be expressed also as a similarity [7] (Berry, 1999). It can be measured as a correlation between feature vectors. For interval-scaled data, Pearson correlation coefficient is used; for ordinal data, Goodman-Kruskal gamma correlation coefficient is used. Further possibility is a *cosine measure*. Cosine of feature vectors is calculated according the following formula:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{p} x_{il} x_{jl}}{\sqrt{\sum_{l=1}^{p} x_{il}^2} \sqrt{\sum_{l=1}^{p} x_{jl}^2}}$$

Further, we can use *Jaccard coefficient* or *Dice coefficient*. The former can be expressed as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{l=1}^{p} x_{il} x_{jl}}{\sum_{l=1}^{p} x_{il}^2 + \sum_{l=1}^{p} x_{jl}^2 - \sum_{l=1}^{p} x_{il} x_{jl}}$$

and the latter as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{2 \times \sum_{l=1}^{p} x_{il} x_{jl}}{\sum_{l=1}^{p} x_{il}^2 + \sum_{l=1}^{p} x_{jl}^2}$$

As concerning as *binary* variables, we distinguish symmetric ones (both categories are equally important – e.g. male and female) and asymmetric ones (one category carries more importance than the other). For document clustering, the latter has to be considered. Let us consider the following contingency table:

$\mathbf{x}_i / \mathbf{x}_j$	1	0
1	a	b
0	c	d

with frequencies a, b, c and d. For asymmetric variables, we can use for example Jaccard coefficient or Dice coefficient. The former can be expressed as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{a}{a+b+c}$$

and the latter as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{2a}{2a+b+c}$$

We can also use cosine of feature vectors, i.e. Ochiai coefficient

$$s(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\frac{a}{a+b} \times \frac{a}{a+c}}$$

If data set is a contingency table with frequencies of categories, we can use dissimilarity measures based on the *chi*-square test of equality for two sets of frequencies:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{p} \frac{(x_{il} - E(x_{il}))^2}{E(x_{il})} + \sum_{l=1}^{p} \frac{(x_{jl} - E(x_{jl}))^2}{E(x_{jl})}}$$

where $E(x_{il})$ and $E(x_{jl})$ are expected values on the assumption of independency in the contingency table

$$E(x_{il}) = \frac{\sum_{m=1}^{p} (x_{im}) \times (x_{il} + x_{jl})}{\sum_{m=1}^{p} x_{im} + \sum_{m=1}^{p} x_{jm}}$$

We can also use *phi*-square between sets of frequencies: *chi*-square statistic is divided by the total number of cases and square root of this value is computed.

There are a lot of measures for clustering. We will mention a way how a distance between clusters can be measured. *Log-likelihood* distance between clusters a and b is

$$d(a,b) = \zeta_a + \zeta_b - \zeta_{<a,b>}$$

where $< a,b >$ denotes a cluster created by joining objects from clusters a and b, and

$$\zeta_v = -n_v \sum_{l=1}^{p} \frac{1}{2} \log(s_l^2 + s_{vl}^2)$$

where n_v is the number of objects in the v^{th} cluster, p is the number of variables, s_l^2 is a sample variance of the l^{th} continuous variable, and s_{vl}^2 is a sample variance of the l^{th} continuous variable in the v^{th} cluster. This measure can be also used for investigation a distance between objects.

Partitioning Algorithms

These methods divide the data set into k clusters, where the integer k needs to be specified by the user. An initial classification is modified by moving objects from one group to another

if this will reduce the sum of squares. Algorithm *k*-means is very often described in the literature. For large data sets some algorithms are based on the PAM (Partitioning Around Medoids) algorithm. Algorithms *k*-means and *k*-medoids belong to methods of hard clustering. However, we have to consider also the possibility of overlapping clusters. One approach how to solve this task is fuzzy clustering.

Partitioning Around Medoids. The algorithm proceeds at two steps. First, for a given cluster assignment *centrally located objects* (medoids) are selected by minimizing total distance to other objects in the cluster. At the second step, each object is assigned to the closest medoids. Object $\mathbf{x}_i$ is put into cluster v when medoid m_v is nearer than any other medoid m_w, i.e.

$$d(\mathbf{x}_i, m_v) \leq d(\mathbf{x}_i, m_w) \text{ for all } w = 1, 2, \ldots, k.$$

These two steps are repeated until assignments do not change.

The PAM algorithm was extended to the CLARA (Clustering LARge Applications) method [58] (Kaufman, 1990). CLARA clusters a sample from the data set and then it assigns all objects in the data set to these clusters. The process is repeated several times and then the clustering with the smallest average distance is selected.

The improvement of CLARA algorithm is CLARANS (Clustering Large Applications based on a RANdomized Search) [77] (Ng, 1994). It proceeds by searching a random subset of the neighbours of a particular solution. Thus the search for the best representation is not confined to a local area of the data.

Fuzzy Cluster Analysis. The aim of these methods is to compute memberships u_{iv} for each object $\mathbf{x}_i$ and each cluster v. Memberships have to satisfy the following conditions [32,52] (Gordon, 1999; Höppner, 2000):

$$u_{iv} \geq 0 \text{ for all } i = 1, \ldots, n \text{ and all } v = 1, \ldots, k,$$
$$\sum_{v=1}^{k} u_{iv} = 1 \text{ for all } i = 1, \ldots, n$$

The memberships are defined through minimization of function f:

$$f = \sum_{v=1}^{k} \frac{\sum_{i,j=1}^{n} u_{iv}^2 u_{jv}^2 d(\mathbf{x}_i, \mathbf{x}_j)}{2 \times \sum_{j=1}^{n} u_{jv}^2}$$

where dissimilarities $d(\mathbf{x}_i, \mathbf{x}_j)$ are known and memberships u_{iv} an u_{jv} are unknown.

Hierarchical Algorithms

A hierarchical agglomerative algorithm starts with each object in a group of its own. Then it merges clusters until only one large cluster remains which is the whole data set. The user must select variables, choose dissimilarity or similarity measure and agglomerative procedure. At

the first step, when each object represents its own cluster, the dissimilarity $d(\mathbf{x}_i, \mathbf{x}_j)$ between objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by the chosen dissimilarity measure. However, once several objects have been linked together, we need a linkage or amalgamation rule to determine when two clusters are sufficiently similar to be linked together. Numerous linkage rules have been proposed.

The distance between two different clusters can be determined by the distance of the two closest objects in the clusters (single linkage method), the greatest distance between two objects in the clusters (complete linkage method), or average distance between all pairs of objects in the two clusters (unweighted pair-group average method). Further, this distance can be determined by weighted average distance between all pairs of objects in the two clusters (the number of objects in a cluster is used as a weight), or distance between centroids (unweighted or weighted). Moreover, we can use the method that attempts to minimize the sum of squares of differences of individual values from their average in the cluster (Ward's method).

The hierarchical approach is used in some algorithms proposed for clustering large data sets. We can mention the BIRCH [101] (Zhang, 1996) (Balanced Iterative Reducing and Clustering using Hierarchies) method as an example. Objects in the data set are arranged into subclusters, known as "cluster-features". These cluster-features are then clustered into k groups, using a traditional hierarchical clustering procedure. A cluster feature (CF) represents a set of summary statistics on a subset of the data. The algorithm consists of two phases. In the first one, an initial CF tree is built (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data). In the second one, an arbitrary clustering algorithm is used to cluster the leaf nodes of the CF tree. Disadvantage of this method is its sensitivity to the order of the objects.

Two-way Joining Algorithm

Two-way joining is useful in (the relatively rare) circumstances when one expects that both objects and variables (documents and features) will simultaneously contribute to the uncovering of meaningful patterns of clusters. The difficulty with interpreting these results may arise from the fact that the similarities between different clusters may pertain to (or be caused by) somewhat different subsets of variables. Thus, the resulting structure (clusters) is by nature not homogeneous. However, this method offers a powerful exploratory data analysis tool (the detailed description of this method is in [42] (Hartigan, 1975)).

We can explain the use of this method by a simple example. Let us suppose that we have three variables. Two of them are categorical. We know only one value of the third variable corresponding to the certain combination of categories of categorical variables. This value is a zero or one. We investigate the similarity of categories for each categorical variable on the basis of values of the third variable. If values of the third variable are written into cross-table, where categories of one variable are situated in rows and categories of the second one in columns, both "row clusters" and "column clusters" can be distinguished.

At each step of the algorithm, such pair of rows or columns is joined that are closest in a certain distance measure. The closest pair of rows (columns) makes a new row (column) by using a certain linkage rule. This algorithm can be generalized to many-way tables.

Subspace clustering

In high dimensional spaces, clusters often lie in a *subspace*. To handle this situation, some algorithms were suggested. Instead of creation of reduced matrix based on new features (obtained for example by linear combination of original features), subspaces of the original data space are investigated. The task is based on the original features which have a real meaning while linear combination of many dimensions may be sometimes hard to interpret. Subspace clustering is based on density based approach. The aim is to find subsets of features that projections of the input data include high density regions. The principle is partitioning of each dimension into the same number of equal length intervals. The clusters are unions of connected high density units within a subspace.

CLIQUE (CLustering In QUEst) suggested for numerical variables by Agrawal et al. in [2] (Agrawal, 1998). is a clustering algorithm that finds high-density regions by partitioning the data space into cells (hyper-rectangles) and finding the dense cells. Clusters are found by taking the union of all high-density cells. For simplicity, clusters are described by expressing the cluster as a DNF (disjunctive normal form) expression and then simplifying the expression.

MAFIA (Merging of Adaptive Finite Intervals (And more than a CLIQUE)) is a modification of CLIQUE that runs faster and finds better quality clusters. pMAFIA is the parallel version. MAFIA was presented by Goil et al. in [36,74] (Goil, 1999; Nagesh, 2001). The main modification is the use of an adaptive grid. Initially, each dimension is partitioned into a fixed number of cells.

Moreover, we can mention the algorithm ENCLUS (ENntropy-based CLUStering) suggested by Cheng et al. in [13] (Cheng, 1999). In comparison with CLIQUE, it uses a different criterion for subspace selection.

GRAPH CLUSTERING

Networks arising from real life are concerned with relations between real objects and are important part of modern life. Important examples include links between Web pages, citations of references in scientific papers, social networks of acquaintance or other connections between individuals, electric power grids, etc. Word "network" is usually used for what mathematicians and a few computer scientists calls graphs [75] (Newman, 2003). A *graph* (*network*) is a set of items called *nodes* (*vertices*) with connections between them, called *edges* (*links*). The study of graph theory is one of the fundamental pillars of discrete mathematics.

A *social network* is a set of people or groups of people with some pattern of contacts or interactions between them. Social networks have been studied extensively since the beginning of 20 th century, when sociologists realized the importance of the understanding how the human society is functioned. The traditional way to analyze a graph is to look at its picture, but for large networks this is unusable. A new approach to examine properties of graphs has been driven largely by the availability of computers and communication networks, that allow us to analyze data on a scale far larger than before now [37,76] (Guimerà, 2003; Newman, 2004).

Interesting source of reliable data about personal connections between people is communication records of certain kinds. For example, one could construct a network in which each vertice represents an email address and directed edges represent messages passing from

one address to another.

Complex networks such as the Web or social networks or emails often do not have an engineered architecture but instead are self-organized by the actions of a large number of individuals. From these local interactions nontrivial global phenomena can emerge as, for example, small-world properties or a scale-free distribution of the degree [75] (Newman, 2003). In *small-world networks* short paths between almost any two sites exist even though nodes are highly clustered. *Scale-free networks* are characterized by a power-law distribution of a node's degree, defined as the number of its next neighbours, meaning that structure and dynamics of the network are strongly affected by nodes with a great number of connections. There is reported in [18] (Ebel, 2002) that networks composed of persons connected by exchanged emails show both the characteristics of small-world networks and scale-free networks.

The Web can be considered as a graph where nodes are HTML pages and edges are hyperlinks between these pages. This graph is called the *Web graph*. It has been the subject of a variety of recent works aimed at understanding the structure of the Web [96] (Xiaodi, 2003).

A *directed graph* $G = (V, E)$ consists of a set of nodes, denoted V and a set of edges, denoted E. Each edge is an ordered pair of nodes (u, v) representing a directed connection from u to v. The graph $G = (V, E)$ is often represented by the *adjacency matrix* W by $|V| \times |V|$, where $w_{ij} = 1$ if $(v_i, v_j) \in E$ and $w_{ij} = 0$ in other cases. The *out-degree* of a node u is the number of distinct edges $(u, v_1) \ldots (u, v_k)$ (i.e., the number of links from u), and the *in-degree* is the number of distinct edges $(v_1, u) \ldots (v_k, u)$ (i.e., the number of links to u). A *path* from node u to node v is a sequence of edges $(u, u_1), (u_1, u_2), \ldots (u_k, v)$. One can follow such a sequence of edges to "walk" through the graph from u to v. Note that a path from u to v does not imply a path from v to u. The *distance* from u to v is one more than the smallest k for which such a path exists. If no path exists, the distance from u to v is defined to be infinity. If (u, v) is an edge, then the distance from u to v is 1.

Given a directed graph, a *strongly connected* component (*strong component* for brevity) of this graph is a set of nodes such that for any pair of nodes u and v in the set there is a path from u to v. In general, a directed graph may have one or many strong components. The strong components of a graph consist of disjoint sets of nodes. One focus of our studies will be in understanding the distribution of the sizes of strong components on the web graph.

An *undirected graph* consists of a set of nodes and a set of edges, each of which is an unordered pair $\{u, v\}$ of nodes. In our context, we say there is an edge between u and v if there is a link between u and v, without regard to whether the link points from u to v or the other way around. The *degree* $deg(u)$ of a node u is the number of edges incident to u. A path is defined as for directed graphs, except that now the existence of a path from u to v implies a path from v to u. A *component* of an undirected graph is a set of nodes such that for any pair of nodes u and v in the set there is a path from u to v. We refer to the components of the undirected graph obtained from a directed graph by ignoring the directions of its edges as the weak components of the directed graph. Thus two nodes on the web may be in the same weak component even though there is no directed path between them (consider, for instance, a node u that points to two other nodes v and w; then v and w are in the same weak component even though there may be no sequence of links leading from v to w or vice versa). The interplay of strong and weak components on the (directed) web graph turns out to reveal some unexpected properties of the Web's connectivity.

Informally we can say that two nodes are considered *similar* if there are many short paths connecting them. On the contrary, the "shortest path" distance does not necessarily decrease when connections between nodes are added, and thus it does not capture the fact that strongly connected nodes are at a smaller distance than weakly connected nodes.

The main findings about the Web structure are as follows:

- A power-law distribution of degrees [65] (Kumar, 1999): in-degree and out-degree distribution of the nodes of the Web graph follows the power law.
- A bow-tie shape [9] (Broder, 2000): the Web's macroscopic structure.
- The average path length between two Web pages: 16 [9] (Broder, 2000) and 19 [4] (Barabasi, 1999).
- Small world phenomenon [1] (Adamic, 1999): Six degrees of separation between any two Web pages.
- Cyber-communities [65] (Kumar, 1999): groups of individuals who share a common interest, together with the most popular Web pages among them.
- Self-similarity structure [17] (Dill, 2002): the Web shows a fractal structure in many different ways.

Link analysis plays an import role in understanding of the Web structure. There are three well known algorithms for ranking pages, such as, HITS, PageRank, and SALSA [87] (Schenker, 2005).

The book [87] (Schenker, 2005) describes exciting new opportunities for utilizing robust graph representations of data with common machine learning algorithms. Graphs can model additional information which is often not present in commonly used data representations, such as vectors. Through the use of graph distance a relatively new approach for determining graph similarity the authors show how well-known algorithms, such as *k*-means clustering and *k*-nearest neighbours classification, can be easily extended to work with graphs instead of vectors. This allows for the utilization of additional information found in graph representations, while at the same time employing well-known, proven algorithms.

Linear algebra background

Any $m \times n$ matrix A can be expressed as

$$A = \sum_{t=1}^{r} \sigma_t(A) u(t) v(t)^T,$$

where r is the rank of A, $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_r(A) > 0$ are its *singular values* and $u(t) \in \mathrm{R}^m$, $v(t) \in \mathrm{R}^n$, $t = 1,\ldots,r$ are its left and right singular vectors, respectively. The $u(t)$'s and the $v(t)$'s are orthonormal sets of vectors; namely, $u(i)^T u(j)$ is one if $i = j$ and zero otherwise. We also remind the reader that

$$\|A\|_F^2 = \sum_{i,j} A_{i,j}^2 = \sum_{i=1}^{r} \sigma_i^2(A)$$

$$\|A\|_2 = \max_{x \in R^n : \|x\|=1} \|Ax\| = \max_{x \in R^m : \|x\|=1} \|x^T A\| = \sigma_1(A)$$

In matrix notation, SVD is defined as $A = U\Sigma V^T$ where U and V are orthogonal (thus $U^TU = I$ and $V^TV = I$, an I matrix is the identity matrix $I = \{\delta_{ij}\}$ where δ_{ij} is the Kronecker symbol) matrices of dimensions $m \times r$ and $n \times r$ respectively, containing the left and right singular vectors of A. $\Sigma = diag(\sigma_1(A),\ldots,\sigma_r(A))$ is an $r \times r$ diagonal matrix containing the singular values of A. If we define $A_l = \sum_{t=1}^{l} \sigma_t(A)u(t)v(t)^T$, then A_l is the best rank l approximation to A with respect to the 2-norm and the Frobenius norm. Thus, for any matrix D of rank at most l, $\|A - A_k\|_2 \le \|A - D\|_2$ and $\|A - A_k\|_F \le \|A - D\|_F$. A matrix A has a "good" rank l approximation if $A - A_l$ is small with respect to the 2-norm and the Frobenius norm. It is well known that $\|A - A_k\|_F^2 = \sum_{t=l+1}^{r} \sigma_t^2(A)$ and $\|A - A_k\|_2 = \sigma_{l+1}(A)$. From basic Linear Algebra, $A_l = U_l\Sigma_l V_l^T = AV_lV_l^T = U_lU_l^TA$, where U_l and V_l are sub-matrices of U and V, containing only the top k left or right singular vectors of A respectively; for a detailed treatment of SVD see Golub and Van Loan [31] (Golub, 1989).

Eigenvector Clustering of Graphs

Donath and Hoffman [15] (Donath, 1973) introduced the use of eigenvectors for the purpose of partitioning an undirected graph in a balanced way. Since then, there has been a lot of work on spectral approaches for graph partitioning. See Chung [14] (Chung, 1997) for an excellent overview of the field. Shi and Malik [86] (Shi, 2000) showed that the eigenvectors of different matrices based on the adjacency matrix of a graph are related to different kinds of balanced cuts in a graph. Let W be the adjacency matrix of an undirected graph $G = (V, E)$ with nodes $1, 2, \ldots, n$ and let D be a diagonal matrix with $d_i = deg(i)$. Let A and B be sets of nodes and let $E(A, B)$ be the set of edges (u, v) with $u \in A$ and $v \in B$. Two subsets A and B of V, such that $A \cup B = V$ and $A \cap B = \varnothing$, define a *cut* in G, which we denote as (A, B).

The *average association* of a set A is

$$|E(A, A)| / |A|.$$

The *average cut* of a set A is

$$|E(A, V - A)| / |A| + |E(A, V - A)| / |V - A|.$$

The *normalized cut* of a set A is

$$|E(A, V - A)| / |E(A, V)| + |E(A, V - A)| / |E(V - A, V)|.$$

Then Shi and Malik show that

- the second largest eigenvector of W is related to a set that maximizes the average association;
- the second smallest eigenvector of $D - W$ (also known as the algebraic connectivity or Fiedler value [23] (Fritzke, 1974) is related to a set that minimizes the average cut; and
- the second smallest eigenvector of the generalized eigenvector problem $(D - W)x = \lambda Dx$ gives an approximation of the smallest normalized cut.

These results hold for undirected graphs, but the Web graph is a directed graph. Thus, it

would be interesting to understand what the above relationships are for directed graphs, i.e., whether the eigenvectors of the corresponding matrices of a directed graph are also related to balanced decompositions of the directed graph. It is possible that this would lead to an interesting clustering of the Web graph or for a topic-specific subgraph. The first step in this direction was taken by Gibson et al. [35] (Gibson, 1998). They used the eigenvectors of the matrix AA^T and the matrix A^TA, where A is the adjacency matrix of a topic-specific subgraph, to decompose topic-specific subgraphs. They show that the principal eigenvector and the top few nonprincipal eigenvectors decompose the topic graphs into multiple "hyperlinked communities," i.e., clusters of pages on the same subtopic [47] (Henzinger, 2003). Lot of examples of eigenvector computations we can found in the survey paper [66] (Langville, 2005).

Roughly speaking, from spectral analysis we obtain decomposition of graph to "high order" connected component [21,22] (Fiedler, 1973; Fiedler, 1975). The work [45] (He, 2001) compares clustering based on Fiedler vector [21,22] (Fiedler, 1973; Fiedler, 1975)with *k*-means clustering method and founds the results of spectral partitioning usually much better.

Connectivity Clustering of Graphs

Although there are numerous algorithms for cluster analysis in the literature, we briefly review the approaches that are closely related to the structure of a graph.

Matula [69,70,71] (Matula, 1970; Matula, 1972; Matula, 1987) uses a high connectivity in similarity graphs to cluster analysis, which is based on the cohesiveness function. The function defines every node and edge of a graph to be the maximum edge-connectivity of any subgraph containing that element. The *k-connected subgraphs* of the graph are obtained by deleting all elements with cohesiveness less than k in the graph, where k is a constant value. It is hard to determine the connectivity values in real clustering applications with this approach.

There are approaches using biconnected components (maximal $2-$connected subgraphs). The work [11] (Canutescu, 2003) introduces a new algorithm for protein structure prediction based on biconnected components. In [46] (Henzinger, 1997) Henzinger presents fully dynamic algorithms for maintaining the biconnected components.

There is a recent work related to clustering of a graph. The HCS algorithms [41] (Hartuv, 2000) use a similarity graph as the input data. The algorithm recursively partitions a current set of elements into two subsets. It then identifies highly connected subgraphs, in which the number of edges exceeds half the number of their corresponding nodes, as kernels among them. A kernel is considered as a cluster. Unfortunately, the result of the clustering is not uniquely determined.

The CLICK algorithm [85] (Sharan, 2000) builds on a statistical model. It uses the same basic scheme as HCS to form kernels, and includes the following processing: singleton adoption, recursive clustering process on the set of remaining singletons, and an iterative merging step.

The CAST [5] (Ben-Dor 1999) uses a single parameter t, and starts with a single object. Objects are added or removed from the cluster if their affinity is larger or lower than t, respectively, until the process stabilizes.

In [96] (Xiaodi Huang, 2003) there are introduced definitions of homogeneity and separation to measure the quality of a graph clustering.

In [93] (White, 2005) Newman's Q function is used for graph embedding into Euclidean

space. This representation is used for fast geometric clustering.

Combined methods

Deng Cai et al. in [10] (Cai, 2004) described a method to organize Web image search results. Based on the Web context, they proposed three representations for Web images, i.e. representation based on a visual feature, representation based on a textual feature and representation induced from the image link graph. Spectral techniques were applied to cluster the search results into different semantic categories. They show that the combination of textual feature based representation and graph based representation actually reflects the semantic relationships between Web images.

In [67] (Lian, 2004) the algorithm S-GRACE is presented. S-GRACE is a hierarchical clustering algorithm on XML documents, which applies categorical clustering algorithm (ROCK [38] (Guha, 1999) on the *s*-graphs (structure graph) extracted from the XML documents.

For two XML documents $\mathbf{x}_i$ and $\mathbf{x}_j$, the distance between them is defined by

$$d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{|\, sg(\mathbf{x}_i) \cap sg(\mathbf{x}_j)\,|}{\max(|\, sg(\mathbf{x}_i)\,|, |\, sg(\mathbf{x}_j)\,|)}$$

where $sg(\mathbf{x}_i)$ is a structure graph ($i = 1, 2$), $|\, sg(\mathbf{x}_i)\,|$ is the number of edges in $sg(\mathbf{x}_i)$; and $sg(\mathbf{x}_i) \cap sg(\mathbf{x}_j)$ is the set of common edges of $sg(\mathbf{x}_i)$ and $sg(\mathbf{x}_j)$.

ARTIFICAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) belong to the adaptive class of techniques in the machine learning area. ANNs try to mimic the biological neural network, the brain to solve basic computationally hard problems of AI.

There are three important, and attractive, features of ANNs:

- it is their capability of learning from example (extracting knowledge from data),
- there are natural parallel, and thus should be computationally effective, and
- they should work incrementally - not whole data set is necessary at once.

This feature makes ANNs a very interesting and promising clustering choice for large data sets including multimedia and text files.

Most models of ANNs are organized in the form of a number of processing units (also called artificial neurons, or simply neurons [72] (McCulloch, 1943), and a number of weighted connections (artificial synapses) between the neurons. The process of building an ANN, similar to its biological inspiration, involves a learning episode (also called training). During learning episode, the network observes a sequence of recorded data, and adjusts the strength of its synapses according to a learning algorithm and on the observed data. The process of adjusting the synaptic strengths in order to be able to accomplish a certain task, much like the brain, is called learning. Learning algorithms are generally divided into two types, supervised

and unsupervised. The supervised algorithms require labelled training data. In other words, they require more a priori knowledge about the training set.

There is a very large body of research that has resulted in a large number of ANN designs. For a more complete review of the various ANN types see [43,82] (Hassoun, 1995; Rumelhart, 1988). In this chapter, we discuss only some of the types that have been used data mining area.

Layered, feed-forward, backpropagation neural networks

These are a class of ANNs whose neurons are organized in layers. The layers are normally fully connected, meaning that each element (neuron) of a layer is connected to each element of the next layer. However, self-organizing varieties also exist in which a network starts either with a minimal number of synaptic connections between the layers and adds new ones as training progresses (*constructive*), or starts as a fully connected network and prunes connections based on the data observed in training (*destructive)* [43,82].

Backpropagation [82] is a learning algorithm that, in its original version, belongs to the gradient descent optimization methods [94]. The combination of backpropagation learning algorithm and the feed-forward, layered networks provide the most popular type of ANNs. These ANNs have been applied to virtually all pattern recognition problems, and are typically the first networks tried on a new problem. The reason for this is the simplicity of the algorithm, and the vast body of research that has studied these networks. As such, in sequencing, many researchers have also used this type of network as a first line of attack. Examples can be mentioned in [94,95]. In [94] Wu has developed a system called gene classification artificial neural system (GenCANS), which is based on a three layered, feed-forward backpropagation network.

Self-organizing neural networks

These networks are a very large class of neural networks whose structure (number of neurons, number of synaptic connections, number of modules, or number of layers) changes during learning based on the observed data. There are two classes of this type of networks: destructive and constructive. Destructive networks are initially a fully connected topology and the learning algorithm prunes synapses (sometime entire neurons, modules, or layers) based on the observed data. The final remaining network after learning is complete, usually is a sparsely connected network. Constructive algorithms start with a minimally connected network, and gradually add synapses (neurons, modules, or layers) as training progresses, in order to accommodate for the complexity of the task at hand.

Self-Organizing Map. A self-organizing map (SOM) [61] is a neural network paradigm first proposed by Kohonen [62]. SOMs have been used as a divisive clustering approach in many areas. Several groups have used SOMs to discover patterns clusters in Web pages or in textual documents [3]. Special version of this paradigm WEBSOM was developed for Web pages clustering [56,63]. With the WEBSOM method a textual document collection is organized onto a graphical map display that provides an overview of the collection and facilitates interactive browsing. Interesting documents can be located on the map using a content-directed search. Each document is encoded as a histogram of term categories which are formed by the SOM algorithm based on the similarities in the contexts of the terms. The

encoded documents are organized on another self-organizing map, a document map, on which nearby locations contain similar documents. Special consideration is given to the computation of very large document maps which is possible with general-purpose computers if the dimensionality of the term category histograms is first reduced with a random mapping method and if computationally efficient algorithms are used in computing the SOMs.

SOM as a clustering method has some disadvantages. One of them is necessity of introduction of decay coefficient that stops the learning (clustering) phase. If the map is allowed to grow indefinitely, the size of SOM is gradually increased to a point when clearly different sets of expression patterns are identified. Therefore, as with *k*-means clustering, the user has to rely on some other source of information, such as PCA, to determine the number of clusters that best represents the available data. For this reason, Sasik [84] and his colleagues believe that "SOM, as implemented by Tamayo et al. [88], is essentially a restricted version of *k*-means: Here, the k clusters are linked by some arbitrary user-imposed topological constraints (e.g. a 3×2 grid), and as such suffers from all of the problems mentioned above for *k*-means (and more), except that the constraints expedites the optimization process". [84] There are many varieties to SOM, among which the self-organizing feature maps (SOFM) should be mentioned [61,62]. The *growing cell structure* (GCS) [23] is another derivative of SOFM. It is a selforganizing and incremental (constructive) neural learning approach.

Self-organizing trees. Self-organizing trees are normally constructive neural network methods that develop into a tree (usually binary tree) topology during learning. Among examples of these networks the work of Dopazo et al. [16], Wang et al. [91], and Herrero et al. [49] can be mentioned. Dopazo and Carazo introduce the self-organizing tree algorithm (SOTA) [16]. SOTA is a hierarchical neural network that grows into a binary tree topology. For this reason SOTA can be considered a hierarchical clustering algorithm. SOTA is based on Kohonen's SOM discussed above and Fritzke's growing cell [23]. The SOTA's performance is superior to that of classical hierarchical clustering techniques. Among the advantages of SOTA as compared to hierarchical cluster algorithms are its lower time complexity, and its top-to-bottom hierarchical approach. SOTA's runtimes are approximately linear with the number of items to be classified, making it suitable for large data sets. Also, because SOTA forms higher clusters in the hierarchy before forming the lower clusters, it can be stopped at any level of hierarchy and still produces meaningful intermediate results. There are many other types of self-organizing trees.

Recurrent ANNs

ART and its derivatives. *Adaptive Resonance Theory* was introduced by Stephen Grossberg [33,34] in 1976. Networks based on ART are unsupervised and self-organizing, and only learn in the so called "resonant" state. ART can form (stable) clusters of arbitrary sequences of input patterns by learning (entering resonant states) and self-organizing. Since the inception, many derivatives of ART have emerged. Among these ART-1 (the binary version of ART; forms clusters of binary input data) [12], ART-2 (analog version of ART) [24], ART-2A (fast version of ART-2) [25], ART-3 (includes "chemical transmitters" to control the search process in a hierarchical ART structure) [26], ARTMAP (supervised version of ART) [27] can be mentioned. Many hybrid varieties such as Fuzzy-ART [28], Fuzzy-ARTMAP (supervised Fuzzy-ART) [29,30] and simplified Fuzzy-ARTMAP (SFAM) [57] have also been developed.

The ART family of networks. These networks have a broad application in virtually all areas of clustering. In general, in problem settings when the number of clusters is not previously known apriori, researchers tend to use unsupervised ART, where when the number of clusters is known a priori, usually the supervised version, ARTMAP, is used. Among the unsupervised implementations, the work of Tomida et al. [89] should be mentioned. Here the authors used Fuzzy ART for expression level data analysis. Fuzzy ART incorporates the basic features of all ART systems, notably, pattern matching between bottom-up input and top-down learned prototype vectors. This matching process leads either to a resonant state that focuses attention and triggers stable prototype learning or to a self-regulating parallel memory search. If the search ends by selecting an established category, then the category's prototype may be refined to incorporate new information in the input pattern. If the search ends by selecting a previously untrained node, then learning of a new category takes place. Fuzzy ART performs best in noisy data. Although ART has been used in several research works as a text clustering tool, the level of quality of the resulting document clusters has not been clearly established. In [68] the author presents experimental results with binary ART that address this issue by determining how close clustering quality is to an upper bound on clustering quality.

Associative Clustering Neural Networks. Since the introduction of the concept of auto-associative memory by Hopfield [51], there have been many associative memory models built with neural networks [59,64]. Most of them can be considered into store-recall models and the correlation between any two *D*-bit bipolar patterns $s(\mathbf{x}_i, \mathbf{x}_j)$, $x_{id} \in \{-1,1\}$ for all $l = 1,\ldots, p$ is often determined by a static measurement such as

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{p} \sum_{l=1}^{p} x_{il} x_{jl}.$$

The human mind, however, associates one pattern in memory to others in a much more sophisticated way than merely attempting to homogeneously link vectors. Such associations would interfere with each other [50]. To mimic the formation of such associations in cybernetics, Yao at al. [97] build a recurrent neural network to dynamically evaluate the association of any pairwise patterns through the interaction among a group patterns and incorporate the results of interaction into data clustering. The novel rule based on the characteristic of clusters has been proposed to determine the number of clusters with a reject option. Such a hybrid model they named Associative Clustering Neural Network (ACNN). The performance of ACNN has been studied by authors on simulated data only, but and the results have demonstrated that ACNN has the feasibility to cluster data with a reject option and label the data robustly.

Bayesian Neural Networks. There are a number of recent networks that have been suggested as solutions clustering. For instance, Bayesian neural networks (BNNs), are another technique that has been recently used for Web clustering. Her et al. [48] have used the BNNs for clustering Web query results. Their BNN is based on SOM and it differs in the last step when *n* documents are assigned under each cluster by Bayesian rule. The BNNs are an important addition to the host of ANN solutions that have been offered to the problem at hand, as they represent a large group of hybrid ANNs that combine classical ANNs with statistical classification and prediction theories.

WEB CLUSTERING

The Web has undergone exponential growth since its birth, which is the cause of a number of problems with its usage. Particularly, the quality of Web search and corresponding interpretation of search results are often far from satisfying due to various reasons like huge volume of information or diverse requirements for search results.

The lack of a central structure and freedom from a strict syntax allow the availability of a vast amount of information on the Web, but they often cause that its retrieval is not easy and meaningful. Although ranked lists of search results returned by a search engine are still popular, this method is highly inefficient since the number of retrieved search results can be high for a typical query. Most users just view the top ten results and therefore might miss relevant information. Moreover, the criteria used for ranking may not reflect the needs of the user. A majority of the queries tend to be short and thus, consequently, non-specific or imprecise. Moreover, as terms or phrases are ambiguous in the absence of their context, a large amount of search results is irrelevant to the user.

In an effort to keep up with the tremendous growth of the Web, many research projects were targeted on how to deal its content and structure to make it easier for the users to find the information they want more efficiently and accurately. In last years mainly data mining methods applied in the Web environment create new possibilities and challenges.

Methods of Web data mining can be divided into a number of categories according to kind of mined information and goals that particular categories set. In [79] three categories are distinguished: *Web structure mining* (WSM), *Web usage mining* (WUM), and *Web Content Mining* (WCM). Particularly, WCM refers broadly to the process of uncovering interesting and potentially useful knowledge from Web documents.

WCM shares many concepts with traditional text mining techniques. One of these, *clustering*, groups similar documents together to make information retrieval more effective. When applied to Web pages, clustering methods try to identify inherent groupings of pages so that a set of clusters is produced in which clusters contain relevant pages (to a specific topic) and irrelevant pages are separated. Generally, text document clustering methods attempt to collect the documents into groups where each group represents some topic that is different than those topic represented by the other groups. Such clustering is expected to be helpful for discrimination, summarization, organization, and navigation for unstructured Web pages.

In a more general approach, we can consider Web documents as collections of Web pages including not only HTML files but also XML files, images, etc. An important research direction in Web clustering is Web XML data clustering stating the clustering problem with two dimensions: content and structure [90].

WUM techniques use the Web-log data coming from users' sessions. In this framework, Web-log data provide information about activities performed by a user from the moment the user enters a Web site to the moment the same user leaves it. In WUM, the clustering tries to group together a set of users' navigation sessions having similar characteristics [90]. Concerning WSM techniques, graph-oriented methods described in Section 3 can be used.

Considering Web clustering techniques, it is important to be aware of two main categories of approaches:

- clustering Web pages in a space of resources to facilitate some search services and
- clustering Web search results.

In [8] these categories are called *offline clustering* and *online clustering*, respectively. We mention approaches of both categories although the main accent is put on the latter.

Application of Web clustering

Web clustering is currently one of the crucial IR problems related to Web. It is used by many intelligent software agents in order to retrieve, filter, and categorize Web documents. Various forms of clustering are required in a wide range of applications: efficient information retrieval by focusing on relevant subsets (clusters) rather than whole collections, clustering documents in collections of digital libraries, clustering of search results to present them in an organized and understandable form, finding mirrored Web pages, and detecting copyright violations, among others.

Clustering techniques are immensely important for Web applications to assist the automated (or semiautomated) generation of proper categories of documents and organize repositories of search engines. Hierarchical categorization of documents is often used (see Google, Yahoo, Open Directory, and LookSmart as examples). The reason is that the search results are not summarized in terms of topics; they are not well suited for browsing tasks. One possible solution is to create manually a static hierarchical categorization of a reasonable part of the Web and use these categories to organize the search results of a particular query. However, this solution is feasible only for small collections. To categorize the entire Web either manually or automatically is, unfortunately, not real.

In [78], document clustering and a WUM technique are used for construction of Web Community Directories, as a means of personalizing Web services. Also effective summarization of Web page collections becomes more and more critical as the amount of information continues to grow on the Web. The significance of Web collection clustering for automatic Web collection summarization is investigated in [100].

Clustering is also useful in extracting salient features of related Web documents to automatically formulate queries and search for other similar documents on the Web.

Principles of Web clustering methods

Most of the documents clustering methods that are in use today are based on the VSM. A similarity between documents is measured using one of several similarity measures that are based on relations of feature vectors, e.g. cosine of feature vectors (see Section 2.1). Many of traditional algorithms based on VSM, however, falter when the dimensionality of the feature space becomes high relative to the size of the document space. In a high dimensional space, the distance between any two documents tends to be constant, making clustering on the basis of distance ill-defined. This phenomenon is called a *curse of dimensionality*. Therefore the issue of reducing the dimensionality of the space is critical. The methods presented in Section 2 are often used.

Traditional clustering algorithms either use a priori knowledge of document structures to define a distance or similarity among these documents, or use probabilistic techniques such as Bayesian classification.

Taxonomies are generated using document clustering algorithms which typically result in topic or concept hierarchies. This classification and clustering techniques are combined. Concept hierarchies expose the different concepts presented in the Web pages (or search result) collection. The user can choose the concept he/she is interested in and can browse it in

detail.

Classification of Web clustering methods

Generally, clustering approaches could be classified in two broad categories [92]: *term-based clustering* and *link-based clustering*. Recent work in online clustering has included both link-based and term-based methods.

Term-based clustering. We start with methods where each term is a single word. Zamir et al. mention in [98] very simple *word-intersection clustering* method, where words that are shared by documents are used to produce clusters. Let n denote the number of documents to be clustered. The method runs in $O(n^2)$ and produces good results for Web documents originating rather from on a corpus of texts. We point out that standard methods such as *k*-means, are also in this category since they usually exploit single words as features. Most of methods based on VSM belong to this category. They do not make use of any word proximity or phrase-based approach.

Word-based clustering that is used on common words shared among documents does not adapt well to Web environment since it ignores the availability of hyperlinks between Web pages and is susceptible to spam. Also the curse of dimensionality restricts a usability of these methods. A more successful clustering in this case (also ignoring links among documents) is based on multi-word terms (phrases, sentences). Then we speak about *term-based clustering* [100]. Extracting terms significantly reduces the high dimensionality. Authors of [100] show that this reduction is almost an order of magnitude while maintaining comparable performance with word-based model.

Among first works using phrases in clustering we find approach [99] based on Suffix Tree Clustering (STC). STC firstly transforms the string of text representing each document to a sequence of stems. Secondly, it identifies the sets of documents that shared a common phrase as base clusters by a suffix tree. Finally, these base clusters are combined into clusters. Tree building often requires $O(n \log n)$ time and produces high quality clusters. On the other hand, the suffix tree model can have a high number of redundancies in terms of the suffixes stored in the tree. However, the STC clustering based on phrases shared between documents generates inferior results to those based on the full text of the document.

In [39] a system for Web clustering is based on two key concepts. The first is the use of weighted phrases as an essential constituent of documents. Similarity between documents will be based on matching phrases and their weights. The similarity calculation between documents combines single-word similarity and phrase-based similarity. The latter is proven to have a more significant effect on the clustering quality due to its insensitivity to noisy terms that could lead to incorrect similarity measure. The second concept is the incremental clustering of documents using a histogram-based method to maximize the tightness of clusters by carefully watching the similarity distribution inside each cluster. In the system a novel phrase-based document index model is used, the Document Index Graph (DIG), that captures the structure of sentences in the document set, rather than single words only. The DIG model is based on graph theory and utilizes graph properties to match any-length phrase from a document to any number of previously seen documents in a time nearly proportional to the number of words of the document. Improvement over traditional clustering methods was 10 to 29 percent.

Link-based clustering. Links among Web pages could provide valuable information to determine the related page since they give objective opinions for the topic of the pages they point to. Many works tried to explore link analysis to improve the term-based methods. In general these methods belong to the category of graph clustering (Section 3). Kleinberg in [60] suggested that there are two kinds of pages on the Web for a specific query topic: hub and authority and they reinforce each other. HITS algorithm, which was used to locate hubs and authorities from the search results given a query topic, provided a possible way to alleviate the problems. However, sometimes one's "most authoritative" pages are not useful for others. It is also observable that many "authority" pages contain very little text. The work [92] combines successfully link-based features (co-citations and bibliographic coupling) and contents information in clustering. Co-citation measures the number of citations (out-links) in common between two documents and coupling measures the number of document (in-links) that cites both of two documents under consideration.

Structure of clusters. Two clustering algorithms that can effectively cluster documents, even in the presence of a very high dimensional feature space are described in [44]. These clustering techniques, which are based on generalizations of graph partitioning, do not require pre-specified ad hoc distance functions, and are capable of automatically discovering document similarities or associations.

As we mentioned in Introduction, most of clustering methods can be divided into two categories: hierarchical clusters and flat clusters. Hierarchical clustering is exceedingly slow when it is used for online for very high n. Its implementing time can be from $O(n^2)$ up to $O(n^3)$.

The flat clustering algorithms are model-based algorithms that search for the model parameters given the data and prior expectation. For example, *k*-means is $O(nkT)$ algorithm, where T is the number of iterations, but the task to determine model describing data complicates its use for large collections, particularly in a Web environment.

Clustering with snippets

Today search engines return with a ranked list of search results also some contextual information, in the form of a Web page excerpt, the so called *snippet*. *Web-snippet clustering* is an innovative approach to help users in searching the Web. It consists of clustering the snippets returned by a (meta-) search engine into a hierarchy of folders which are labelled with a term. The term expresses latent semantics of the folder and of the corresponding Web pages contained in the folder. The folder labels vary from a bag of words to variable-length sentences.

Web-snippet clustering methods are classified in [20] according to two dimensions: words vs. terms and flat vs. hierarchical. Four categories of approaches are distinguished.

Word-based and flat clustering. This category includes systems like SCATTER-GATHER and WEBCAT. Other systems use e.g. fuzzy relations [55] or take into account in-linking and out-linking pages to improve precision.

Term-based and flat clustering. Authors of [100] used sentences of variable length to label the folders, but these sentences were drawn as contiguous portions of the snippets by means of a Suffix Tree data structure. Other systems use SVD on a term-document matrix to find meaningful long labels. This approach is restricted by the time complexity of SVD applied to a large number of snippets. In addition, the similar snippets can lead to very high overlap, means of a STC.

Word-based and hierarchical clustering. There are approaches based on the Frequent Itemsets Problem and a concept lattice [81] on single words in order to construct the folder hierarchy.

Term-based and hierarchical clustering. This class includes the best meta-search engines of the years 2000-2003 Vivisimo and Dogpile. These tools add to the flat list of search results a hierarchy of clusters built on-the-fly over snippets. It improves precision over recall by using a snippet representation made of pair of words (not necessarily contiguous) linked by a lexical affinity, i.e. a correlation of their common appearance. Among older approaches there is a simple extension of Grouper [99] to hierarchical clustering based on the size of folders overlap. A hierarchical engine SNAKET introduced in [20] organizes on-the-fly the search results from 16 commodity search engines and offers folder labelling with variable-length sentences. Hierarchies are overlapping because snippet might cover multiple themes.

CONCLUSION

Clustering is currently one of the most crucial techniques for

- dealing with massive amount of heterogeneous information on the Web,
- organizing Web search results.

Unlike clustering in other fields, Web clustering separates unrelated pages and clusters related pages (to a specific topic) into semantically meaningful groups, which is useful for discrimination, summarization, organization and navigation of unstructured Web pages. In this chapter we have presented a lot of general approaches to clustering as well as a lot of various classifications of clustering algorithms. Consequently, two important questions arise:

- why so many clustering algorithms and
- which of them are usable for Web clustering?

In his paper [19] Estivill-Castro tries to answer the first question in terms of the model of data to be clustered and the cluster model (inductive principle in his terminology). For a single model of data and a cluster model there are many clustering algorithms. As there are cluster models and many algorithms for each cluster models, there are many clustering algorithms. And why are here so many clustering models? Because clustering is in part beholder dependent. Cluster models are just formal models of what researchers believe is a definition of cluster. Thus, it is very hard to compare particular approaches.

To answer the second question, we can first consider the techniques that are not usable for Web clustering. Observe that clustering in a Web environment eliminates naturally a use of some general clustering techniques. The reason is easy. Since clustering translates into optimization problem, its computational complexity is typically intractable in the case of huge

Web data collections.

Another reason for inapplicability of some classical techniques is associated with usability of the clustering achieved. Given a large document collection, it is difficult to provide the number of real categories for users when they attempt to categorize the documents. Organizing Web search results into a hierarchy of topics and subtopics facilitates browsing the collection and locating results of interest. Traditional clustering techniques are inadequate for Web since they do not generate clusters with highly readable names. It seems that Web-snippet clustering methods deal successfully with this issue. We have also mentioned how link information can be used to improve classification results for Web collections. In practice, it is desirable to combine term-based clustering and link-based clustering.

This survey represents only a small part of the research being conducted in the area. Furthermore, as new techniques and algorithms are being proposed for Web data sets, it makes survey such as this highly time dependent.

Acknowledgement

The work was partly supported by the project 1ET100300419 of the Program Information Society of the Thematic Program II of the National Research Program of the Czech Republic and the project 201/05/0079 of the Grant Agency of the Czech Republic.

REFERENCES

[1] Adamic, L. A. (1999). *The Small World Web*, S. Abiteboul, A. M. Vercoustre (Eds.): ECDL'99, LNCS 1696, 443-452.

[2] Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *ACM SIGMOD Record*, 27, 2, 94-105.

[3] Anonymous: *5384 works that have been based on the Self-OrganizingMap (SOM) method developed by Kohonen*, Part I : http://www.cis.hut.fi/research/som-bibl/references_a-k.ps, 1.4MB resp. Part II: http://www.cis.hut.fi/research/som-bibl/references_l-z.ps, / 1.3MB 4.08.2005

[4] Barabasi, A.L., & Albert, R. (1999). Emergence of scaling in random networks, *Science, Vol 286, Issue 5439, 15 October 1999*, 509-512.

[5] Ben-Dor, A., Shamir, R., & Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of Computational Biology (1999)*, 6(3/4): 281-297.

[6] Berkhin, P. (2002). Survey of Clustering Data Mining Techniques. *Accrue Software, Inc., San Jose, 2002.* www.ee.ucr.edu/barth/EE242/clustering_survey.pdf

[7] Berry, M.W., & Browne, M. (1999). Understanding Search Engines: Mathematical Modeling and Text Retrieval (Software, Environments, Tools). *Society for Industrial & Applied Mathematics.*

[8] Boley, D., Gini M., Gross R., Han, E.-H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore. J. (1999). Partitioning-Based Clustering for Web Document Categorization. *Journal of Decision Support Systems, Vol. 27, No. 3,* 329-341.

[9] Broder, A., Kumar, R., Maghoul, R., Raghavan, P., Rajagopalan, P., Stata, R., Tomkins,

A., & Wiener, J. (2000). Graph structure in the Web, *The 9th international WWW Conference (2000) Amsterdam*, The Netherlands. http://www9.org/w9cdrom/160/160.html

[10] Cai, D., He, X. Li, Z., Ma, W., & Wen, J. (2004). Hierarchical Clustering of WWW Image Search Results Using Visual, Textual and Link Information. *ACM MM'04, October 10-16, 2004*, New York, New York, USA.

[11] Canutescu, A.A., Shelenkov, A.A., & Dunbrack R.L. (2003). A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science*, 12: 2001-2014.

[12] Carpenter G.A., & Grossberg, S.: (1987). Invariant pattern recognition and recall by an attentive self-organizing art architecture in a nonstationary world. *In Proceedings of the IEEE First International Conference on Neural Networks*, June 1987, 737-745.

[13] Cheng, C., Fu, A.W., & Zhang, Y. (1999). Entropy-Based Subspace Clustering for Mining Numerical Data. In: *Proc. of 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, San Diego, 84-93.

[14] Chung, F. R. K. (1997). Spectral Graph Theory. *In CBMS Regional Conference Series in Mathematics, Volume 92. Providence, RI: American Mathematical Society*.

[15] Donath, W. E., & Hoffman, A. J. (1973). Lower Bounds for the Partitioning of Graphs. *IBM Journal of Research and Development 17* 420-425.

[16] Dopazo J., & Carazo J. M. (1997). Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of Molecular Evolution 44,* 226-233.

[17] Dill, S., Kumar, R., McCurley, K., Rajagopalan, S., Sivakumar, D., & Tomkins, A. (2002). Self-similarity in the Web, ACM Trans. *Internet Techn. 2(3):* 205-223.

[18] Ebel, H., Mielsch, L.I., & Bornholdt, S. (2002). Scale-free topology of e-mail networks, *Phys. Rev. E, 66 (2002), art. no. 035103.*

[19] Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter, Volume 4, Issue 1 (June 2002)*, 65-75.

[20] Ferragin, P., & Gulli, A. (2005). A personalized search engine based on Web-snippet hierarchical clustering. *In: Proc. of 14th international conference on World Wide Web 2005,* Chiba, Japan, 801-810.

[21] Fiedler, M. (1973). Algebraic connectivity of graphs. *Czech. Math. J., 23:* 298-305.

[22] Fiedler, M. (1975). A property of eigenvectors of non-negative symmetric matrices and its applications to graph theory. *Czech. Math. J., 25(100):* 619-633.

[23] Fritzke, B. (1974). Growing cell structures–A self-organizing network for unsupervised and supervised learning. *Neural Network 7:* 1141-1160.

[24] Carpenter, G. A., & Grossberg, S. (1987). Art 2: Selforganisation of stable category recognition codes for analog input patterns. *Applied Optics, 26:* 4919-4930.

[25] Carpenter, G. A., Grossberg, S., & Rosen, D.B. (1991). Art2-a: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks, 4:* 493-504.

[26] Carpenter, G. A., & Grossberg, S. (1990). Art3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks, 3:* 129-152.

[27] Carpenter, G. A., Grossberg, S., & Reynolds, J.H. (1991). Artmap: Supervised real-time

learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks, 4:* 565-588.

[28] Carpenter, G. A., Grossberg, S., & Rosen, D.B. (1991). Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4:* 759-771.

[29] Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J.H., & Rosen, D.B. (1992). Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks, 3(5)*: September 1992, 698-713.

[30] Carpenter, G. A., Grossberg, S., & Reynolds, J.H. (1995). A fuzzy artmap nonparametric probability estimator for nonstationary pattern recognition problems. *IEEE Transactions on Neural Networks, 6(6): November 1995*, 1330-1336.

[31] Golub, G., & Loan, Van C. (1989). *Matrix computations*. Johns Hopkins University Press.

[32] Gordon, A.D. (1999). *Classification, 2nd Edition.* Chapman & Hall/CRC, Boca Raton.

[33] Grossberg, S. (1988). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23: 1976*. Reprinted in Anderson and Rosenfeld, 121-134.

[34] Grossberg, S. (1976). Adaptive pattern recognition and universal recoding: II. Feedback, expectation, olfaction, and illusions. *Biological Cybernetics, 23:* 187-202.

[35] Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Inferring Web Communities from Link Topology. *In Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, New York: ACM Press, 225-234.

[36] Goil, S., Nagesh, H., & Choudhary, A. (1999) *MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets*. Technical Report No. CPDC-TR-9906-010, Northwestern University.

[37] Guimerà, R., Danon, L., Díaz-Guilera, A., F. Giralt, F., & Arenas, A. (2003) Self-similar community structure in a network of human interactions. *Physical Review, vol. 68 (2003)*, 065103.

[38] Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A Robust Clustering Algorithm For Categorical Attributes, *Proc. 15th Int'l Conf. Data Eng.,* 512-521.

[39] Hammouda, K.M., & Kamel, M.S. (2004). Efficient Phrase-Based Document Indexing for Web Document Clustering. *IEEE Transactions on Knowledge and Fata Engineering, Vol. 18, No. 10*, October 2004, 1279-1296.

[40] Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco.

[41] Hartuv, E., & Shamir, R. (2000). A clustering algorithm based on graph connectivity. *Information Processing Letters,* 76 (4-6): 175-181.

[42] Hartigan, J.A. (1975). *Clustering Algorithms*. John Wiley & Sons, New York.

[43] Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks*. MIT Press.

[44] Haveliwala, T., Gionis, A., & Indyk, P. (2000). Scalable Techniques for Clustering the Web. In *Proceedings of WebDB*.

[45] He, X., Ding C. H. Q., Zha, H., & Simon, H. D. (2001). Automatic Topic Identification Using Webpage Clustering. *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 01),* 195-203.

[46] Henzinger, M. R. (1997). *Improved Data Structures for Fully Dynamic Biconnectivity*, report, Digital Equipment Corporation.

[47] Henzinger, M. R. (2003). Algorithmic Challenges in Web Search Engines. *Internet Mathematics, Vol. 1, No. 1*: 115-126.

[48] Her, J.H., Jun, S.H., Choi, J.H, & Lee, J.H. (1999). A Bayesian Neural Network Model for Dynamic Web Document Clustering, In. *Proceedings of the IEEE Region 10 Conference (TENCON 99)*, Vol. 2, 1415-1418.

[49] Herrero, J., Valencia, A., & Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics, 17:* 126-136.

[50] Hinton, G. E., & Anderson, J. A. (1989). *ParallelModels of Associative Memory*, New Jersey: Hillsidale.

[51] Hopfield, J. J. (1982). Neural Network and Physical Systems with Emergent Collective Computational Abilities, *Proc. Acad. Sci. U.S.A. 79,* 2554-2558.

[52] Höppner, F., Klawon, F., Kruse, R., & Runkler, T. (2000). *Fuzzy Cluster Analysis. Methods for Classification, Data Analysis and Image Recognition.*, Wiley, New York.

[53] Jain, A.K., & Dubes, R.C. (1988). Algorithms for Clustering Data. Prentice Hall. New Jersey.

[54] Jain, A.K., Murty, M.N., & Flynn, P.J. (1999). Data Clustering: A Review. *ACM Computing Surveys, Vol. 31, No. 3,* 264-323.

[55] Joshi, A., & Jiang, Z. (2002). Retriever: *Improving Web Search Engine Results Using Clustering*, Idea Group Publishing.

[56] Kaski, S. Honkela, T., Lagus, K., & Kohonen T. (1998). WEBSOM - Self-organizing maps of document collections, *Neurocomputing 21,* 101-117.

[57] Kasuba, T. (1993). Simplified fuzzy ARTMAP, *AI Expert, November (1993)*, 18-25.

[58] Kaufman,L., & Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

[59] Kawamura, M., Okada, M., & Hirai, Y. (1999). Dynamics of Selective Recall in an Associative Memory Model with One-to-Many Associations, *IEEE Trans. Neural Networks 10(3),* 704-713.

[60] Kleinberg, J.M. (1999). Authoritative sources in a hyperlinked environment. *JACM, Volume 46, Issue 5,* 604-632.

[61] Kohonen, T. (2001). *Self-Organizing Maps, Third Extended Edition*. Springer, Berlin, Heidelberg, New York.

[62] Kohonen, T. (1991). Self-Organizing Maps, *Proc. IEEE 78*, 1464-1480.

[63] Kohonen, T., Kaski, S., Lagus, K., Salogärui, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *IEEE Trans. Neural Netw. 11*, 574-585.

[64] Kosko, B. (1987). Adaptive Bidirectional Associative Memories, *Appl. Opt. 26(23),* 4947-4960.

[65] Kumar, S.R., Raghavan, P., Rajagopalan, S., & Tomkins, A. (1999). Trawling the Web for Emerging Cyber Communities. *Proc. of the 8th WWW Conference,* 403-416.

[66] Langville, A. N., & Meyer, C. D. (2005). A Survey of Eigenvector Methods for Web Information Retrieval. *SIAM Review, Vol. 47, No . 1,* 135-161.

[67] Lian, W., Cheung, D.W.L., Mamoulis, N., & Yiu, S.M. (2004). An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. *IEEE Trans. Knowl. Data Eng. 16(1):* 82-96.

[68] Massey, L. (2003). On the quality of ART1 text clustering: *Neural Networks Volume 16, Issue 5-6 (June 2003)*, 771-778.

[69] Matula, D.W. (1970). Cluster analysis via graph theoretic techniques. In R.C Mullin, K.B Reid, and D. Roselle, editors, Proc. Louisiana Conference on Combinatorics, *Graph Theory and Computing,* 199-212.

[70] Matula, D.W. (1972). K-Components, clusters and slicings in graphs. *SIAM J.Appl. Math. 22(3)* 459-480.

[71] Matula, D.W. (1987). Graph theoretic techniques for cluster analysis algorithms. In J. Van Ryzin, editor, *Classification and Clustering,* 95-129.

[72] McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5: 115-133.

[73] Mercer, D.P. (2003). *Clustering large datasets*. Linacre College.

[74] Nagesh, H., Goil, S., & Choudhary, A. (2001). Adaptive grids for clustering massive data sets, *In Proceedings of the 1st SIAM ICDM, Chicago, IL.* 477.

[75] Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review, vol. 45,* 167-256.

[76] Newman, M. E. J., Balthrop, J., Forrest, S., & Williamson, M. M. (2004). Technological networks and the spread of computer viruses. *Science, vol. 304,* 527-529.

[77] Ng, R. T., & Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. *Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile,* 144-155.

[78] Pierrakos, D., Paliouras, G., Papatheodorou, C., Karkaletsis, V., & Dikaiakos M. (2003). Construction of Web Community Directories using Document Clustering and Web Usage Mining. *ECML/PKDD - 2003, First European Web Mining Forum*, Ed. by Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, Cavtat - Dubrovnik, Croatia.

[79] Pal, S.K., Talwar V, & Mitra P. (2002). Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. *IEEE Transactions on Neural Networks, Vol. 13, No. 5*, 1163-1177.

[80] Řezanková, H., Húsek, D., & Snášel, V. (2004). Clustering as a Tool for Data Mining. In: Klíma, M. (ed.). *Applications of Mathematics and Statistics in Economy*. Professional Publishing, Praha, 203-208.

[81] Rice, M.D., & Siff M. (2001). Clusters, Concepts, and Pseudo-metrics. *Electronic Notes in Theoretical Computer Science, Elsevier, Volume 40, March,* 323-346.

[82] Rumelhart, D.E., & McClelland, J.L. (1988). Parallel Distributed Processing: *Explorations in the Microstructure of Cognition, Vols. 1 and 2*. MIT Press, Cambridge, MA.

[83] Salton, G., & Buckley, C. (1988). Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management, 24 5,* 513-523.

[84] Sásik, R., Hwa, T., Iranfar, N. & Loomis, W. F. (2001). Percolation Clustering: A Novel Approach to the Clustering of Gene Expression Patterns, in *Dictyostelium Development PSB Proceedings 6*, 335-347.

[85] Sharan, R., & Shamir, R. (2000). CLICK: A clustering algorithm for gene expression analysis. In S. Miyano, R. Shamir, and T. Takagi, editors, *Currents in Computational Molecular Biology (2000) 6-7*. Universal Academy Press.

[86] Shi, J., & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22:8,* 888-905.

[87] Schenker, A., Kande, A., Bunke, H., & Last, M. (2005). *Graph-Theoretic Techniques for Web Content Mining*, World Scientific Publishing.

[88] Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., & Golub, T.R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differtiation, *Proc. Natl. Acad. Sci. USA 96*, 2907-2912.

[89] Tomida, S., Hanai, T., Honda, H., & Kobayashi, T. (2001). Gene Expression Analysis Using Fuzzy ART, *Genome Informatics 12:* 245-246.

[90] Vakali, A., Pokorný, J., & Dalamagas, Th. (2004). An Overview of Web Clustering Practices. In: Current Trends in Database Technology, International Workshop on Database Technologies for Handling XML information on the Web, *DataX, EDBT 2004, Heraklion - Crete, Greece. Vol. 3268 of LNCS Springer-Verlag,* 597-606.

[91] Wang, H.C, Dopazo, J., & Carazo, J.M. (1998). Self-organizing tree growing network for classifying amino acids, *Bioinformatics 14(4),* 376-377.

[92] Wang, Y., & Kitsuregawa, M. (2002). Evaluating Contents-Link Web Page Clustering for Web Search Results. *CIKM'02 November 4-9, 2002, McLean, Virginia, USA, ACM*, 499-506.

[93] White, S., & Smyth, P. (2005). A Spectral Clustering Approach To Finding Communities in Graph. *SDM.*

[94] Wu, C. H. (1995). Chapter titled "Gene Classification Artificial Neural System" in Methods In Enzymology: *Computer Methods for Macromolecular Sequence Analysis*, Edited by Russell F. Doolittle, Academic Press, New York.

[95] Wu, Cathy, S. Zhao, Chen, H. L., Lo, C. J., & McLarty, J. (1996). Motif identification neural design for rapid and sensitive protein family search. *CABIOS 1996, 12 (2)*, 109-118.

[96] Xiaodi Huang, & Wei Lai. (2003). Identification of Clusters in the Web Graph Based on Link Topology, *Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS'03)*, 123-130.

[97] Yuhui Yao, Lihui Chen, & Yan Qiu Chen. (2001). Associative Clustering for Clusters of Arbitrary Distribution Shapes, *Neural Processing Letters 14:* 169-177.

[98] Zamir, O., Etzioni, O., Madanim, O., & Karp, R.M. (1997). Fast and Intuitive Clustering of Web Documents. *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining*, 287-290.

[99] Zamir, O., & Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration, *Proceedings of the 21st International ACM SIGIR Conference on Research*

and Development in Information Retrieval, 46-54.

[100] Zamir, O., & Etzioni, O. (1999). Grouper: a dynamic clustering interface to Web search results. Computer Networks: *The International Journal of Computer and Telecommunications Networking archive Volume 31, Issue 11-16 (May 1999)*, 1361-1374.

[101] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. *ACM SIGMOD Record, 25, 2,* 103-114.