NetCluster: a Clustering-Based Framework for Internet Tomography

Elena Baralis †, Andrea Bianco *, Tania Cerquitelli †, Luca Chiaraviglio *, Marco Mellia *
* Dip. di Elettronica, Politecnico di Torino, Italy, Email: {last name}@tlc.polito.it
† Dip. di Automatica e Informatica, Politecnico di Torino, Italy, Email: {name.surname}@polito.it

***Abstract*— In this paper, Internet data collected via passive measurement are analyzed to obtain localization information on nodes by clustering (i.e., grouping together) nodes that exhibit similar network path properties. Since traditional clustering algorithms fail to correctly identify clusters of homogeneous nodes, we propose a novel framework, named "NetCluster", suited to analyze Internet measurement datasets. We show that the proposed framework correctly analyzes synthetically generated traces. Finally, we apply it to real traces collected at the access link of our campus LAN and discuss the network characteristics as seen at the vantage point.**

I. Introduction and Motivations

The Internet is a complex distributed system which continues to grow and evolve. The unregulated and heterogeneous structure of the current Internet makes it challenging to obtain information on the actual configuration of the network. Network tomography [1] has emerged as a powerful tool to infer internal network properties from passive, end-to-end, network measurements. Therefore, network tomography is a fundamental element to understand and predict the behavior of a large-scale system such the Internet is.

In the past, several research groups have been attracted by Internet Tomography. Starting from the seminal work of Yardi [1], in which the authors study how to derive the traffic matrix from link load measurements, several authors studied how to infer internal network properties from data measurements. Most works focus on path and topology characteristics, such as internal network loss [2], or bottleneck link identification [3], based on either sending active (multicast) probes, or passively inferring end-to-end losses exploiting TCP flow observation. Passive monitoring is usually preferred, since it permits to obtain information without injecting additional traffic, thus having no impact on network status.

More recently, a new problem attracted the attention of the research community: How to define a network-wide positioning system to locate nodes. Node position knowledge is then exploited to improve network performance. For example, considering P2P system, the knowledge of other peers position could be exploited when building the overlay topology, so that neighboring nodes are logically connected together to avoid exchanging data over long (and possibly congested) paths [4]. While several proposals have been defined to derive node position in the Internet (see [5] and [6] to cite only the landmark paper and the most recent one), to the best of our knowledge all previous works require the explicit cooperation of end systems and, possibly, specialized nodes (called landmarks) to achieve the goal. Thus, signaling and active probes are often adopted.

In this paper, we propose the idea of exploiting passive measurements to obtain information on node position in the network, so that neither signaling nor active probes must be exchanged among nodes. Network tomography ideas will be applied, since the direct measurement of node position is not feasible in the current Internet. In particular, we wish to identify sets of nodes that exhibit similar network path properties, such as delay, loss, throughput. We show that, based on the passive monitoring of traffic that is exchanged from/to a vantage point, it is possible to group nodes to form homogeneous sets. We believe that this kind of information is appealing to devise novel and more intelligent applications. For example, considering Content Delivery Networks, nodes could directly contact the closest server without leveraging on pure load-balancing techniques or centralized control schemes. Similarly, considering P2P applications, the knowledge of other peers location could be exploited to improve the structure of the overlay topology. This will be beneficial to the network as well, since it will enforce traffic flow locality properties.

Data Mining algorithms [7] will be instrumental to achieve this goal. Data mining allows sifting through large amounts of data and picking out relevant information. In particular, data clustering assigns objects to different groups based on their similarity. More precisely, clustering partitions a data set into subsets (called clusters), so that data in each subset share some common trait - often measured as proximity according to some defined distance measure. It is therefore natural to apply data clustering techniques to identify clusters of nodes/connections that exhibit similar network properties. However, in this paper we show that traditional clustering algorithms fail to correctly identify clusters of homogeneous nodes/connections, and therefore we propose a novel framework, called "NetCluster" to achieve this goal. Indeed, most clustering algorithms prefer a non dense measurement space, so that borders between clusters could be more easily defined. However, this assumption does not hold on the typical Internet measurement dataset. Thus, new approaches are needed, as the one proposed in this paper, which relies on a selection of a proper clustering algorithm plus the introduction of pre- and post-processing phases. Extensive tests performed both on artificial traffic and real traffic traces prove that our technique outperforms several well known clustering algorithms.

Therefore, the contribution of the paper is threefold:

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

- the identification of a clustering algorithm tailored to the analysis of typical Internet measurement datasets;
- the definition of a framework for Internet Tomography, that analyzes passive measurements and permits the identification of clusters of Internet nodes that are network-wide (geographically) close to each other;
- the analysis of network characteristics as seen at a measurement point.

II. Feature Selection

Clustering algorithms group objects (samples) that have similar characteristics in clusters, according to a notion of distance among objects and clusters in a metric space $X = \mathbb{R}^n$. Our goal is to identify clusters (groups) of nodes that, from a given vantage point, are network-wide similar. We consider as vantage point the Politecnico di Torino campus network, which includes more than 7000 hosts A passive probe sniffs all the packets flowing on the link that connects the internal LANs to the Internet via a single edge router. Therefore, the passive probe can monitor all incoming and outgoing packets, i.e., packets going to a host inside the campus LAN and coming from a host in the Internet, or vice versa.

The probe runs Tstat [8], [9], a passive monitoring tool that permits to derive network and transport layer measurements. Tstat rebuilds each TCP connection by matching incoming and outgoing segments: thus, a flow-level analysis can be performed [9]. A TCP flow is identified by snooping the signaling flags (SYN, FIN, RST), and the status of the TCP sender is rebuilt by matching sequence numbers on data segments with the corresponding acknowledgment (ACK) numbers.

Among the large set of available variables, the following indices were selected, since they contain the most useful information for network-wide localization

- the minimum IP packet Time-To-Live (TTL) observed on packets belonging to the TCP flow, i.e., the number of hops from the remote host to the vantage point[1];
- the minimum Round-Trip-Time (RTT) observed on a TCP flow, i.e., the minimum time lag between the observation of a TCP segment and of the corresponding ACK, a variable strongly related to the distance between the two hosts;
- the flow reordering probability ($P\{reord\}$), which can be useful to distinguish different paths;
- the flow dropping probability ($P\{drop\}$), that can be used to separate a low-speed noisy path from a backbone high-speed one;
- the flow duplicate probability ($P\{dup\}$), that can highlight a destination served by multiple paths.

Notice that another possible candidate index, the IP address, does not always provide a proper "distance" information. For example, two consecutive IP addresses might belong to two different ISPs operating in different locations. Furthermore, this metric does not take into account the network condition. Indeed, a key requirement is that the variables depend on link state and on node congestion, since we look for a dynamic node localization.

Thus, a sample in the metric space is defined by the tuple $\{TTL, RTT, P\{reord\}, P\{drop\}, P\{dup\}\}$. Only TCP flows which last more than $P = 100$ packets are considered, to obtain reliable estimates.

[1]The initial TTL value is set by the source, typical values being 64 and 128. TTL values are converted to the range 0-64 to normalize this variable.

III. The NetCluster Framework

We tested several types of clustering algorithms to evaluate their ability in grouping samples. While it is beyond the scope of this paper to present a detailed comparison between different clustering algorithms (details can be found in [10]), we found that most of the off-the-shelf algorithms suffer from significant errors, as we will show in the next section. The principle reasons are: the difficult in setting input parameters, since most of the algorithms require precise knowledge on density and distribution of the input data in the metric space; the tendency to create many clusters too fragmented or large meaningless subsets. Therefore, we propose a new clustering framework, called "NetCluster", that overcomes the limits of well-known algorithms when applied to network tomography. The framework includes three phases: (i) pre-processing, (ii) running the clustering algorithm and (iii) post-processing. The data pre-processing and post-processing phases and the selection of the proper clustering algorithms are all needed to correctly cluster samples in the context of Internet measurements, as shown in Sec. IV-A.

A. Pre-Processing

Traces are pre-processed to improve algorithm speed and cluster quality. In the context of Internet measurements, some dimensions of the metric space represent indices for which even a small variation corresponds to a huge distance on the sample space. For example, two samples with different TTLs should belong to distinct clusters, since, if they were sharing the same path, they would likely have the same TTL[2].

When it is hard to define a distance metric, it is appropriate to enforce an a-priori dataset partitioning so that homogeneous samples belong to the same (large) cluster. This is achieved by a pre-processing phase, in which the original dataset is split into a number of disjoint subsets, by partitioning on the values of the dimension for which a metric definition hardly holds.

Considering the metric space under analysis, we applied the pre-processing phase on the TTL dimension. Samples are partitioned on the basis of their TTL value. The sample space is then reduced to a four dimensional space, i.e., $\{RTT, P\{reord\}, P\{drop\}, P\{dup\}\}$. Since the clustering algorithm operates on a smaller subset of samples and on a $(n-1)$ dimension space, this step also reduces the complexity of the following phases.

[2]The latter consideration is correct if the measurement window is not too large, so that the Internet routes are stationary, as reported in [11]

B. Clustering algorithm

Among clustering algorithms, we choose the grid-based WaveCluster algorithm [12] due to its remarkable properties when used in the context of network traffic analysis, such as scalability, ability to remove noise and multi-resolution analysis capability. Scalability is the ability to produce an output in linear time with the number of data samples. Noise reduction is a fundamental requirement for the analysis of any measurement affected by noise, such as typical Internet measurements. This goal is obtained by the Wavelet transform. Finally, multi-resolution analysis permits to easily define the cell size independently for each variable in the metric space, i.e. to easily manage the granularity of the grid definition. A detailed comparison among clustering algorithms is available in [10]. Results shown in Sec. IV justify this choice. In the following the Traditional version (WCT, WaveCluster Traditional) is presented, while the post-processing phase is detailed in Sec. III-C.

1) Traditional WaveCluster Algorithm (WCT): The WaveCluster algorithm builds on three main steps: (i) quantization, (ii) wavelet transform, (iii) and cluster definition.

Initially, the metric space is quantized into a finite set of cells. The cell granularity is set as an input parameter. For each cell, a density value is stored, defined as the number of samples belonging to the cell. This phase reduces the number of objects to analyze, since clustering will operate on cells and not on samples.

The next step consists of applying the discrete Wavelet transform to smooth the density value of each cell according to the density values of adjacent cells. Thus, regions containing high density cells are emphasized, whereas regions containing low density cells are smoothed out.

The final step consists in cluster definition on the transformed space. A cluster is defined as the maximal set of connected cells with a non negligible density. A threshold ρ_m is defined to identify high density cells.

C. Post-processing

Finally, a post-processing phase is run to refine the cluster definition. This is a key point that makes NetCluster different from other clustering techniques. Indeed, when analyzing Internet measurement data, the classical algorithms identify either few, very large clusters, or a large number of highly fragmented clusters. This is due to the nature of the Internet measurements, in which the intrinsic variability of the measurements tends to spread-out samples. The WaveCluster algorithm creates large clusters, since a large number of cells are connected together, being the cell adjacency the criterion adopted to form clusters. To avoid this behavior, we enforce a maximum "size" for each identified cluster, e.g., to avoid that a cluster includes sample with RTTs ranging from $[0, 500]$ms.

To reach this goal, we first define, for each dimension in the metric space, a maximum radius that will be used to group cells in the same cluster. This process defines a multi-dimensional ellipsis in the metric space. Then, for each cluster obtained by the WaveCluster algorithm, the cell with the maximum frequency is identified. This cell becomes the center of the multi-dimensional ellipsis, because the largest portion of homogeneous samples is concentrated in its proximity. All cells of the current cluster which are included in the multi-dimensional ellipsis are then assigned to the newly created cluster, while cells outside the ellipsis remains in the previous cluster. This process is iterated considering all clusters.

At the end of the post-processing phase, all samples belonging to the same cluster are characterized by a distance smaller that the ellipsis "diameter".

IV. Results

A. Real Traffic from known servers

To test the effectiveness of the proposed algorithm, we first run a set of experiments in which several connections to known servers were artificially generated. We considered a set of N HTTP mirrors located in different geographical positions. For each HTTP server, we downloaded the same file for C times. Therefore, we would like to identify N clusters, each including C samples.

Tstat is used to characterize each TCP flow and to extract the features of the corresponding sample. Then, clustering algorithms are run. Identified clusters are then compared to the expected set of clusters. We repeated the experiment considering the set of $N = 59$ UBUNTU mirrors to download the distribution of "wget" and the set of $N = 25$ sourceforge mirrors to download the distribution of "visualwget". Experiments were repeated during the day and at night, to observe the impact of different network conditions.

For each selected algorithm, a set of tests were performed to find the optimal input parameter settings, using, when available, tools to optimize algorithm performance like [13]. Due to lack of space, we only report results considering the most critical scenario, i.e., the UBUNTU dataset collected during the day, and the best parameter settings for each algorithm.

Fig. 1 reports the number of clusters identified by classical clustering algorithms (DBSCAN [14], EM [15], and the traditional WaveCluster (WCT)) and by the NetCluster framework (which includes the post-processing phase) (NC, NetCluster). Measurement data were pre-processed according to the previoulsy described pre-processing phase for all algorithms. 59 clusters should be identified in this scenario. EM requires the number of clusters as an input parameter; hence, EM (obviously) identifies the correct number of clusters. DBSCAN identifies a large number of small clusters, while WCT tends to identify few, large clusters. NC identifies a number of cluster very close to the expected one (55 clusters), thanks to the post-processing phase.

Fig. 2 reports the percentage of error in the clustering composition, as determined by the different algorithms. For each server $X = 1, \ldots, N$, let $N_{OK}(X)$ be the number of flows in the cluster containing the largest number of samples from X. Let $N_{KO}(X)$ be the number of flows in X assigned

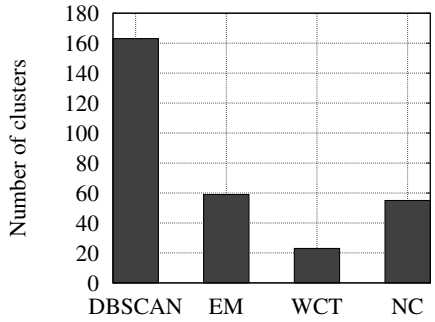

Fig. 1. Number of clusters identified considering different clustering algorithms.

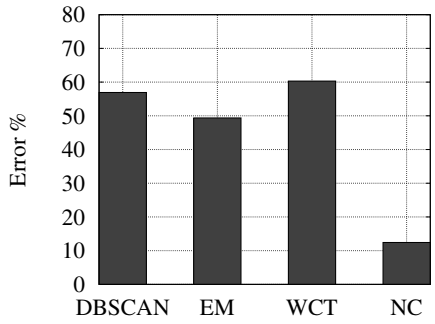

Fig. 2. Percentage of error considering different clustering algorithms.

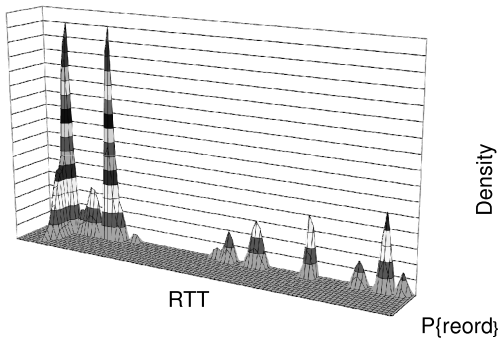

Fig. 3. Graphical representation of cell density in a two-dimensional metric space, including the round-trip time and the reordering probability.

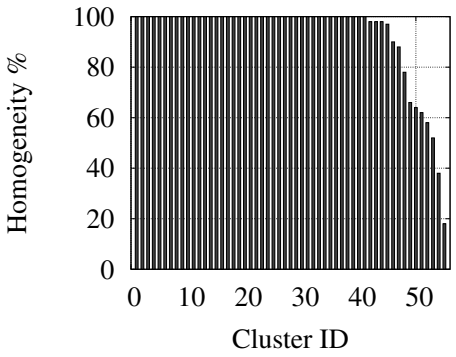

Fig. 4. Clusters homogeneity considering the NetCluster framework.

to other clusters. The percentage of error is then evaluated as

$$\eta = \frac{\sum_X N_{KO}(X)}{CN} \times 100$$

Results show that NetCluster outperform the other algorithms, showing a percentage of errors of about 10%, whereas the other algorithms range around 50%. Note that EM, even if considering exactly N clusters, shows a large percentage of error. This is due to the fact the EM does not explicitly consider noise. Thus, isolated samples may be erroneously considered as a single cluster instead of outliers/noise.

Fig. 3 shows a graphical representation of part of the sample space after the first two steps of the WaveCluster algorithms, i.e., quantization and wavelet transform. The resulting cell density versus $\{RTT, P\{reord\}\}$ is plotted for a given TTL value. Clusters referring to a given server clearly emerge. However, the group of cells with small RTT values forms a single set of connected cells, i.e., a single huge cluster. The NetCluster post-processing phase is able to split this cluster into several smaller clusters, better representing the original dataset.

To give an intuition of cluster composition, Fig. 4 plots the cluster homogeneity, defined as 100 $\times$ the ratio between the number of samples from the prevailing server versus the cluster size. Clusters are sorted in decreasing values of homogeneity. 39 clusters contain only flows from a single server, while only 3 clusters are characterized by an homogeneity smaller than 50%. Investigating further, the most heterogeneous clusters group together flows coming from mirrors that are very close to each other, e.g., a mirror in Bern and one in Losanne (CH).

All the presented results show the ability of NetCluster in identifying homogeneous clusters. NetCluster outperforms other algorithms, especially thanks to its post-processing phase, when a maximum cluster size is enforced.

B. Internet Traffic

In this section we report the analysis performed by running NetCluster on real Internet traces. We show results collected on a 24-hours trace collected on May 22nd 2007 on our Campus LAN 155Mbit/s access link. Our aim is to find the group of Internet nodes which exchange traffic with our Campus LAN. In other words, we apply the algorithms to a large data set that potentially covers the entire Internet.

Tstat was used to obtain the characterization of TCP flows. 24 sub-sets of samples were obtained, one set for each hour. Obviously, it is not possible in this scenario to evaluate the cluster definition accuracy, since we have no control on the flow destination. We therefore use some simple indicators (e.g. the cluster homogeneity) to prove the effectiveness of NC.

We show results obtained by running the NetCluster framework only, which has been proven in the previous Section to be the most reliable in finding homogeneous clusters. Classical

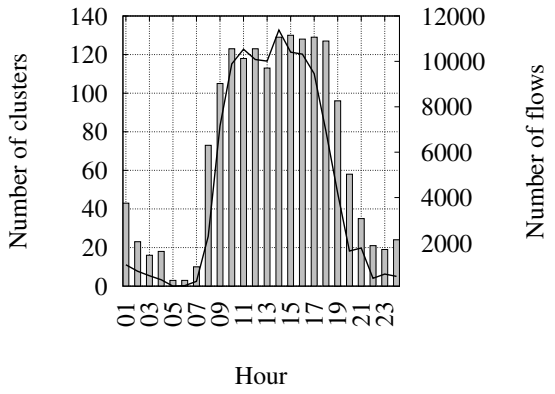

Fig. 5. Resulting number of clusters and TCP flows (solid line) in the dataset.

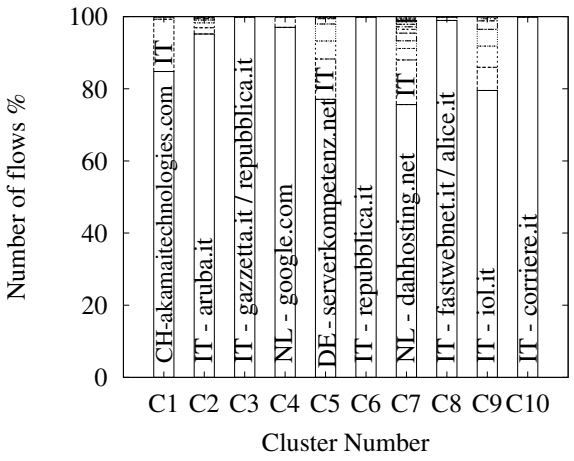

Fig. 6. Graphical representation of cluster structure considering the 10 largest clusters. The dataset considers flows active from 12:00 to 13:00.

clustering algorithms were also tested, and cluster homogeneity was much worse than the one reached by NetCluster.

Fig. 5 depicts the number of flows with more than 100 packets (solid line) and the number of cluster identified (solid bars) in each subset. As expected, the number of clusters follows a day-night trend, as would the number of TPC flows, due to the higher load offered during the day. Indeed, during peak hours from 9:00 to 18:00, the number of cluster is almost stationary, varying between 110 and 130.

To show some examples of the quality of identified clusters, Tab. I shows the IP addresses, the number of flows and the reported DNS server name of samples belonging to the largest cluster, which includes 369 flows. By looking at the IP addresses, it can be observed that 97% of the contacted servers are Google servers (belonging to different subnets), while only 11 servers are not registered by Google. However, all servers are located in Amsterdam, the Netherlands.

Fig. 6 reports the breakdown of the largest ten clusters, showing the countries to which the IP addresses belong to, and the largest ISP name, as provided by the WHOIS service. Cluster homogeneity is quite astonishing. This means that NetCluster is very effective in correctly grouping servers together.

V. Conclusions

We proposed NetCluster, a framework able to deal with a dense dataset representing Internet passive measurements. Extensive tests performed both on artificial and Internet traffic traces prove that NetCluster outperforms several well known clustering algorithms.

TABLE I

Composition of a cluster

IP Address	**Flow %**	Number of Flows	Server Name
66.249.93.X	59%	217	ug-in-fX.google.com
66.249.91.X	22%	83	ik-in-fX.google.com
64.233.183.X	16%	58	nf-in-fX.google.com
82.94.210.200	2%	7	-
194.109.217.140	0.5%	2	emo.blender.org
62.50.24.217	0.5%	2	amst2.eu.psigh.com

References

[1] Y.Yardi, "Network Tomography: estimating source-destination traffic intensities from link data", *Journal American Statistics Association*, v.91, pp.365-377, 1996.

[2] R.Caceres, N.G.Duffield, J.Horowitz, D.F.Towsley, "Multicast-based inference of network-internal loss characteristics", *IEEE Transactions on Information Theory*, v.45, n.7, pp.2462-2480, Nov. 1999.

[3] E.Brosh, G.Lubetzky-Sharon, Y.Shavitt, "Spatial-temporal analysis of passive TCP measurements", *IEEE INFOCOM 2005*, Miami, Florida, USA, March 2005.

[4] V.Aggarwal, A.Feldmann, C.Scheideler, "Can ISPs and P2P systems co-operate for improved performance?", *ACM SIGCOMM Computer Communications Review*, v.37, n.3, pp.29-40, July 2007.

[5] S.Ratnasamy, M.Handley, R.Karp, S.Shenker, "Topologically-Aware Overlay Construction and Server Selection", *IEEE INFOCOM 2002*, New York, NY, USA, June 2002.

[6] C.Barakat, W.Dabbous, M.A. Kaafar, L.Mathy, K.Salamatian, T.Turletti, "Securing internet coordinate embedding systems", *ACM SIGCOMM Conference*, Kyoto, Japan, August 2007.

[7] J.Han, M.Kamber, "Data Mining: Concepts and Techniques", *Morgan Kaufmann*, San Francisco, 2006.

[8] A.Carpani, R.Lo Cigno, M.Mellia, "Measuring IP and TCP behavior on a Edge Node with Tstat", *Computer Network*, v.47, n.1, pp.1-21, Jan 2005.

[9] M.Mellia, M.Meo, L.Muscariello, D.Rossi, "Passive Identification and Analysis of TCP Anomalies", *IEEE ICC)*, Istanbul, Turkey, June 2006.

[10] L.Chiaraviglio, E.Baralis, A.Bianco, T.Cerquitelli, M.Mellia, "NetCluster: an Incremental Clustering Algorithm for Internet Tomography", Technical Report DE-TLC-NET-April-2008-Clustering-1.

[11] J.Rexford, J.Wang, Z.Xiao, Y.Zhang, "Bgp routing stability of popular destinations," *ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[12] S.Chatterjee, G.Sheikholeslami, A.Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases", *VLDB*, New York, NY, USA, August 1998.

[13] M. Ankerst, M. Breuning, H. Kriegel, J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *ACM SIGMOD*, Philadelphia, PA, USA, June 1999.

[14] M.Ester, H.P.Kriegel, J.Sander, X.Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, Oregon, USA, August 1996.

[15] G.McLachlan and T. Krishnan, *The EM algorithm and extensions*, John Wiley and Sons, 1997.

[16] R.Agrawal, J.Gehrke, D.Gunopulos, P.Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", *ACM SIGMOD*, Seattle, Washington, USA, June 1998.