Image Data Source Selection Using Gaussian Mixture Models

Soufyane El Allali, Daniel Blank, Wolfgang Müller, and Andreas Henrich

University of Bamberg
Faculty of Information Systems and Computer Informatics
Chair of Media Informatics*
D-96045 Bamberg, Germany
soufyane.el-allali@wiai.uni-bamberg.de
http://www.uni-bamberg.de/wiai/minf

Abstract. In peer-to-peer (P2P) networks, computers with equal rights form a logical (overlay) network in order to provide a common service that lies beyond the capacity of every single participant. *Efficient similarity search* is generally recognized as a frontier in research about P2P systems. In literature, a variety of approaches exist. One of which is data source selection based approaches where peers summarize the data they contribute to the network, generating typically one summary per peer. When processing queries, these summaries are used to choose the peers (data sources) that are most likely to contribute to the query result. Only those data sources are contacted.

In this paper we use a Gaussian mixture model to generate peer summaries using the peers' local data. We compare this method to other local unsupervised clustering methods for generating peer summaries and show that a Gaussian mixture model is promising when it comes to locally generated summaries for peers without the need for a distributed summary computation that needs coordination between peers.

1 Introduction

Peer-to-peer (P2P) networks are made up of independently administered computers with equal rights (peers) that cooperate with each other. They form a logical overlay network in order to provide a common service that lies beyond the capabilities of each of its participants. P2P systems have shown their viability in a number of large-scale applications. Their success as means of large-scale data distribution in so-called file-sharing networks has motivated research in P2P data management. One focus of research is the search on stored data in a P2P network. Current research is spanning diverse areas such as efficient exact search (via distributed hash tables [19,17]), similarity search on text [20] and multimedia data [18,13], and semantic networks expressed via RDF [15].

The topic of this paper is similarity search for Information Retrieval (IR) and in particular Content-Based Image Retrieval (CBIR) in P2P networks. The basic

* This work is funded by the German Research Foundation DFG HE 2555/12-1.

approach followed by our algorithm is that of probabilistic *source selection*: As each peer holds data, each peer is a potential source of interesting data. At query time, peers that are most probably a source of interesting data are selected. This selection is done based on *peer data summaries* that are maintained by the P2P network. The selected peers are queried, then their results are merged. This general approach that stems from classical Distributed IR [11,3] stands aside from many mainstream P2P approaches that use distributed indexing structures or the ones that try to improve the link structure in order to gain performance. We feel that the Distributed IR approach is not only promising in terms of performance, but also in terms of costs that have to be paid for maintaining such a network. For an overview on CBIR in P2P networks and their adaptivity properties we refer the reader to [14].

For our source selection mechanism, our approach builds Gaussian mixture models (GMM) based on peers' indexing data. We choose GMM because they allow adaptive multimedia retrieval. For instance, relevance feedback approaches have been proposed using GMM [16]. In our setting, every model corresponds to a peer summary that is shared with the other peers in our P2P network described in section 2.1. This allows us to reduce the costs of summary creation drastically given that the peers do not need to globally coordinate the creation of their summaries. We discuss next the cost associated with such an approach:

Joining the network: The cost of joining the network that has to be paid up-front is comparatively low. Typically, the summary of a joining peer has to be replicated once or multiple times within the network. Some approaches [6,13,12,5] allow flexible load balancing for distributing the cost of joining the network over multiple peers.

One peer summary usually has about the size of a very small number of index data items. So one can afford much replication before joining a summary-based network becomes as expensive as joining a distributed indexing structure based network.

Operation: The actual performance and cost of querying in a summary-based system depends on the *quality of the summaries employed* and the *quality of the peer ranking algorithm*. In literature successful summaries have been described for IR [6,1] and CBIR [13,9], however these approaches need global coordination between the peers in order to generate the peers' summaries. A task that we avoid when using our approach.

Leaving the network: When a peer leaves, the summaries need to be expired in the network. Depending on the architecture this task can be simple. Since each peer generates its summary locally, it becomes easy for the network to expire peers as no extra work is needed for the remaining peers to generate new summaries, all they need to do is to disable the leaving peer and its summary from their known peers' list. The networks considered here can perform expiry cheaply as part of their usual maintenance protocol.

The remainder of this paper is organized as follows: In section 2 we briefly describe our P2P environment and we present the source selection approach together with a description of the data source (*i.e.* the peer) summaries. Section 3.1

explains our experimental setup: the data acquisition and the performance measure used for evaluating the experiments. Our experimental results are shown in section 3.2. Section 4 finalizes our paper with a conclusion and an outlook on future work.

2 P2P Information Retrieval: A Summary-Based Network for Single-Hop Semantic Routing

We are looking for an approach that permits efficient indexing of documents without expensive replication of full indexing data. As indexing of high-dimensional vectors is hard, we choose a single-hop semantic routing approach, also known as source selection. We will seek to use summaries that are *simple to generate*, *cheap to distribute*, and *selective* by means of successfully permitting efficient source selection.

In a summary-based P2P network for single-hop semantic routing, the query processing consists of identifying peers that probably contain relevant data (*i.e.* ranking the peers), forwarding the query to them and afterwards collecting and merging the results. The advantage of this approach is that only summaries are distributed throughout the network (as opposed to full indexing data). And as mentioned before, the query performance is determined by the quality of the summaries and the peer selection method.

While our focus is rather on summary and selection quality, we give a short overview of the underlying P2P architecture.

2.1 The P2P Environment

Our considerations are based on PlanetP [6]. In PlanetP each peer knows the summary of every other single peer participating in the network. This makes routing simple and the network extremely robust. As a downside, this approach is not scalable. Depending on the churn rate (*i.e.* the number of peers arriving and leaving per minute related to the total number of peers in the network) and the type of summaries used, PlanetP starts to fail at a couple of thousands of peers. When the number of nodes in the network and the churn rate are too high, the peers are mainly busy forwarding summaries and the network is not able to process queries anymore.

As a solution to these scalability problems, a variant of PlanetP called Rumorama [13] has been proposed. Rumorama builds hierarchies of networks that are accessible by an efficient multicast. Its leaf networks behave like PlanetP. Therefore, while we examine ranking of peers and the effect of local clustering in PlanetP-like middle sized networks, our method can immediately be utilized (as is) in Rumorama-like networks.

The original PlanetP implementation is designed for text documents and summarizes each peer by a Bloom filter [2]. Bloom filters are lossy, efficient, and compact representations of sparse sets. Unfortunately, Bloom filters are not adapted to the indexing of densely populated high-dimensional vectors [14].

Within this paper we use Gaussian mixture models to generate peer data summaries using 36-dimensional indexing data as described in sections 2.2 and 3.1. Then, the summaries are distributed in the P2P network and the querying of documents is done based on the following scheme: rank the peers for a given query based on their summaries, contact them, receive their result sets, and merge them to obtain the top-N documents retrieved.

It is worth noting that our approach can also be used in super-peer networks [22] as a type of unstructured networks. In super-peer networks, some nodes take more responsibilities (*i.e.* load) than the average peer. In classical super-peer systems such as [22], super-peers hold *all* the indexing data present in the network. Queries are processed by looking only at the super-peers. However, super-peers can also be used in a summary-based context. Each (normal) peer transfers its *summary* to one super-peer. The super-peers then process the queries by source selection using the summaries stored in them. Since the super-peers contain all summaries of the peers that are attached to them, they are able to determine which peers in their sub-network are needed to be contacted. As a consequence, super-peers would obtain the same result as a PlanetP network, with roughly the same query cost. However, other networking properties would be much different from PlanetP and out of scope here. The botton line is that *also super-peer architectures could immediately be used with summaries proposed in this paper.*

Next we describe the probabilistic approach for P2P image retrieval.

2.2 Summaries for Efficient Source Selection: A Probabilistic Retrieval View

The probabilistic view of retrieval and similarity treats the image retrieval problem as a vector classification problem. We can define a mapping from images to image classes. It has been shown that this view is very flexible and successful [21]. In our case we try to find the best mapping between query images and peers in the network. This mapping maps a query image feature vector $\boldsymbol{x}$ to the peer $peer_\alpha$ that is most likely to contain similar feature vectors to $\boldsymbol{x}$. In other words we choose the peer that maximizes the probability $p(peer_\alpha|\boldsymbol{x})$[1]. Using Bayes rule we can define this as follows:

$$
\begin{aligned}
g^*(\boldsymbol{x}) &= \underset{peer_\alpha}{\operatorname{argmax}}\Big(p(peer_\alpha|\boldsymbol{x})\Big) \\
&= \underset{peer_\alpha}{\operatorname{argmax}}\left(\frac{p(\boldsymbol{x}|peer_\alpha)p(peer_\alpha)}{p(\boldsymbol{x})}\right) \\
&= \underset{peer_\alpha}{\operatorname{argmax}}\Big(p(\boldsymbol{x}|peer_\alpha)p(peer_\alpha)\Big) \text{ since } p(\boldsymbol{x}) \text{ is constant}
\end{aligned}
$$

[1] $p(peer_\alpha|\boldsymbol{x})$ in itself means only maximizing the probability of finding $\boldsymbol{x}$ in $peer_\alpha$. However, if we assume that the probability distribution over peers is smooth, $p(peer_\alpha|\boldsymbol{x})$ also means finding the peers that most probably contain similar documents to the query $\boldsymbol{x}$.

where g is a mapping from indexing data $\boldsymbol{x} \in X$ to the peers. $p(\boldsymbol{x}|peer_\alpha)$ is the probability that $\boldsymbol{x}$ is found in $peer_\alpha$, and $p(peer_\alpha)$ is the prior probability of drawing the result for $\boldsymbol{x}$ from $peer_\alpha$ without any prior knowledge other than $peer_\alpha$'s size. $p(peer_\alpha)$ is proportional to the number of documents contained in $peer_\alpha$.

Now the goal is to find a representation for every peer in the netwok that enables the estimation of the probability to find a given data vector $\boldsymbol{x}$ within a given peer $peer_\alpha$. In order to achieve this, every peer generates its own model that we refer to as *peer model*, this model is then shared with other peers. Once a query is issued, the querying peer makes a ranking of peers to contact based on their models. The following are the criteria and the characteristics of our mechanism.

Simple to generate: Every peer generates its own model. This is done independently of other peers using the peers' local indexing data. In this paper we use a Gaussian mixture model (see section 2.3) to represent every peer and compare this model to models obtained through *k-means* and *linkage clustering*. In contrast to other methods that perform distributed clustering, for instance distributed *k-means* [8], locally generating a data model removes the costs induced by distributed clustering.

Cheap to distribute: On entering the P2P network, each peer needs to generate its model. After that, only this model's parameters need to be exchanged. The parameters' size need to be small and as such keep the distribution cheap. The cost of the summaries distribution process is proportional to the summary sizes. Section 3.2 describes the parameter sizes obtained by our approach. Furthermore, when a peer adds or removes data from its collection, all it needs to do is to generate a new Gaussian mixture model to describe its data representation and to let the other peers know its new peer model.

Selective: Based on the summaries, the peer issuing the query ranks the peers and contacts them based on this ranking. The efficiency of the ranking is summary specific. For example in the GMM case we rank the peers based on the maximum likelihood that a peer's summary/model generates for a particular query. In summary-based retrieval the query costs are proportional to the number of peers contacted.

2.3 Retrieval Using Compact Peer Descriptions

A Gaussian Mixture Approach: Every peer $peer_\alpha$ contains a set of indexing data $X_{peer_\alpha} = \{\boldsymbol{x}_n | \boldsymbol{x}_n \in \mathbb{R}^d, n \in [1, N_{peer_\alpha}]\}$, where N_{peer_α} is the number of indexing data items $peer_\alpha$ contains. The peer uses this indexing data to generate a Gaussian mixture model with parameters $\boldsymbol{\Theta}_{peer_\alpha} = \{\theta_j = (\boldsymbol{\mu}_j, \Sigma_j, p_j) | j \in [1, k_{peer_\alpha}]\}$ in order to represent its data distribution, where $\boldsymbol{\mu}_j$, Σ_j and p_j are the mean, covariance matrix and prior probability of every kernel (Gaussian) in the model. The generation of the model is initiated using a ten-fold cross validation in order to specify the number of clusters k_{peer_α}, where clusters correspond to kernels of the GMM. The probabilities p_j for each cluster $c_j \in C_{peer_\alpha}$ are

determined using *k-means* to find $k_{peer_\alpha} = |C_{peer_\alpha}|$ that generates the minimum squared error for all clusters, where C_{peer_α} is the set of all clusters generated for $peer_\alpha$. We use the Expectation-Maximization (EM) algorithm [7] in order to estimate the Gaussian mixture models' parameters. For completeness we describe below the EM algorithm and its characteristics. In the following $\mathcal{N}(\boldsymbol{\mu}, \Sigma, \boldsymbol{x})$ is the multivariate normal distribution.

$$\mathcal{N}(\boldsymbol{\mu}, \Sigma, \boldsymbol{x}) = (2\pi)^{-d/2}|\Sigma|^{-1/2} exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

EM estimates a probability distribution for each indexing data item $\boldsymbol{x}_n$ given the set $\boldsymbol{\Theta}_{peer_\alpha}$.

$$p(\boldsymbol{x}_n|\boldsymbol{\Theta}_{peer_\alpha}) = \sum_{j=1}^{k_{peer_\alpha}} p_j \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j, \boldsymbol{x}_n)$$

The EM method determines $\boldsymbol{\Theta}_{peer_\alpha}$ that maximizes the log-likelihood L such that:

$$L(\boldsymbol{\Theta}_{peer_\alpha}) = \sum_{n=1}^{N_{peer_\alpha}} ln(p(\boldsymbol{x}_n|\boldsymbol{\Theta}_{peer_\alpha})) = \sum_{n=1}^{N_{peer_\alpha}} ln\left(\sum_{j=1}^{k_{peer_\alpha}} p_j \mathcal{N}(\boldsymbol{\mu}_j, \Sigma_j, \boldsymbol{x}_n)\right)$$

$\boldsymbol{\Theta}_{peer_\alpha}$'s parameters are then computed in the following steps:

1. Choose initial parameter settings $\theta_j \ \forall j \in [1, k_{peer_\alpha}]$:
   - $\boldsymbol{\mu}_j \in X_{peer_\alpha}$ where $\boldsymbol{\mu}_j \neq \boldsymbol{\mu}_i \forall i \in [1, k_{peer_\alpha}]$
   - $\Sigma_j = (\boldsymbol{x}_n - \boldsymbol{\mu}_j)(\boldsymbol{x}_n - \boldsymbol{\mu}_j)^T$ where $\boldsymbol{x}_n$ is the nearest feature vector to $\boldsymbol{\mu}_j$
   - $p_j = \frac{|c_j|}{N_{peer_\alpha}}$
2. Repeat $\forall j \in [1, k_{peer_\alpha}]$ until likelihood convergence is reached:

   M step: maximization
   - $\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N_{peer_\alpha}} p(\theta_j|\boldsymbol{x}_n)\boldsymbol{x}_n}{\sum_{n=1}^{N_{peer_\alpha}} p(\theta_j|\boldsymbol{x}_n)}$
   - $\Sigma_j = \frac{\sum_{n=1}^{N_{peer_\alpha}} p(\theta_j|\boldsymbol{x}_n)(\boldsymbol{x}_n-\boldsymbol{\mu}_j)(\boldsymbol{x}_n-\boldsymbol{\mu}_j)^T}{\sum_{n=1}^{N_{peer_\alpha}} p(\theta_j|\boldsymbol{x}_n)}$
   - $p_j = \frac{1}{N_{peer_\alpha}} \sum_{n=1}^{N_{peer_\alpha}} p(\theta_j|\boldsymbol{x}_n)$

   E step: expectation
   - $p(\theta_j|\boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n|\theta_j^{old})\, p(\theta_j^{old})}{\sum_{i=1}^{k_{peer_\alpha}} p(\boldsymbol{x}_n|\theta_i^{old})\, p(\theta_i^{old})} = \frac{\mathcal{N}(\boldsymbol{\mu}_j^{old}, \Sigma_j^{old}, \boldsymbol{x}_n) p_j^{old}}{\sum_{i=1}^{k_{peer_\alpha}} \mathcal{N}(\boldsymbol{\mu}_i^{old}, \Sigma_i^{old}, \boldsymbol{x}_n) p_i^{old}}$

Once the GMM parameter sets $\boldsymbol{\Theta}$ are generated they are distributed in the PlanetP network in order for every $peer_\alpha$ to be able to reconstruct $peer_\beta$'s GMM given $\boldsymbol{\Theta}_{peer_\beta}$. This distribution process is taken care of by the PlanetP setup.

When a peer $peer_\alpha$ is given a query $\boldsymbol{q}$ all it needs to do is figure out which other peer's distribution in the network produces the maximum likelihood for $peer_\alpha$'s query. Hence, a ranking R of the peers is produced based on this likelihood such that:

$$p(peer_\alpha|\boldsymbol{q}) \geq p(peer_\beta|\boldsymbol{q}) \Rightarrow R(peer_\alpha) \geq R(peer_\beta)\ \forall \alpha \neq \beta$$

where $p(peer_\alpha|\boldsymbol{q})$ is the probability of a $peer_\alpha$ given a query $\boldsymbol{q}$ such that:

- $p(peer_\alpha|\boldsymbol{q}) = p(\boldsymbol{q}|peer_\alpha)p(peer_\alpha) = p\left(\boldsymbol{q}|\boldsymbol{\Theta}_{peer_\alpha}\right)p(peer_\alpha)$
- $\boldsymbol{\Theta}_{peer_\alpha}$ is the GMM parameters set corresponding to $peer_\alpha$
- $p\left(\boldsymbol{q}|\boldsymbol{\Theta}_{peer_\alpha}\right)$ is the probability of $\boldsymbol{q}$ given a model's parameter set $\boldsymbol{\Theta}_{peer_\alpha}$

Therewith, we rank $peer_\alpha$ higher than $peer_\beta$ given a query $\boldsymbol{q}$ if the query has higher likelihood to come from $peer_\alpha$'s model than from $peer_\beta$'s one.

A Linkage Clustering Approach: We compare the previous approach to *k-means*, *complete link*, *average link* and *single link* [7] clusterings for the generation of peer summaries. Every peer locally clusters its data and sends the generated centroids to the other peers as a representation for its local data. We use seven centroids to represent a peer's data summary, this is a higher number than the average number of clusters in the GMM (see section 3.2), giving advantage to the linkage clustering approach with respect to the amount of summary data that can be shipped. Once a peer $peer_\alpha$ is given a query $\boldsymbol{q}$, it determines a ranking R of the peers to contact based on the closest centroid to the query using the distance $dist_{euclid}$.

$$dist_{euclid}(\boldsymbol{q}, C_{peer_\alpha}) \leq dist_{euclid}(\boldsymbol{q}, C_{peer_\beta}) \Rightarrow R(peer_\alpha) \geq R(peer_\beta)\ \forall \alpha \neq \beta$$

where $C_{peer_\alpha} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, ...\boldsymbol{c}_k\}$ is the set of centroids of $peer_\alpha$'s clusters and $dist_{euclid}(\boldsymbol{q}, C_{peer_\alpha})$ computes the minimum Euclidean distance of the query to all centroids $\boldsymbol{c}_i \in C_{peer_\alpha}$.

3 Experiments

3.1 Experimental Setup

In the following we describe the data that our experiments are based on as well as the measure that is used for evaluating retrieval performance.

Data Acquisition: In our experimental setting we use a real world data set. The test data is a subset from a crawl of Flickr.com[2]. Flickr.com is a web-based

[2] Flickr, *http://www.flickr.com*

community portal to store, share and search images. Each user may store an arbitrary number of photographs and other pictures in his/her account. We take a randomly chosen subset of 2,623 user accounts so that the total number of images in these accounts is 50,000. We assign each user account to one peer to simulate Flickr.com in a P2P setting.

The feature set used to index the images is a 36-bin color histogram as used in [23,10] where the HSV color space is quantized into 36 intervals. The hue component is split into seven colors that correspond to the seven Chinese colors. The saturation/value components are split into six regions where for values of $V \leq 0{,}2$ only one bin is reserved for V independently from the S and H values. This results in $7 \cdot 5 + 1 = 36$ bins. Eight of the 36 colors are gray tones making the quantization suitable for both color and gray images. Every image is therefore represented as a single 36-dimensional real-valued vector.

Defining a Performance Measure: The main performance measure used in our experiments is the *fraction of peers contacted* on average to find a fraction out of the top-20 matches for a query. We concentrate on this measure since the processing cost for obtaining a GMM is not much of a problem for reasonable peer collection sizes. The top-20 matches are computed based on the global document collection. This is done prior to executing the query in the P2P network and can be seen as a baseline against which to compare the P2P retrieval system. We choose this measure since contacting other peers during query processing, sending the query and receiving the result sets of the other peers are the main query cost factors in our P2P information retrieval scheme. In our experiments the performance measure is averaged over 100 queries in order to minimize the influence of outliers on the results.

3.2 Empirical Results

Summary Sizes: The EM algorithm computes one GMM parameter set $\mathbf{\Theta}_{peer_\alpha}$ per $peer_\alpha$, as described in section 2.3. Each GMM approximates the true distribution of data points as a sum of k d-variate Gaussians with diagonal covariance matrices. The correlation of data points is expressed by the fact that the GMM is a mixture of *multiple* Gaussians. *Locally* the dimensions of the data points are independent of each other. Because of this independence assumption, Σ is a diagonal matrix *i.e.* the number of nonzero entries is equal to the dimension d of the indexing data. Otherwise there would be d^2 nonzero entries. This means that given d as the dimension of the indexing data (*i.e.* the feature vectors), the size of a single Gaussian θ_j is:

$$\begin{aligned} size(\theta_j) &= size(\boldsymbol{\mu}_j) + size(\Sigma_j) + size(p_j) \\ &= d + d + 1 \end{aligned}$$

where $d = 36$ in our case since we use a 36-dimensional indexing vector, and the covariance matrix can be reconstructed using its diagonal that is of size d. Therefore, the total summary size of a peer $peer_\alpha$ is $size(\mathbf{\Theta}_{peer_\alpha}) = (2d + 1)k_{peer_\alpha}$,

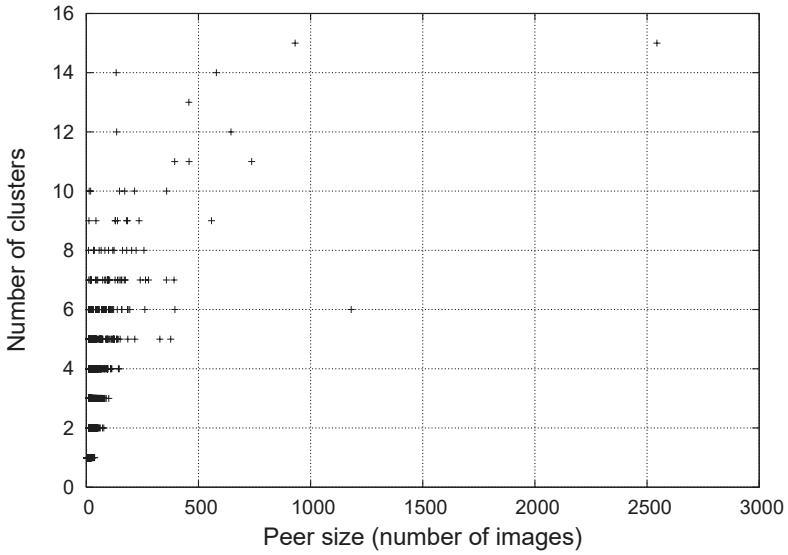

Fig. 1. Number of clusters k per peer vs. peer sizes. The average number of clusters per peer is 2. There are 2,623 points in the plot: one for every peer.

where k_{peer_α} is the number of clusters determined by cross validation. Hence, an important question arises: what is the number of clusters k_{peer_α} determined per peer in our P2P network?

Since we use cross validation to determine the number of clusters k_{peer_α}, it becomes necessary to have it constrained to a small range given that a big number is not applicable for data transfer between peers in a P2P network. In particular peer sizes can range from small sized peers with only few documents to very big peers with many documents. In our P2P network the peer size ranges from one document to 2,544 documents per peer. We plot the number of clusters determined per peer vs. the size of the peers (*i.e.* # of documents per peer) in figure 1.

From figure 1 we see that the maximum number of clusters determined for a peer is 15, and the average number of clusters is approximately 2 clusters per peer. This is a summary size that can easily be transfered in the network as part of its setup procedure. If we compare it to the local clustering techniques that distribute $\#centroids \times d \times \#bytes/float = 7 \times 36 \times 4 = 1008$ bytes per peer as summaries, we only distribute $size(\mathbf{\Theta}_{peer_\alpha}) \times \#bytes/float = (2 \times 36 + 1) \times 2 \times 4 = 584$ bytes per peer on average while achieving a better retrieval performance as will be shown in the next section 3.2.

Retrieval Cost: Table 1 shows the effect of the GMM on the retrieval. We see that the fraction of peers contacted (10.5%) is relatively low when we need to retrieve 80% of the top-20 documents. And for the retrieval of all of the top-20 documents an increase of 113% is observed to reach 22.4% of the peers on average.

We compare in figure 2 the effect of the different approaches discussed in section 2. From this figure we see that the GMM performs better than the other

Table 1. Fraction of peers contacted to retrieve a fraction of the top-20 documents

top-20 [in %]	fraction of peers [in %]
10	0.4
20	1.1
30	1.8
40	3.0
50	4.0
60	5.6
70	7.8
80	10.5
90	14.5
100	22.4

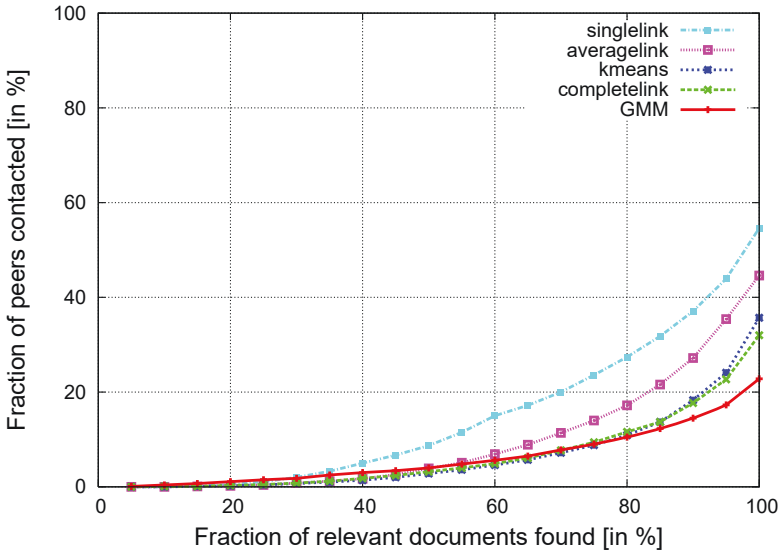

Fig. 2. Fraction of peers contacted vs. documents retrieved for *GMM*, *k-means*, *complete link*, *average link* and *single link*. Lower curves indicate lower retrieval cost for same retrieval performance.

clustering methods. Namely, in order to retrieve 100% of the top-20 documents the *GMM* approach contacts 22.4% of the peers whereas *single link*, *average link*, *k-means*, and *complete link* need to contact 54.6%, 44.6%, 35.7%, and 32% of the peers respectively in order to retrieve the top-20 documents. It is worth noting that *k-means*, *complete link* and the *GMM* approach have comparatively almost the same retrieval cost when retrieving 80% of the top-20 documents, however *GMM* starts to perform better than *k-means* and *complete link* as we retrieve all of the top-20 documents.

4 Conclusion

We have presented an approach to approximate peers' indexing data as Gaussian mixture models. In contrast to previous approaches [13,9], where for example

distributed *k-means* is used to cluster the peers' indexing data, requiring thereof coordination between the peers, our new approach does not need coordination between peers. This approach has the advantage of representing the peers (*i.e.* data sources) using small and compact descriptions: *peer models*, which have little distribution cost in the network. These *peer models* allow a querying peer to determine the best peers to contact for its query in a probabilistic manner. We experimentally compared the Gaussian mixture approach to other unsupervised clustering algorithms and showed that on real-world data the GMM approach performs better than clustering-based approaches that also do not need coordination between peers.

For future work, our approach provides a basis where adaptive multimedia retrieval can be utilized, we can use GMM-based *relevance feedback* [21,16] in the context of CBIR for P2P networks.

We also see the approach presented here suitable as a basis for investigating the performance of more elaborate techniques, namely *meta-learning methods* (*e.g.* [4]). Here, the main idea is to have each PlanetP-peer combine the knowledge present in the summaries it holds in order to improve the peer ranking.

References

1. Bender, M., Michel, S., Triantafillou, P., Weikum, G., Zimmer, C.: Minerva: collaborative P2P search. In: VLDB 2005: Proc. of the 31st Intl. Conf. on Very large data bases. VLDB Endowment, pp. 1263–1266 (2005)
2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 13(7) (1970)
3. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: Proc. 18th ACM SIGIR, Seattle, Washington (1995)
4. Chan, P.K.-W.: An extensible meta-learning approach for scalable and accurate inductive learning. PhD thesis, Sponsor-Salvatore J. Stolfo (1996)
5. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009. Springer, Heidelberg (2001)
6. Cuenca-Acuna, F.M., Nguyen, T.: Text-based content search and retrieval in ad hoc P2P communities. Technical Report DCS-TR-483, Department for Computer Science, Rutgers University (2002)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley-Interscience (2001)
8. Eisenhardt, M., Müller, W., Henrich, A.: Classifying documents by distributed P2P clustering, 286–291 (2003)
9. Eisenhardt, M., Müller, W., Henrich, A., Blank, D., El Allali, S.: Clustering-based source selection for efficient image retrieval in peer-to-peer networks. In: IEEE MIPR 2007, pp. 823–830 (2006)
10. El Allali, S., Blank, D., Eisenhardt, M., Henrich, A., Müller, W.: Untersuchung des Einflusses verschiedener Bild-Features und Distanzmaße im inhaltsbasierten P2P Information Retrieval. In: BTW 2007, 12th GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (2007)
11. Gravano, L., García-Molina, H., Tomasic, A.: Gloss: text-source discovery over the internet. ACM Trans. Database Syst. 24(2), 229–264 (1999)

12. Kronfol, A.Z.: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine. Final Thesis, Princeton (May 2002), http://www.searchlore.org/library/kronfol final thesis.pdf
13. Müller, W., Eisenhardt, M., Henrich, A.: Scalable summary based retrieval in P2P networks. In: CIKM 2005: Proc. of the 14th ACM Intl. Conf. on Information and knowledge management, pp. 586–593. ACM Press, New York (2005)
14. Müller, W., Henrich, A., Eisenhardt, M.: Aspects of adaptivity in P2P information retrieval. In: The 4th International Workshop on Adaptive Multimedia Retrieval AMR 2006 (2006)
15. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In: Proc. of the Intl. World Wide Web Conf. (2003)
16. Qian, F., Li, M., Zhang, L., Zhang, H.-J., Zhang, B.: Gaussian mixture model for relevance feedback in image retrieval. In: IEEE International Conference on Multimedia and Expo, 2002. ICME 2002 (2002)
17. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A scalable content-addressable network. In: Proc. 2001 Conf. on applications, technologies, architectures, and protocols for computer communications, San Diego, CA, United States (2001)
18. Sahin, O.D., Gulbeden, A., Emekci, F., Agrawal, D., Abbadi, A.E.: PRISM: indexing multi-dimensional data in P2P networks using reference vectors. In: Proc. of the 13th annual ACM Intl. Conf. on Multimedia, pp. 946–955. ACM Press, New York (2005)
19. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable Peer-To-Peer lookup service for internet applications. In: Proc. ACM SIGCOMM Conf., San Diego, CA, USA (2001)
20. Tang, C., Xu, Z., Mahalingam, M.: pSearch: Information retrieval in structured overlays. In: First Workshop on Hot Topics in Networks (HotNets-I). Princeton, NJ (2002)
21. Vasconcelos, N.: Bayesian Models for Visual Information Retrieval. PhD thesis, MIT (June 2000)
22. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: IEEE Intl. Conf. on Data Engineering (2003)
23. Zhang, L., Lin, F., Zhang, B.: A cbir method based on color-spatial feature. In: IEEE Region 10 Annual International Conference 1999, pp. 166–169 (1999)