Semantic Based Clustering for Image Retrieval

[1]Prof. Suchita Walke, [2]Ms. Mohini Sarode
[1]Head of Computer Department, [2]PG Student
[1]HOD (Computer) YTCEM, Mumbai, India
[2]Student of YTGOIFOE, Mumbai, India

***Abstract*— Existing web image search engines such as Google and AltaVista return a large quantity of search results, ranked by their relevance to the given query. Web users have to go through the list and look for the desired ones. This is a time consuming task since the returned results always contain multiple topics and these topics are mixed together. Things become even worse when one topic is overwhelming but it is not what the user desires.**

We consider the problem of clustering Web image search results. Generally, the image search results returned by an image search engine contain multiple topics. Organizing the results into different semantic clusters facilitates users' browsing. *(Abstract)*

***Index Terms*— About Click-through logs, image-search goals, semi-supervised clustering, spectral clustering, Feedback sessions** *(key words)*

I. Introduction

In web search applications, users submit queries (i.e., some keywords) to search engines to represent their search goals[1] However, in many cases, queries may not exactly represent what they want since the keywords may be polysemous or cover a broad topic and users tend to formulate short queries rather than to take the trouble of constructing long and carefully stated ones. Let us take a look at a simple example. Figure 1 shows the image search results of the query of "Pluto". Note that, the query used in this example is a hot query in image search according to the statistical result of Google image search engine.

Clearly, the search results of both image search engines contain two different topics: Pluto in the solar system and the dog named "Pluto" in Disney world. For this query, it is difficult to say which search engine performs better since we do not know what the user is really looking for. In fact, all these results are related to the query. However, in different situations, the results of Pluto about solar system may be noise to the user who is looking for dog Pluto[2]. A possible solution to this problem is to cluster search results into different groups with different topics. Many works have been done on web text search. We consider the problem of clustering image search.

In web image search, a good organization of the search results is as important as the search accuracy.

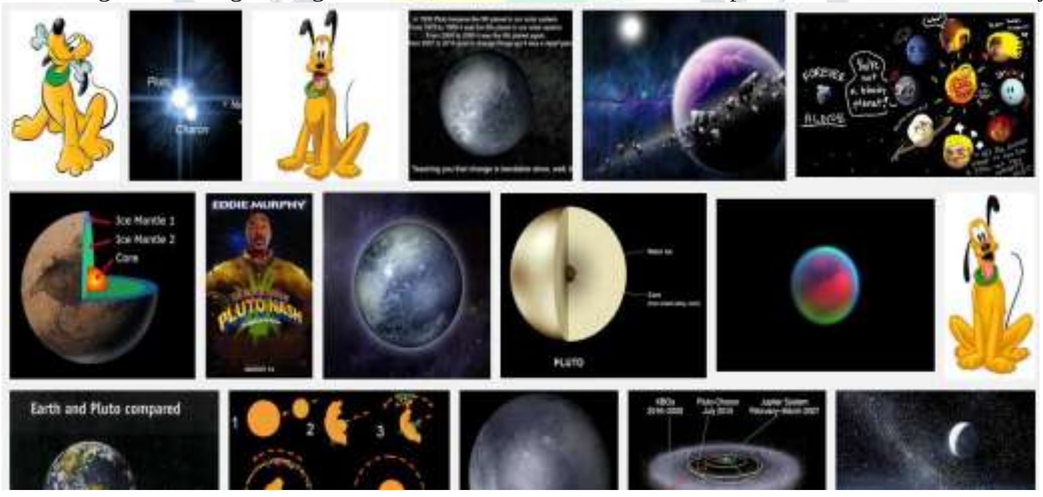

Fig1.1.Query search "Pluto"

In traditional Content-Based Image Retrieval (CBIR) area, image clustering techniques are often used to design a convenient user interface, which helps to make more meaningful representations of search results. However, as the images were usually represented by the low level visual features, it is hard to get good clustering result from semantic perspective. WWW images have a lot of properties which are quite different from those images in small database such as Corel images and family album. Web images are associated with text and link information. In this paper, based on the Vision-based Page Segmentation (VIPS), we consider the image and the block containing that image as a whole.

Users have different search goals for the same query due to the following three reasons -

• Multi-concepts: a keyword may represent different things. For example, besides being a kind of fruit, "apple" is endowed with new concepts by Apple, Inc.

• Multi-forms: the same thing may have different forms. Take "Bumblebee" in the film Transformers as an example. It has two modes: car mode and humanoid mode. These two modes are the two forms of "Bumblebee."

• Multi-representations: in image search, the same thing can be represented from different angles of view such as the query leaf. It can be represented in a real scene or by a close-up.

Query	Different user image-search goals		
1. apple			
2. Bum-blebee			
3. leaf			

Fig.2. Users have different search goals

Inferring user search goals is very important in improving search-engine relevance and user experience. Normally, the captured user image-search goals can be utilized in many applications. For example, we can take user image-search goals as visual query suggestions to help users reformulate their queries during image search. Besides, we can also categorize search results for image search according to the inferred user image-search goals to make it easier for users to browse.

However, although there has been much research for text search, few methods were proposed to infer user search goals in image search. Some works try to discover user image-search goals based on textual information (e.g., external texts including the file name of the image file, the URL of the image, the title of the web page that contains that image and the surrounding texts in image search results and the tags given by users). However, since external texts are not always reliable (i.e., not guaranteed to precisely describe the image contents) and tags are not always available (i.e., the images may not have corresponding tags that need to be intentionally created by users), these textual information based methods still have limitations.

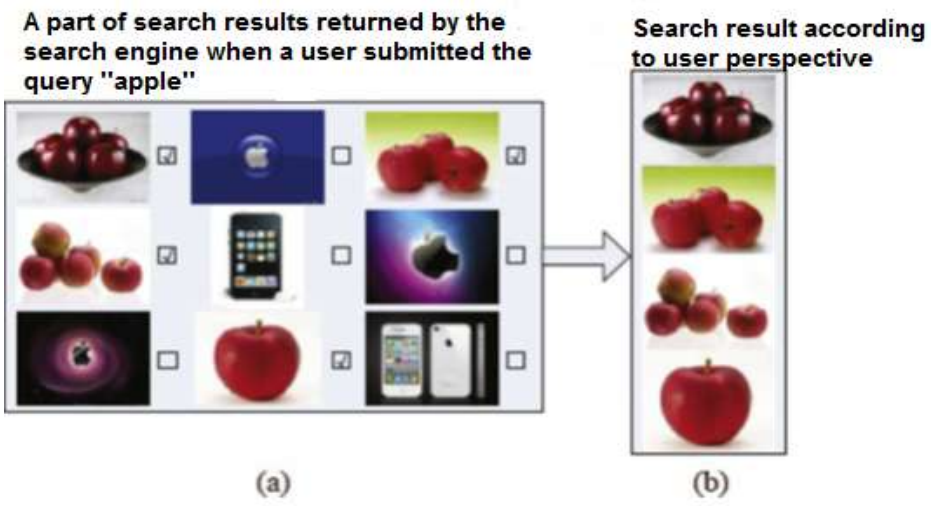
A part of search results returned by the search engine when a user submitted the query "apple"
Search result according to user perspective
(a)
(b)

Fig.3 Search result according to user perspective

It should be possible to infer user image-search goals with the visual information of images (i.e., image features) since different image-search goals usually have particular visual patterns to be distinguished from each other. However, since there are semantic gaps between the existing image features and the image semantics, inferring user image-search goals by visual information is still a big challenge.

As per our study user should have following result on search of the query "Apple". So we have to study regarding the clustering of the images which helps to retrieval of the images according to user perspective.

II. LITERATURE REVIEW

In recent years, the research on inferring user goals or intents for text search has received much attention. Many early researches define user intents as navigational and informational, or by some specific predefined aspects, such as product intent and job intent. Submit your manuscript electronically for review.

Content-Based Image Retrieval (CBIR) -

Content-Based Image Retrieval (CBIR) is defined as a process that searches and retrieves images from a large database on the basis of automatically-derived features such as colour, texture and shape. The techniques, tools and algorithms that are used in CBIR, originate from many fields such as statistics, pattern recognition, signal processing, and computer vision. It is a field of research that is attracting professionals from different industries like crime prevention, medicine, architecture, fashion and publishing. The volume of digital images produced in these areas has increased dramatically over the past 10 decades and the World Wide Web plays a vital role in this upsurge.

Several companies are maintaining large image databases, where the requirement is to have a technique that can search and retrieve images in a manner that is both time efficient and accurate (Xiaoling, 2009).In order to meet these requirements, all the solutions, in general, perform the retrieval process in two steps. The first step is the 'feature extraction' step, which identifies unique signatures, termed as feature vector, for every image based on its pixel values. The feature vector has the characteristics that describe the contents of an image. Visual features Such as colour, texture and shape are more commonly used in this step. The classification step matches the features extracted from a query image with the features of the database images and groups images according to their similarity. Out of the two steps, the extraction of features is considered most critical because the particular features made available for discrimination directly influence the efficacy of the classification task.

Traditional image search and clustering techniques are content based. They are usually based on small and static (compared to the Internet) image databases, like family albums. It is still a hard problem to learn the semantic meaning of an image from low level visual features. This makes traditional image retrieval techniques not directly applicable to web image search and organization. Although there exists some systems using traditional CBIR techniques in WWW image search. They all have the problems of scalability and performance. Almost all the commercial image search engines use the text extracted from HTML pages to index the images. In such cases, the web image search problem is converted to a text search problem.

Traditional text retrieval techniques, such as inverted indexing, TF-IDF weighting and cosine similarity measure, etc. can be used for comparing the images to the query keywords. Hyperlink is another kind of information useful for image search and organization in web context. Recently some researchers have used link information to improve image search and image clustering. So far the search problem is the primary focus of research. However, the problem of how to organize the search results is of the same importance.

Tagging –

Some works focus on tagging queries with more hierarchical predefined concepts to improve feature representation of queries. These applications belong to query classification. User search goals and the number of them should be arbitrary and not predefined. The click session information is not fully utilized.

Image clustering using similarity graphs -

There has been some research on image clustering with different types of information. The first use textual and link information to cluster the images in web pages, and then they use visual information to further cluster the images in each cluster. They consider that a single web page often contains multiple semantics and the blocks in a page containing different semantics (instead of pages) should be regarded as information units to be analyzed. They define link information as the relationships between page, block, and image. However, when we cluster the images for a query to infer user goals, there are no such blocks or link information.

Queries to reduce the semantic gap -

They define the semantic similarity graph as an undirected bipartite graph, whose edges connect a set of relative queries and the clicked images for these queries. However, if the set of queries are irrelative, there may be few or no images shared by multiple queries (e.g., the users submitting the different queries do not click the same image). In this case, the queries and their clicked images in the bipartite graph are independent and the semantic similarity graph cannot provide any semantic information. This situation often happens if we randomly select a small set of queries from query logs (i.e., do not purposely select the specific relative queries).

III. Framework of Our Approach

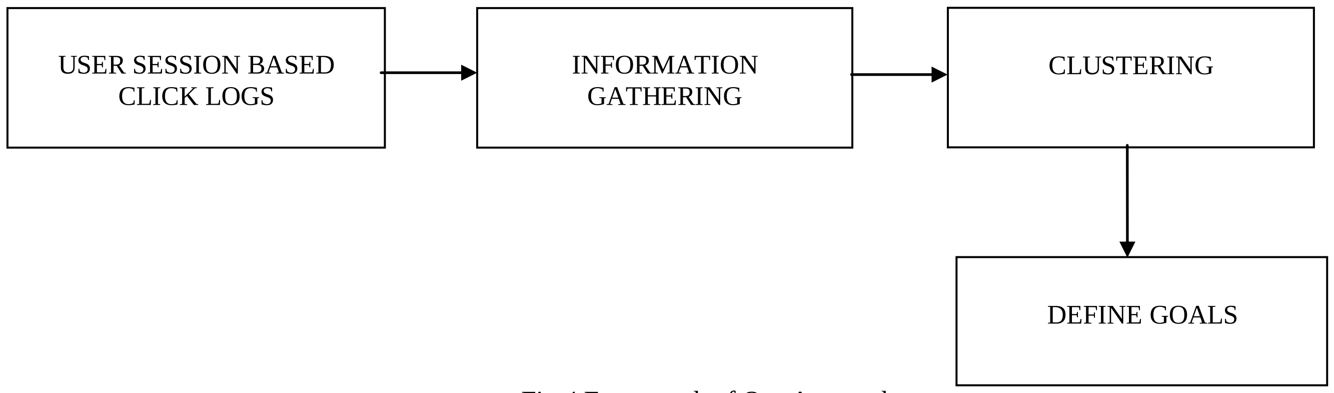

Fig.4 Framework of Our Approach

Step 1-

We first extract the visual information of the clicked images from user click-through logs. Normally, the images clicked by the users with the same search goal should have some common visual patterns, while the images clicked by the users with different search goals should have different visual patterns to be distinguished from each other. For example, for the query apple, there must be some visual patterns to distinguish fruit apples from phones.

- Extract the click session information from user click-through logs.
- Density based denoising to cluster number of the points in the neighbourhood of it.

Step 2 -

Image visual information is combined with click session information for further clustering by one of the two proposed strategies, named edge-reconstruction-based strategy and goal-image-based strategy. It should be noted that these two strategies are alternatives by using different ways to model the clicked images for a query with similarity graph.

Step 3-

Introduce spectral clustering algorithm to cluster the image graph that contains both image visual information and click session information. Spectral clustering is introduced in this step because clusters representing different user goals may have arbitrary shapes in visual feature space when clustering.

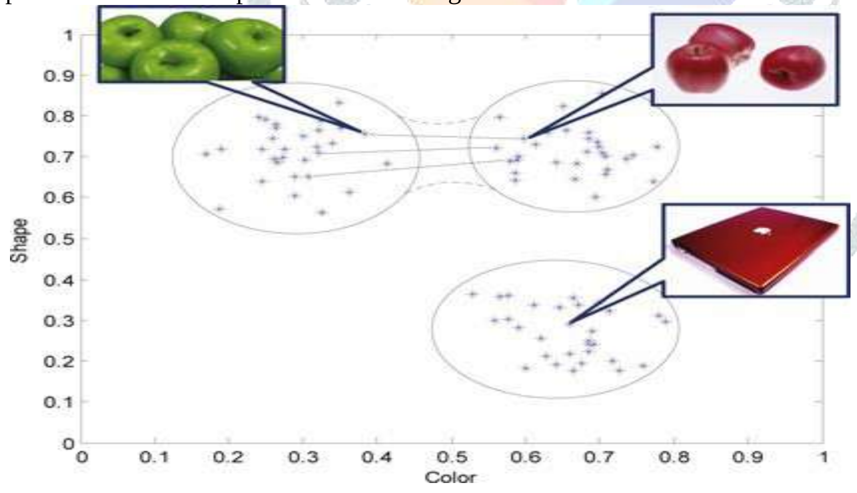

Fig.5 Clustering

Step 4 -

We first set the number of clusters k to be several probable values. Then we evaluate the clustering performances for each value of k according to the CR-based evaluation criterion. Finally, we choose the optimal value of k to be the number of user search goals.

Clustering techniques

There are many clustering methods available and each of them may give a different grouping of datasets.

Partitioning method-

Data Partitioning Algorithms, which divide data into several subsets. Because checking all possible subset systems is computationally infeasible, certain greedy heuristics are used in the form of iterative optimization. Specifically, this means different relocation schemes that iteratively reassign points between the k clusters. Unlike traditional hierarchical methods, in which clusters are not revisited after being constructed, relocation algorithms gradually improve clusters. With appropriate data, this results in high quality clusters. One approach to data partitioning is to take a conceptual point of view that identifies the cluster with a certain model whose unknown parameters have to be found. More specifically, probabilistic models assume that the data comes from a mixture of several populations whose distributions and priors we want to find. Corresponding algorithms are described in the sub-section Probabilistic Clustering. One clear advantage of probabilistic methods is the interpretability of the constructed clusters. Having concise cluster representation also allows inexpensive computation of intra-clusters measures of fit that give rise to a global objective function. Another approach starts with the definition of objective function depending on a partition.

As we have seen (sub-section Linkage Metrics), pair-wise distances or similarities can be used to compute measures of inter- and intra-cluster relations. In iterative improvements such pair-wise computations would be too expensive. Using unique cluster representatives resolves the problem: now computation of objective function becomes linear in N (and in a number of clusters).Depending on how representatives are constructed, iterative optimization partitioning algorithms are subdivided into k-medoids and k-means methods.

K-medoids is the most appropriate data point within a cluster that represents it. Representation by k-medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In k-means case a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier. On the other hand, centroids have the advantage of clear geometric and statistical meaning. The corresponding algorithms are reviewed in the sub-sections K Medoids Methods and K-Means Methods.

K-Medoids Methods-

In k-medoids methods a cluster is represented by one of its points. We have already mentioned that this is an easy solution since it covers any attribute types and that medoids have embedded resistance against outliers since peripheral cluster points do not affect them. When medoids are selected, clusters are defined as subsets of points close to respective medoids, and the objective function is defined as the averaged distance or another dissimilarity measure between a point and its medoid. Two early versions of k-medoid methods are the algorithm PAM (Partitioning Around Medoids) and the algorithm CLARA (Clustering Large Applications) [Kaufman & Rousseeuw1990].

PAM is iterative optimization that combines relocation of points between perspective clusters with re-nominating the points as potential medoids. The guiding principle for the process is the effect on an objective function, which, obviously, is a costly strategy.

CLARA uses several (five) samples, each with 40+2k points, which are each subjected to PAM. The whole dataset is assigned to resulting medoids, the objective function is computed, and the best system of medoids is retained. Further progress is associated with Ng & Han [1994] who introduced the algorithm CLARANS (Clustering Large Applications based upon Randomized Search) in the context of clustering in spatial databases. Authors considered a graph whose nodes are the sets of k medoids and an edge connects two nodes if they differ by exactly one medoid. While CLARA compares very few neighbours corresponding to a fixed small sample, CLARANS uses random search to generate neighbours by starting with an arbitrary node and randomly checking max neighbour neighbours. If a neighbour represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until num local minima are found (value num local=2 is recommended). The best node (set of medoids) is returned for the formation of a resulting partition. The complexity of CLARANS is O(N*N) in terms of number of points. Ester et al. [1995] extended CLARANS to spatial VLDB. They used R*-trees [Beckmann 1990] to relax the original requirement that all the data resides in core memory, which allowed focusing exploration on the relevant part of the database that resides at a branch of the whole data tree.

K-Means Methods-

The k-means algorithm [Hartigan 1975; Hartigan & Wong 1979] is by far the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of k clusters Cj by the mean (or weighted average) cj of its points, the so-called centroid. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function. For example, the L2-norm based objective function, the sum of the squares of errors between the points and the corresponding centroids, is equal to the total intra-cluster variance

$$E(C) = \bullet \underset{j=1:k}{\bullet} \bullet \underset{x_i \in C_j}{\bullet} \left\| x_i - c_j \right\|^2 .$$

The sum of the squares of errors can be rationalized as (a negative of) log-likelihood for normally distributed mixture model and is widely used in statistics (SSE). Therefore, k means algorithm can be derived from general probabilistic framework (see sub-section Probabilistic Clustering) [Mitchell 1997]. Note that only means are estimated. A simple modification would normalize individual errors by cluster radii (cluster standard deviation), which makes a lot of sense when clusters have different dispersions. An objective function based on L2-norm has many unique algebraic properties. For example, it coincides with pair-wise errors and with the difference between the total data variance and the inter-cluster variance.

$$E'(C) = {}^{1}\!/_{2} \bullet {}_{j=1:k} \bullet {}_{x_i, y_i \in C_j} \left\| x_i - y_i \right\|^2$$

Therefore, the cluster separation is achieved simultaneously with the cluster tightness. Two versions of k-means iterative optimization are known. The first version is similar to EM algorithm and consists of two-step major iterations that (1) reassign all the points to their nearest centroids, and (2) re compute centroids of newly assembled groups. Iterations continue until a stopping criterion is achieved (for example, no reassignments happen). This version is known as Forgy's algorithm [Forgy 1965] and has many advantages:

Advantages:-

· It easily works with any Lp–norm

· It allows straightforward parallelization

· It is insensitive with respect to data ordering.

Hierarchical method-

Hierarchical clustering builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom-up) and divisive (top-down) [Jain & Dubes 1988; Kaufman & Rousseeuw 1990]. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. Advantages of hierarchical clustering include:

· Embedded flexibility regarding the level of granularity

· Ease of handling of any forms of similarity or distance

· Consequently, applicability to any attribute types

Disadvantages of hierarchical clustering are related to:

· Vagueness of termination criteria

· The fact that most hierarchical algorithms do not revisit once constructed (intermediate) clusters with the purpose of their improvement. The classic approaches to hierarchical clustering are presented in the sub-section Linkage Metrics. Hierarchical clustering based on linkage metrics results in clusters of proper(convex) shapes. Active contemporary efforts to build cluster systems that incorporate our intuitive concept of clusters as connected components of arbitrary shape, including the algorithms CURE and CHAMELEON, are surveyed in the sub-section.

Divisive techniques based on binary taxonomies are presented in the sub-section Binary Divisive Partitioning. The sub-section Other Developments contains information related to incremental learning, model-based clustering, and cluster refinement. In hierarchical clustering our regular point-by-attribute data representation is sometimes of secondary importance. Instead, hierarchical clustering frequently deals with the N X N matrix of distances (dissimilarities) or similarities between training points. It is sometimes called connectivity matrix. Linkage metrics are constructed (see below) from elements of this matrix. The requirement of keeping such a large matrix in memory is unrealistic. To relax this limitation different devices are used to introduce into the connectivity matrix some sparsity. This can be done by omitting entries smaller than a certain threshold, by using only a certain subset of data representatives, or by keeping with each point only a certain number of its nearest neighbours. For example, nearest neighbour chains have decisive impact on memory consumption.

A sparse matrix can be further used to represent intuitive concepts of closeness and connectivity. Notice that the way we process original (dis)similarity matrix and construct a linkage metric reflects our a priori ideas about the data model. With the (sparsified) connectivity matrix we can associate the connectivity graph G = (X, E) whose vertices X are data points, and edges E and their weights are pairs of points and the corresponding positive matrix entries. This establishes a connection between hierarchical clustering and graph partitioning.

One of the most striking developments in hierarchical clustering is the algorithm BIRCH. Since scalability is the major achievement of this blend strategy, this algorithm is discussed in the section Scalable VLDB Extensions. However, data squashing used by BIRCH to achieve scalability, has independent importance. Hierarchical clustering of large datasets can be very sub-optimal, even if data fits in memory. Compressing data may improve performance of hierarchical algorithms.

Density-based method-

Crucial concepts of this section are density and connectivity both measured in terms of local distribution of nearest neighbours. The algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) [Ester et al. 1996] targeting low-dimensional spatial data is the major representative in this category. Two input parameters and Min Pts are used to define:

· An -neighbourhood

· A core object (a point with a neighbourhood consisting of more than Min Pts points)

· A concept of a point y density-reachable from a core object x (a finite sequence of core objects between x and y exists such that each next belongs to an - neighbourhood of its predecessor)

· A density-connectivity of two points x, y (they should be density-reachable from a common core object)

So defined density-connectivity is a symmetric relation and all the points reachable from core objects can be factorized into maximal connected components serving as clusters. The points that are not connected to any core point are declared to be outliers (they are not covered by any cluster). The non-core points inside a cluster represent its boundary. Finally, core objects are internal points. Processing is independent of data ordering. So far, nothing requires any limitations on the dimension or attribute types. Obviously, an effective computing of -neighbourhoods presents a problem.

However, in the case of low dimensional spatial data, different effective indexation schemes exist (meaning O(log(N)) rather than O(N) fetches per search). DBSCAN relies on R*-tree indexation [Kriegel et al. 1990]. Therefore, on low-dimensional spatial data theoretical complexity of DBSCAN is O(Nlog(N)). Experiments confirm slight super-linear runtime. Notice that DBSCAN relies on neighbourhoods and on frequency count within such neighbourhoods to define a concept of a core object. Many spatial databases contain extended objects such as polygons instead of points. Any reflexive and symmetric predicate (for example, two polygons have a non-empty intersection) suffice to define a neighbourhood".

Additional measures (as intensity of a point) can be used instead of a simple count as well. These two generalizations lead to the algorithm GDBSCAN which uses the same two parameters as algorithm DBSCAN. With regard to these two parameters and MinPts, there is no straightforward way to fit them to data. Moreover, different parts of data could require different parameters – the problem discussed earlier in conjunction with CHAMELEON. The algorithm OPTICS (Ordering Points To Identify the Clustering Structure) adjusts DBSCAN to this challenge. It builds an augmented ordering of data which is consistent with DBSCAN, but goes a step further: keeping the same two parameters, Min Pts, OPTICS covers a spectrum of all different .The constructed ordering can be used automatically or interactively. With each point, OPTICS stores only two additional fields, the so-called core- and reachability-distances.

For example, the core-distance is the distance to MinPts' nearest neighbour when it does not exceeds or undefined otherwise. Experimentally, OPTICS exhibits runtime roughly equal to 1.6 of DBSCAN runtime.While OPTICS can be considered as a DBSCAN extension in direction of different local densities ,a more mathematically sound approach is to consider a random variable equal to the distance from a point to its nearest neighbour, and to learn its probability distribution. Instead of relying on user defined parameters, a possible conjuncture is that each cluster has its own typical distance-to nearest-neighbour scale. The goal is to discover such scales. Such nonparametric approach is implemented in the algorithm DBCLASD (Distribution Based Clustering of Large Spatial Databases).

Assuming that points inside each cluster are uniformly distributed which may or may not be realistic, DBSCLAD defines a cluster as a non-empty arbitrary shape subset in X that has the expected distribution of distance to the nearest neighbour with a required confidence, and is the maximal connected set with this quality. This algorithm handles spatial data (minefield example is used) (X*X). -test is used to check distribution requirement (standard consequence is a requirement for each cluster to have at least 30 points). Regarding connectivity, DBCLASD relies on grid-based approach to generate cluster approximating polygons. The algorithm contains devices for handling real databases with noise and implements incremental unsupervised learning. Two venues are used. First, assignments are not final: points can change cluster membership. Second, certain points (noise) are not assigned, but are tried later. Therefore, once incrementally fetched points can be revisited internally. DBCLASD is known to run faster than CLARANS by a factor of 60 on some examples. In comparison with much more efficient DBSCAN, it can be 2- 3 times slower. However, DBCLASD requires no user input, while empirical search for appropriate parameter requires several DBSCAN runs. In addition, DBCLASD discovers clusters of different densities.

IV. Conclusion

In this, we have studied that existing web image search engines return a large quantity of search results, ranked by their relevance to the given query. Web users have to go through the list and look for the desired ones. This is a time consuming task since the returned results always contain multiple topics and these topics are mixed together. To get user relevant search there is need of different techniques of clustering. We have studied the different clustering techniques such partitioning method, hierarchical method & density based method. We can use respective method and user click through logs to get user relevant result at image search.

V. REFERENCE

[1] Zheng Lu, Xiaokang Yang, Senior Member, IEEE, Weiyao Lin, Hongyuan Zha, and Xiaolin ChenInferring User Image-Search Goals Under the Implicit Guidance of Users, March 2014.

[2] Z. Lu, H. Zha, X. Yang, W. Lin, and Z. Zheng, "A new algorithm for inferring user search goals with feedback sessions," March 2013.

[3] Mrs Monika Jain, Dr. S.K.Singh A Survey On: Content Based Image Retrieval Systems Using Clustering Techniques for large Data sets, November 2011.

[4] Deng Cai Xiaofei He Zhiwei Li Wei-Ying Ma and Ji-Rong Wen Hierarchical Clustering of WWW Image Search Results Using Visual, Textual and Link Information.

[5] Visual-Semantic Graphs: Using Queries to Reduce the Semantic Gap ate on the importance of the work or suggest applications and extensions.