Chapter 8

Query Morphing: A Proximity-Based Approach for Data Exploration

Jay Patel and Vikram Singh

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.77073

Abstract

We are living in age where large information in the form of structured and unstructured data is generated through social media, blogs, lab simulations, sensors etc. on daily basis. Due to this occurrences, acquisition of relevant information becomes a challenging task for humans. Fundamental understanding of complex schema and content is necessary for formulating data retrieval request. Therefore, instead of search, we need exploration in which a naïve user walks through the database and stops when satisfactory information is met. During this, a user iteratively transforms his search request in order to gain relevant information; morphing is an approach for generation of various transformation of input. We proposed 'Query morphing', an approach for query reformulation based on data exploration. Identified design concerns and implementation constraints are also discussed for the proposed approach.

Keywords: query transformation, query reformulation, proximity-based query processing, data exploration, exploratory search

1. Introduction

A fundament search activity begins with the formulation of search intension and mines meaningful information from available information space. This helps the user in gaining intellectual skills and cognitive understanding. Traditional search systems usually support lookup searching in that user has a proper wisdom of their information goal. This type of search relies on traditional 'Query-Result' paradigm in that user pose a query for the relevant document retrieval, browse through results and analyze them to fulfill his information need. This approach performs well in the case of short navigational information requests and fulfills

an information location need, but fails in information discovery need [39]. For discovery-oriented applications such as uncovering the information pattern from genomics, health care data, scientific data etc., additional assistance is required to formulate queries and navigation in data space to gain the desired information [16]. In such scenarios, the user usually uncertain about his information goals and/or less familiar with data semantics and context that makes the phrasing of information request [12] challenging. Also, initial search aims and intentions evolve as new information is encountered. Hence, the burden of analyzing, reorganizing and keeping track of the information gathered falls on the user alone [16, 17]. Exploratory search is one such emerging research area that realizes the importance of user's efforts in multiple phases of discovering, analyzing, and learning. Exploratory search systems can deliver pleasing quality information due to their recall-oriented reformulation from short typed ill-phrased query to precise query [23, 29, 37, 38].

User's search tasks can be categorized into three behaviors: Lookup, Learn and Investigate that is shown in **Figure 1**. The user may perform multiple types of search task in parallel, therefore searches are denoted by overlapping clouds. Generally, there is interplay between search tasks, for example lookup task interplay with investigate or learn. If we analyze the search behaviors, we can relate traditional search tasks with the lookup tasks in that carefully formulated queries yield precise result with the minimal relevance comparison. For exploratory search tasks, the system seeks more involvement beyond just a query specification and result presentation. A group of tasks allied with exploratory search is of type learn and investigate. Learning behaviour are aiming to knowledge acquisition in that user tries to develop addition, knowledge about the domain and better understand the problem context. It is an iterative process that simulates analogical thinking and relate users' experiences to return a set of data objects. Reformulating queries and comparing results take much time in learning

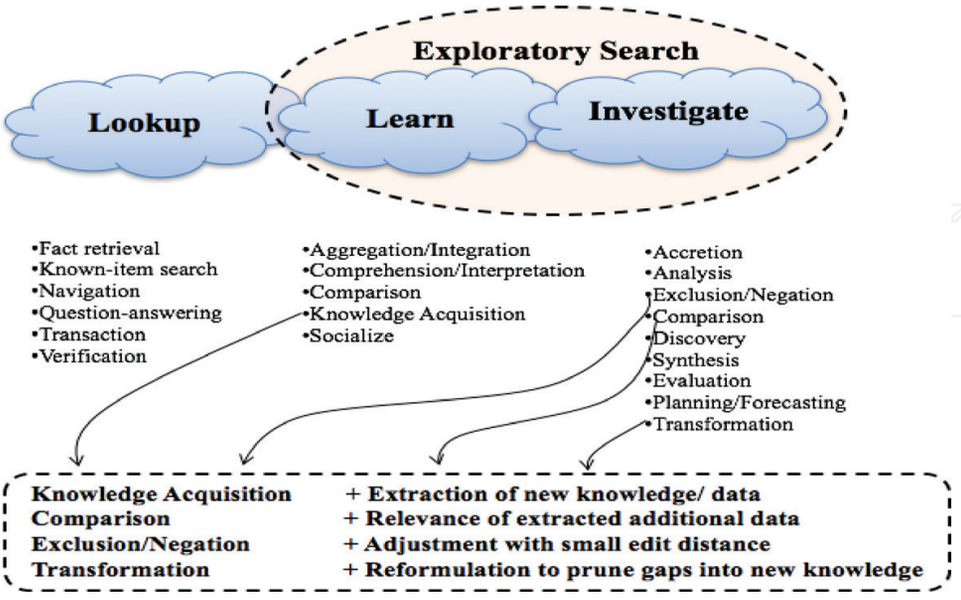


Figure 1. Exploratory search and sub-activities.

http://dx.doi.org/10.5772/intechopen.77073

tasks. Investigate behaviour prunes gap in knowledge and transform existing data into new knowledge.

Increase in several competing technologies leads to the generation of large structured and unstructured operational and transactional data. The key source, includes sensors, lab simulators, social media, web pages etc. In this setting, fundamental understanding of complex schema and content is necessary for formulating a data retrieval request otherwise user often stumbled upon empty or huge result set of his query. For such situations, we came up with an imitative towards 'Query Reformulation' as a vital task of Query Evaluation, named a 'Query morphing'. The proposal extract relevant and additional data objects from available data space and then recognize suggestions to acquire intermediate query reformulation.

Morphing refers to undergo a gradual process of transformation of input, e.g. for Image morphing [24, 10], Data Morphing [20]. Some traditional information retrieval techniques that transform initial query submitted by user are mapped in **Figure 2**. These transformations techniques aim to retrieve relevant information and improve system performance as well. Query reformulation techniques perform various transformations by applying user cognitive effort or system assistance and formulate semantically equivalent queries to reduce costs [26, 32, 40]. Pre classified data is required as database abstraction is performed for query reformulation. For successful reformulation it is better to understand the searchers intend and for that query rewriting can be a good option for query transformation. Query rewriting can be viewed as a generalization of query relaxation, query expansion [1, 40] and query substitution techniques [2, 40]. Query expansion techniques answer additional documents by evaluating inputs and expanding original user query through terms addition. Query relaxation techniques, conflict the expansion techniques [3]. Query relaxation is done to generalize query as sometimes ill-phrased query leads to fewer answer. Transformation process based on typical possible alternatives on original query is done in query substitution techniques [4]. An off-the-shelf dictionary/treasure is required for all these query transformation techniques [5].

Techniques grouped towards the left part in the **Figure 2** assist users for precise and unambiguous query formulation and execution. Various relevant query recommendations are generated and suggested that assist users in real-time query reformulation. Query suggestion techniques determines list of relevant queries that may help to achieve a user's search need [6, 18]. Query auto-completion techniques self-complete the formulation of queries have previously been observed in search logs. During a search, user often search is a sequence of queries of similar information need, query chain identifies this sequence. Query logs of earlier queries posed by global user are required to compute query suggestions list. Query recommendation

Figure 2. Query transformations and various equivalent techniques.

techniques track user's querying behaviour, identify the interested area from the available data space and recommends set of queries that retrieved relevant information. The query is steering [12, 26] is one process that navigates the user through complex data structures. For query recommendation and steering interactive query session is required to achieve ultimate search goal [12].

Due to the big data occurrences, traditional ways of query transformation repeatedly encountered challenges of relevance. To contrive such inherent challenges of transformation and relevance for exploration in large data a technique 'Query morphing' is designed. Our proposal suggests additional relevant data objects for the formulation of precise query by exploring available data space and leveraging use feedback. We concur that query morphing will also acquire the properties of traditional methodologies by observing that search query and respective results analogous to the history log.

1.1. Contribution and outline

The main contribution of this paper is an algorithm designed for query reformulation based on exploration technique. Algorithm named 'Query morphing' explores into the proximity of initial user query and extract additional relevant data objects. These retrieved data objects from the n-dimensional neighborhood assist user in his intermediate query reformulation. Proximate data objects are selected based on implicit and explicit relevance. We expect that our proposal, guides on exploration over several ample databases, such as Medical database, DNA database, social database, scientific database, etc. Finally, various existing reformulation techniques are revisited to establish the fact that how 'Query morphing' is different from traditional transformation techniques.

Next section listed some related research prospects and approaches. In Section 3 the proposed approach is conveyed, in which conceptual design is represented with algorithm and schematic diagram. Various design issues, analysis of implementations as well as intrinsic implementation complexity in proposed approach are recognized in Section 4. Lastly, the conclusion is presented.

2. Literature review

Many of today's query processing platforms carry a much profound repertoire and resilient querying techniques to regulate huge observational data in a limited resource environment. [12, 11]. Below some aspects are reviewed that helps user in searching relevant information from a large data set. We consider some prominent researches delivered for automatic exploration in data space, formulation of approximate quires and techniques that assist the user in the query formulation process to cover our aspects.

2.1. Automatic exploration

Analyzing vast amount of real time data can be an extremely complex task and required automation. In such scenario without any assistance, user ended up with ill-formulated query

that retrieves no result or huge result set. Traditional Database Management tools and systems are constructed by considering that database semantics is well understood by users [39]. Therefore, current applications with huge and complex database do not work well with these traditional Data Base Management techniques. Many interactive data exploration strategies are proposed and developed by researchers that extract and uncover great knowledge from complex data via highly ad-hoc interaction.

Automatic Interactive Data Exploration (AIDE) framework is well explained in [16] by authors. In that, the user is directed towards the data area of interest by deliberately incorporating relevance feedback. Various machine learning and data mining techniques can be integrated in that to achieve the best performance. Similarly, in [17] YAML framework is suggested and it uses attribute-value pair frequency to make exploration effective. Automatic exploration strategy performs formulation of user's queries and leads towards relevant information.

2.2. Query approximation

In exploratory query aspects where the user is satisfied with 'closed-enough' answer, approximation modules implemented in search system help to achieve shorter response time. This approximation module is built without changing underling database architecture. For example, Aqua approximate query answering system [4] rewrites queries using summery synopsis to provide approximate answers. Automatic Query Processing (AQP) widely uses statistical techniques based on the synopsis [14] to analyze large amount of data. Four main key synopsis are used by researchers for approximation which is random sample synopsis, histogram synopsis, wavelet and sketches synopsis.

Most fundamental and commonly used synopsis is a random sampling in that subset of data objects are fetched based on stochastic mechanism. It is easy to draw samples from a small available data, although to make the sampling process scalable, advance sampling techniques are required e.g. BlinkDB [6] architecture. In this architecture samples are selected based on accuracy of query and response time that device dynamic sampling strategy. A Histogram synopsis method group the data values into subset by summarizing the attribute frequency distribution or combined attribute frequency distribution. By using advance methods such as aggregation over joints are also used to approximate more general class of query. Another synopsis is wavelet synopsis which is identical with the above but the only variation is that it transforms and express most substantial data into the frequency domain. A faster response is one characteristic of approximate query processing. Speedup with accuracy is the key objective of AQP, therefore, returned results must be verified. Interactive approximate query processing performs error estimation [5] and error diagnosis via close forms or bootstrap that guarantees runtime efficiency and resource usage.

2.3. Assisted query formulation

Due to the big data contingency and complex schematic structure of data, sensible formalisms of query is required for complex information retrieval which is mastered by a small group of users usually. Most users in real life apply brute force approaches which manipulate data by hand as they have little knowledge regarding query formulation. Assisted query formulation

techniques are proposed to resolve these issues. These techniques assist the user by suggesting some query terms for subsequent formulation of incremental queries and reduction of irrelevant data retrieval. Fundamental operations such as equijoin and semijoin [11] are characterized for the formation of Boolean membership queries in polynomial time. A user membership driven learning algorithms [2] can also serves better formulation for simple Boolean queries. Many other formulation techniques for query construction such as locate minimal project join queries, discovering query approach [34] etc. are developed to answer query formulation similar to example tuples [27].

We termed our approach as 'Query morphing' because in literature a traditional method, morphing points transformation of inputs e.g. Data Morphing [20], Image morphing [10, 24]. Similarly, a small transformation of user queries are also carries out in our approach. We realized that the success of our approach is mainly rely on effective database exploration and user participation. The properties of traditional techniques are also incorporated in our query morphing approach as user's search request and retrieved outcomes analogous to the user history log.

3. Query morphing: a query reformulation approach

There are many tools available to extract knowledge from data, but they are inadequate in finding an appropriate subset of data. A deep analysis is needed to gain relevant knowledge [21] from available information space. Most of the tools follow the traditional lookup behavior that aims to retrieve the best literal match in a short time by assuming that the user is aware of 'what he is looking for'. That means systems are designed by considering that the user has a clear understanding about his search goals and familiar with database schema and context. It is observed that the success of the search process anticipates effective query articulation. Therefore, domain expert user successfully performs the search operation [33] and retrieve relevant results as he had formulated his query with appropriate terms [22, 23, 28]. But naïve user has to face challenges in the formation of his information seeking task due to less domain awareness. To resolve this he should be assisted through flexible query answering system [12] in query construction by delivering additional possible result sets along with original query results [25, 30]. The motive of such system is to reduce user's cognitive effort in subsequent queries [31, 8] by enhancing his knowledge.

Most systems support 'Query-Result' paradigm which is not sufficient as query formulation [12, 34] affects performance of the system. Instead 'Query-Result-Review-Query' paradigm can help as a user's search intention evolves with search progresses. The traditional methodologies retrieve results based on predefined relevant criteria and fails in identifying shift occurs in user's search intensions. Therefore a recall-oriented approach [19, 35] for query reformulation is designed. The idea behind it is as follows, the user poses an initial query Q and system yields effort T on finding the optimal results for Q. A small portion is set aside for the exploration. Various syntactically adjustments are performed with a small edit distance

http://dx.doi.org/10.5772/intechopen.77073

from Q to create variations Q_i. A conceptual example is shown in **Figure 3**. The result set returned by user's original query request is painted by the small red circle. Possible additional relevant result sets of user's interest for queries are explored in the large data spectrum, providing that result set belongs to surrounding closed region of the original request. Orange elliptic represents the query results that correspond to variations of original query request. After analyzing results user may formulate another query to shift interest towards another query result as shown in the right portion of **Figure 3**. A new region of query result of the user's request includes both new and previous variations of the query. A data space expedition and feedback incorporation is observed for the query reformulation and additional relevant data suggestion in this approach. The additional relevant data objects are retrieved by performing exploration and exploitation in available database. The properties of traditional techniques are also incorporated in our query morphing approach as user's search request and retrieved outcomes analogous to the user history log.

Creating transformation of input like text, image, data etc. is a fundamental process in computer science called morphing [10]. We analogous our query with the morphing inputs and named our reformulation algorithm as 'Query morphing' [41]. Our algorithm helps user in formulation of intermediate queries by creating variants/transformation of the original search query. The assistance to the user will be based on the optimal query reformulations derived during exploration and exploitation of dataspace. The proposed algorithms are developed by considering 'Query-Result- Review-Query' paradigm of computing [7, 15, 39]. The design framework for the same is conveyed in following section and shown in **Figure 4**.

3.1. Proposed approach

Our query reformulation approach can be seen into two sub activities, one is tradition query processing the other one is generation of morphs that derive intermediate query reformulation. Initially query Q_i will be validated and processed by the query engine in the traditional query processing mechanism by the DBMS. Data objects retrieved after processing initial query Q_i are identified on d-dimensional space that is already created and partitioned into non-overlapping rectangular cells and exploited for subsequent interactions. If we say more specifically, $S = D_1 \times D_2 \times \ldots, \times D_d$ be a d-dimensional space where $D = \{D_1, D_2, \ldots, D_d\}$ be a

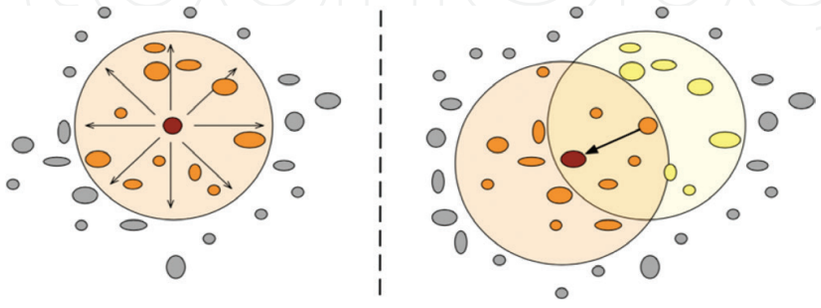


Figure 3. A conceptual example of query morphing.

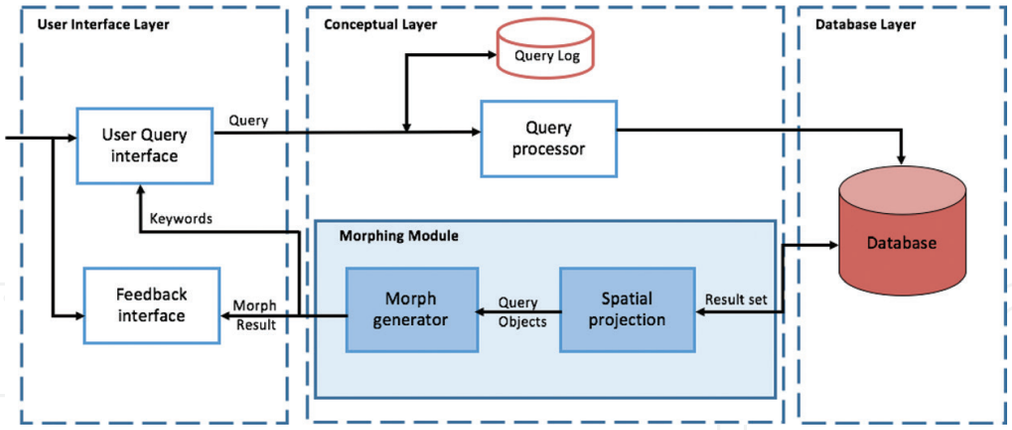


Figure 4. Query morphing and user's interactions.

set of totally ordered and bounded domains (attributes). Divide S into m^d non-overlapping rectangular cells by partitioning every dimension D_i ($1 \leq i \leq d$) into m equal length intervals. A d-dimensional data points, p, is projected in a grid cell, u, if in each attribute the value of v, is less than the right boundary of that attribute in u and greater than or equal to the left boundary of that attribute in u. When we consider exploration in high dimension, it can be assumed that relevant data points would exist in the close neighborhood [13, 16, 36]. Thus, the futuristic query formulation is pivotal by neighborhood exploration of each objects from previous queries. Neighborhood of each data object is initialized as a cluster of most probable results and achieved through sub-space clustering technique. A modified 'cluster-clique' algorithm is proposed for cluster/morph generation.

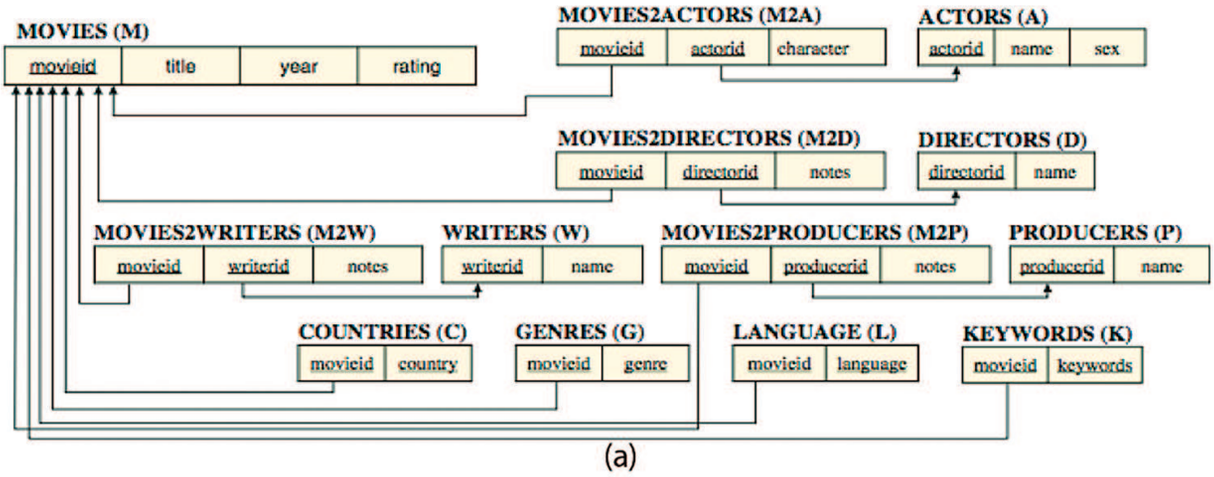
We assume that pre-computed d-dimensional sub space of data point divided into hyper rectangular cells is available. The query result retrieved after processing initial query traditionally is projected over this d-dimensional spatial representation of data. Identify the initial data object in space and consider them as a different unique cluster. Exploration and exploitation is performed in the neighborhood of every data objects to form cluster covering maximal region. The selectivity of a cell containing data points is defined to be the fraction of total data points in the cell. Only cells whose selectivity are greater than the value of model parameter τ are considered as dense and preserved. So, a cell is said to be a dense cell, if the fraction of total data point in that cell excesses input model parameter τ. The computation of dense cells applies to all subspaces of d-dimensional space. Identify neighboring dense cells that form a cluster containing data points at lower dimension. Cluster-clique holds cluster of dense cells at k-dimension also acquire similar projection at *(k-1)* dimension. The projection of subspace is considered from the bottom up to identify subspaces that contain clusters and to identify the dense cells to retain. A cell for giving projection subspace $S_t = A_{t1} \times A_{t2} \times \ldots, \times A_{tk}$ where $k < d$ and $the < t_j$, if $I < j$ is the intersection of an interval in each dimension. The proposed algorithm employs a bottom up scheme by leveraging the *Apriori* algorithm because monotonicity holds: if a group of data points is a cluster in a k-dimensional space, then this group of data points is also part of a cluster in any *(k-1)*-dimensional projections of this space.

http://dx.doi.org/10.5772/intechopen.77073

The proposed algorithm employs a bottom up scheme by leveraging the *Apriori* algorithm because monotonicity holds: if a group of data points is a cluster in a *k*-dimensional space, then this group of data points is also part of a cluster in any *(k-1)*-dimensional projections of this space. The recursive step from *(k-1)*-dimensional cells to *k*-dimensional cells involves a self-join of the *k-1* cells sharing first common *(k-2)*-dimensions. Cluster-clique thins the collection of candidates to reduce the time complexity of the *Apriori* process, and keep only the set of dense cells to form clusters in the next level. The portion of the database enclosed by the dense cells is called its coverage. All the subspaces are sorted according to their coverage and less covered subspaces are eliminated to perform thinning. The selection of cutting point between removed and taken subspaces is computed using MDL principle in information theory. Two connected k-dimensional cells u_1, u_2 have either common face in the subspaces or another *k*-dimensional unit u_s exist such that both the cells u_1 and u_2 is connected to u_s. A maximal set of connected dense units in k-dimensions form a cluster. Computing clusters is equivalent to computing connected components in the graph where the dense cells represent the vertices and cells sharing common face endures edges between them. This can be computed in quadratic time of the number of dense cells in worst situation. After the identification of all the clusters, a finite set of maximal segment or region is specified by applying a DNF expression whose union forms the cluster. Finding the minimal descriptions for the clusters is equivalent to finding optimal cover of the clusters. By examining all dense units, clusters are formed at higher dimensions and derived as query morphs. Top n keywords from the relevancy list are selected for suggestion to user for query reformulation.

Proposed system first process initial query of user Q_i in traditional way and return initial data result objects O $\{o_{i1}, o_{i2} \ldots o_{in}\}$. Returned data objects are projected on pre-computed *d*-dimensional sub space of data point divided into hyper rectangular cells. The algorithm will consider these projected data objects as independent clusters C $\{c_{i1}, c_{i2}, \ldots, c_{in}\}$. Next, cells in proximity (neighborhood) are explored and exploited to form a larger cluster. The cell is dense means cells containing at least τ data point are merged to form such clusters C $\{c_1, c_2, .., c_n\}$ at lower dimension. After identifying clusters at 1-dimentional subspace we subsequently move further in higher dimensions. As per the monotonicity interesting clusters and c_2 at 1-dimension exist, then at 2-dimensional subspace they can be form a unique cluster c_{12} if their intersection is dense enough and we can drop low dimension clusters c_1 and c_2 cluster from the cluster set. Similar computation is performed subsequently at 3rd, 4th, 5th and up to d^{th} dimension to form higher dimension clusters. We stop once we retrieve all the clusters. Now, we consider each cluster from the cluster set as independent morphs of initial user query (Q_i). Top N relevant morphs from the set is recommended to the user formulation of succeeding exploratory queries.

An example: refer schema of the movie database, initial query variant (Q_{i+1}) and corresponding result set shown in the **Figure 5**. {D.name = "D.Fincher"} is 1-dimension subspace cluster. {G.genre = "Drama", 1990 < M.year<2009} is 2-dimention subspace cluster. We are finding interesting fragments of data from the clusters: that values can occur on single or multiple attribute means on 1-dimensional or k-dimensional subspace cluster. We are looking for interesting



(a)

```
SELECT D.name, M.title, M.year, G.genre
FROMD,M2D,M,G
WHERE D.name = 'D. Fincher'
AND D.directorid = M2D.directorid
AND M2D.movieid = M.movieid
AND M.movieid = G.movieid;
```

(b)

D.name	M.title	M.year	G.genre
D. Fincher	The Aviator	2004	Biography
D. Fincher	Gangs of New York	2002	Drama
D. Fincher	Goodfellas	1990	Biography
D. Fincher	Casino	1995	Drama
D. Fincher	Shutter Island	2004	Thriller
D. Fincher	M. Jackson: Video Greatest Hits	1995	Drama
D. Fincher	The Last Waltz	1978	Documentary
D. Fincher	Raging Bull	1998	Drama

(c)

Figure 5. (a) imdb movie database schema (b) variant of initial query Q_{i+1} and (c) Result set of query Q_{i+1}.

pieces of information at the granularity of clusters: this may be value of a single attribute (1-dimensional cluster) or the value of k attributes (m-dimensional cluster). User want to retrieve all movies directed by 'D. Fincher'. For that refer example query shown in **Figure 5(b)**. From the retrieved results we can say that movies with genre "Drama" are frequently directed by "D. Fincher" so user possibly concerned in movies with {G.genre = "Drama"}. Similarlly, for {G.genre = "Drama",1992 < M.year<2009}. Moreover we intend to retrieval of potentially relevant data that may satisfies user's information but may not part of result set retrieved originally from the initial query. Consider following exploratory/variant of initial query (Q_{i+1}):

(Q_{i+1}): SELECT *D.name*

FROM *G, M2D, D, M*

WHERE D.name*! = 'D. Fincher'*

AND *G.genre = 'Drama'*

AND *D.directorid = M2D.directorid*

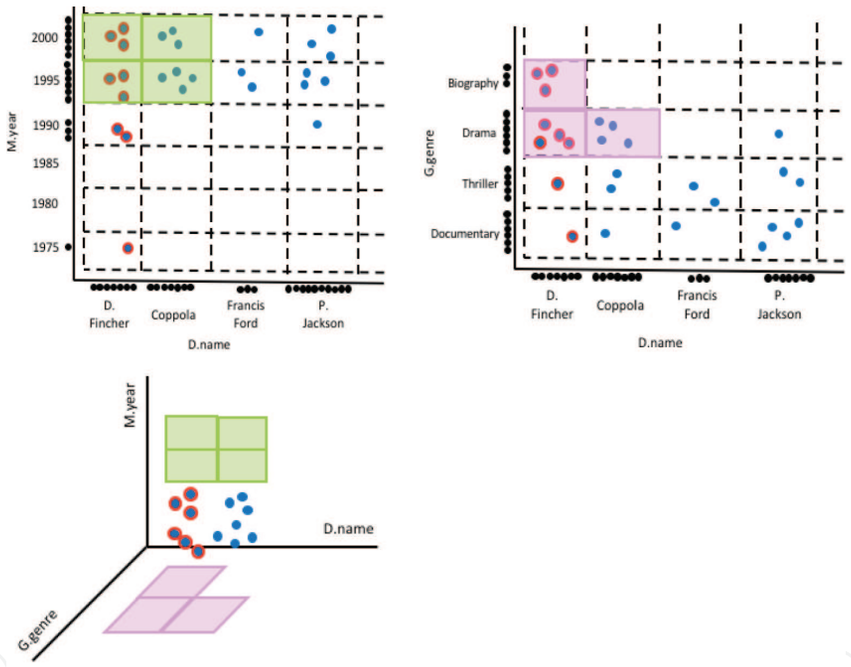
AND *D.directorid = M2D.directorid*

AND *M.movieid = G.movieid*

http://dx.doi.org/10.5772/intechopen.77073

Retrieve movies of other directors who have directed *drama movies* too, by considering that results are of user's interest. In the disigned approach, subspace clustering is used to generate these query morphs/variants and interesting additional results from variant quires. The system will compute dataset (**Figure 5(c)**) of initial query shown in **Figure 5(b)** and project it on the space.

Initially, all data points are projected on the d-dimensional space and data points of initial query result are identified. These data points are treated as initial cluster and then the neighborhood is explored to retrieve the larger cluster. As shown in **Figure 6** algorithms perform exploration and form larger cluster by merging neighborhood cells who are dense enough. In movie database, axis-paralleled histograms are constructed for the year and genre at 1-dimention. After 1-dimention next is to steer towards higher dimensions, and at 2-dimension like {G.genre, D.name} and {G.year, D.name.} etc. as shown in **Figure 6(a)** and **(b)**. Neighborhood exploration is performed and clusters are constructed. After finding all the cluster a finite set



D.name	M.title	M.year	G.genre
D. Fincher	The Aviator	2004	Biography
D. Fincher	Gangs of New York	2002	Drama
D. Fincher	Goodfellas	1990	Biography
D. Fincher	Casino	1995	Drama
D. Fincher	M. Jackson: Video Greatest Hits	1995	Drama
D. Fincher	The Last Waltz	1978	Biography
Coppola	Youth without Youth	2004	Drama
Coppola	A Hollywood Love story	2001	Drama
Coppola	Supernova	2000	Thriller
Coppola	Lani Loa – The Passage	1998	Thriller
Coppola	The Rainmaker	1997	Drama
Coppola	Battle Beyond the Sun	1996	Drama
Coppola	Don Juan DeMarco	1994	Documentary

Figure 6. (a) Cluster formation on 2D space {D.name, M.year} (b) cluster formation on 2D space {D.name, G.genre} (c) cluster formation on 3D space {D.name, M.year, G.genre} (d) results set of generated morphs.

of maximal segment (*regions*) are computed using DNF expression whose union is a cluster shown in **Figure 6(c)** at higher dimension. Subsequently, move to 3rd, 4th, 5th …. dth dimension in search of relevant clusters. This consummates the exploration of each data subspace around the pertinent objects of anterior query. All the computed clusters are equivalent to the morph of initial/previous query. Morphs contains addition relevant results as well subset of originally retrieved results. The data items present in the morphs are dignified as relevant by standard measure to previous query and future probable search interest. Now based on implicit and explicit relevance top N morphs and set of relevant terms are suggested to the user. In our example, after computing relevance score using standard relevance measures we can say that query morphs containing movie genre 'Drama' and directed year >1995 scored higher number then morph with movie genre 'Thriller'. Hence, morphs with movie genre 'Drama' and year >1995 considered as high relevance. The system would also suggest top N terms from computed morphs like 'Coppola' based on relevance to the initial query as well result set. These terms help the user in formulating his next exploratory/variant query. As a next step, user may encounter shift towards different query results after reviewing result variants. The newly formulated query now surrounds both past and new variant of the user request.

4. Design issues and analysis

Many design issues are identified during the development of the propose solution which are as follow:

1. **Neighborhood selection and query morph generation**: The key challenge is identifying and defining the borderline for neighborhood of relevant data objects. Various researchers have addressed in their existing work. In Proposed algorithm, subspace clustering is used to define a non-overlapping boundary based on relevance of neighborhood objects. Each neighborhood region will be explored and exploited for extraction of keywords and phrases query reformulation. If the density of d-dimensional spatial cell is less than the threshold (τ), then cluster forming becomes a challenging task. Exploring cluster at higher dimension may also face issues like cluster overlapping, cluster size, number of clusters.
2. **Evaluation of relevant data objects and Top N morph suggestion**: Relevance is estimated to measure how closely data objects of different clusters are connected and also to define importance of the result items [39]. Identification of various information to define relevance criteria is one of the key challenges, as it influences overall system performance. In our approach, each cluster will be exploited as a region of user's interest and data objects will be extracted based on explicit and implicit relevance measure. Two key issues are identified during designing that are criteria selection for relevance and techniques for the computation of relevance score.
3. **Demonstration of additional information extracted from retrieved data objects through various visualization:** A visualization of entire result set with frequent terms is not a

feasible solution [9], for this various data summarization technique can be employed. For example, relevant terms from the morphs are suggested to user in a selective manner, so that user can use these keywords for intermediate query formation.

Fundamentally several adjustments can be made to perform the query reformulations, such as adding/removing predicates, changing constants, joining operation through foreign key relationships on auxiliary tables, etc. The kind of adjustment for creation of intermediate query may steer towards relevant result set in optimal processing cost. Query morphing technique is regulating proximity-based query reformulation due to neighborhood exploration characteristics. The ultimate goal is to morph the query that pulls user in a direction where information is available at low cost.

5. Conclusion

We proposed an algorithm for query reformulation using object's proximity, 'Query morphing' that mainly design to recommend additional relevant data objects from neighborhood of the user's query results. Each relevant data object of user query act as an exemplar query for generation of optimal intermediate reformulations. Multiple challenges are inferred during solution designing, includes: (i) neighborhood selection and Query morph generation (ii) Evaluation of relevant data objects and Top-K morph (iv) Evaluation of data object's relevance, (III). Demonstration of additional information extracted from retrieved data objects through various visualization. The discussed approach primarily based on proximity-based data exploration, and generalized approach of query creation with small edit distance. It could be realized with major adjustments to the query optimizer. The ultimate goal would be that morphing the query pulls towards the area where information is accessible at low cost.

Author details

Jay Patel and Vikram Singh*

*Address all correspondence to: viks@nitkkr.ac.in

Computer Engineering Department, National Institute of Technology, Kurukshetra, Haryana, India

References

[1] Andolina S, Klouche K, Cabral D, Ruotsalo T, Jacucci G. Inspiration wall: Supporting idea generation through automatic information exploration. In: Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition. ACM; 2015. pp. 103-106

[2] Abouzied A et al. Learning and verifying quantified boolean queries by example. In: Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems. ACM; 2013. pp. 49-60

[3] Abouzied A, Hellerstein JM, Silberschatz A. Playful query specification with DataPlay. Proceedings of the VLDB Endowment. 2012;**5**(12):1938-1941

[4] Acharya S, Gibbons PB, Poosala V, Ramaswamy S. The aqua approximate query answering system. ACM SIGMOD Record. 1999;**28**(2):574-576. ACM

[5] Agarwal S et al. Knowing when you're wrong: Building fast and reliable approximate query processing systems. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. ACM; 2014. pp. 481-492

[6] Agarwal S, Mozafari B, Panda A, Milner H, Madden S, Stoica I. BlinkDB: Queries with bounded errors and bounded response times on very large data. In: Proceedings of the 8th ACM European Conference on Computer Systems. ACM; 2013. pp. 29-42

[7] Ahn JW, Brusilovsky P. Adaptive visualization for exploratory information retrieval. Information Processing & Management. 2013;**49**(5):1139-1164

[8] Andolina S et al. Intentstreams: Smart parallel search streams for branching exploratory search. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM; 2015. pp. 300-305

[9] Author ZY, Gao K, Zhang B, Li P. Time Tree: A novel way to visualize and manage exploratory search process. In: International Conference on Human-Computer Interaction. Chicago: Springer International Publishing; 2016. pp. 313-319

[10] Beier T, Neely S. Feature-based image metamorphosis. ACM SIGGRAPH Computer Graphics. 1992;**26**(2):35-42. ACM

[11] Bonifati A, Ciucanu R, Staworko S. Interactive inference of join queries. In: Gestion de Données-Principes. Technologies et Applications (BDA); 2014

[12] Cetintemel U et al. Query steering for interactive data exploration. CIDR. 2013

[13] Chau DH, Kittur A, Hong JI, Faloutsos C. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM; 2011. pp. 167-176

[14] Cormode G, Garofalakis M, Haas PJ, Jermaine C. Synopses for massive data: Samples, histograms, wavelets, sketches. Foundations and Trends in Databases. 2012;**4**(1-3):1-294

[15] Dhankar A, Singh V. A scalable query materialization algorithm for interactive data exploration. In: Parallel, Distributed and Grid Computing (PDGC), 2016 Fourth International Conference on. IEEE; 2016. pp. 128-133

[16] Dimitriadou K, Olga P, Yanlei D. Explore-by-example: An automatic query steering framework for interactive data exploration. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. ACM; 2014. pp. 517-528

[17] Drosou M, Evaggelia P. YmalDB: Exploring relational databases via result-driven recommendations. The VLDB. 2013;**22**(6):849-874

[18] Fan J, Li G, Zhou L. Interactive SQL query suggestion: Making databases user-friendly. In Data Engineering (ICDE), 2011 IEEE 27th International Conference on pp. 351-362. IEEE (2011)

[19] Glowacka D, Ruotsalo T, Konuyshkova K, Kaski S, Jacucci G. Directing exploratory search: Reinforcement learning from user interactions with keywords. In: Proceedings of the 2013 International Conference on Intelligent User Interfaces. ACM; 2013. pp. 117-128

[20] Hankins RA, Patel JM. Data morphing: An adaptive, cache-conscious storage technique. In: Proceedings of the 29th International Conference on Very Large Data Bases. Vol. 29. VLDB Endowment; 2003. pp. 417-428

[21] Hellerstein JM et al. Interactive data analysis: The control project. Computer. 1999; **32**(8):51-59

[22] Hellerstein JM, Haas PJ, Wang HJ. Online aggregation. In: Proceedings of the ACM SIGMOD Conference on Management of Data; 1997

[23] Idreos S, Papaemmanouil O, Chaudhuri S. Overview of data exploration techniques. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM; 2015. pp. 277-281

[24] Kersten ML, Idreos S, Manegold S, Liarou E. The researcher's guide to the data deluge: Querying a scientific database in just a few seconds. PVLDB Challenges and Visions. 2011:3

[25] Klouche K et al. Designing for exploratory search on touch devices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM; 2015. pp. 4189-4198

[26] Li H, Chan CY, Maier D. Query from examples: An iterative, data-driven approach to query construction. Proceedings of the VLDB Endowment. 2015;**8**(13):2158-2169

[27] Psallidas F, Ding B, Chakrabarti K, Chaudhuri S. S4: Top-k spreadsheet-style search for query discovery. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM; 2015. pp. 2001-2016

[28] Qarabaqi B, Riedewald M. User-driven refinement of imprecise queries. In: Proceedings of the International Conference on Data Endineering (ICDE); 2014

[29] Rocchio J. Relevance feedback in information retrieval. In: The Smart Retrieval System Experiments in Automatic Document Processing. Prentice-Hall Inc; 1971. pp. XXIII-1-XXIII-11

[30] Ruotsalo T, Jacucci G, Myllymäki P, Kaski S. Interactive intent modeling: Information discovery beyond search. Communications of the ACM. 2015;**58**(1):86-92

[31] Ruotsalo T et al. Directing exploratory search with interactive intent modeling. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management. ACM; 2013. pp. 1759-1764

[32] Salton G, Buckley C. Improving retrieval performance by relevance feedback. Readings in Information Retrieval. 1997;**24**(5):355-363

[33] Sellam T, Kersten ML. Meet Charles, big data query advisor. Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR). 2013;**13**:1-1

[34] Shen Y, Chakrabarti K, Chaudhuri S, Ding B, Novik L. Discovering queries based on example tuples. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data. ACM; 2014. pp. 493-504

[35] Singh V, Jain SK. A progressive query materialization for interactive data exploration. In: Proceeding of 1st International Workshop Social Data Analytics and Management (SoDAM'2016) Co-Located at 44thVLDB'2016. VLDB; 2016. pp. 1-10

[36] Stolte C, Tang D, Hanrahan P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. IEEE Transactions on Visualization and Computer Graphics. 2002;**8**(1):52-65

[37] White R, Muresan G, Marchionini G. Report on ACM SIGIR 2006 workshop on evaluating exploratory search systems. Acm Sigir Forum. 2006;**40**(2):52-60. ACM

[38] White R. Interactions with Search Systems. Cambridge University Press; 2016

[39] White RW, Roth RA. Exploratory search: Beyond the query-response paradigm. Synthesis Lectures on Information Concepts, Retrieval, and Services. 2009;**1**(1):1-98

[40] Yu JX, Qin L, Chang L, Ozsu MT. Keyword Search in Databases (Synthesis Lectures on Data Management). Morgan and Claypool Publishers; 2010

[41] Patel J, Singh V. Query morphing: A proximity-based approach for data exploration and query reformulation. In: International Conference on Mining Intelligence and Knowledge Exploration. Springer; 2017. pp. 261-273