A Heuristic Approach to Inductive Inference in Fact Retrieval Systems

C. William Skinner
North Carolina State University

Heuristic procedures are presented which have been developed to perform inferences by generalizing from available information. The procedures make use of a similarity structure which is imposed on the data base using nonnumerical clustering algorithms. They are implemented in a model fact retrieval system which uses a formal query language and a property-list data structure. A program of experiments is described wherein the procedures are used with test data bases which are altered by deleting part of the data and by purposely introducing false data. It is found that the system can infer the correct response under a variety of conditions involving incomplete and inconsistent data.

Key Words and Phrases: inference, inductive inference, clustering, fact retrieval, heuristics
CR Categories: 3.61, 3.71, 3.79

1. Introduction

For the purposes of this paper, a fact retrieval system is any information retrieval system which attempts to provide an explicit answer to an input query. This definition encompasses: (1) natural language question-answering systems which are primarily concerned with the processing of natural language input facts and queries, and their conversion into a logical structure of meaning; (2) data management systems, whose primary concern is the efficient storage and retrieval of facts in large data bases; and (3) inferential systems, where the primary emphasis is on the inference of a response to an input query when the answer is not explicitly available.

In the area of inferential systems, the major thrust of research has been toward a deductive inference capability rather than toward inductive inference. Deduction has a satisfying sense of completeness and consistency; the response is derived directly from known facts using pattern-matching operations or theorem-proving techniques. If the input facts upon which the inference is based are valid, then the response is true. Induction, on the other hand, is the process of generalizing from known facts and can never guarantee the truth of its inferred response. Yet, a major portion of human inference is certainly accomplished by inductive means, and, used in conjunction with deductive inference, it has proved to be a powerful methodology. In fact retrieval systems, inductive inference can be used to arrive at a plausible response when there are insufficient facts available to satisfy the requirements of a pattern-matching rule, or when the data base contains inconsistencies which would invalidate theorem-proving procedures. It may also be used as a preselection technique for deductive inference, to isolate a (hopefully) small set of relevant data from which the response may be deduced.

This paper presents a practical approach to inductive inference which is applicable to fact retrieval systems with large data bases. The method employed makes use of techniques for forming overlapping clusters in the data base and heuristic procedures for developing both an inferred response to an input query and a measure of the support for that response which is provided by the known facts. The inductive inference capability has been implemented in a model fact retrieval system on the CDC 6600, and a newer version is being developed for the IBM 370/165. The results of a number of experiments with the system indicate that this approach to inductive inference can be used effectively to answer questions when the answer is not given explicitly in the data base and when deductive inference is either infeasible or undesirable.

2. Definition of Inductive Inference

A wide variety of views of induction and inductive inference have been presented in the literature [1, 2]. The view taken here was perhaps best stated by John Stuart Mill more than a century ago [3].

> Induction, then, is that operation of mind by which we infer that what we know to be true in a particular case or cases will be true in all cases which resemble the former in certain assignable re-

Author's address: Department of Computer Science, North Carolina State University, Raleigh, NC 27607.

pects. In other words, induction is the process by which we conclude that what is true of certain individuals of a class is true of the whole class, or that what is true at certain times will be true in similar circumstances at all times.

It has been suggested that it is misleading to call this heuristic approach to inference formation "induction," since it is not founded in a firm basis of theory as is, say, statistical inference. The term "pattern extension" has been used [4] and is perhaps a more accurate description. However, the process given here does correspond closely with Mill's definition and fits the overall view of induction as extension from the particular to the general.

In any event, it is not the purpose of this paper to justify a particular view of induction. The intent is rather to present a practical approach for making inferences by generalizing from given facts. For this purpose, the following paradigm for inductive inference is used.

That which is known to be true in a number of cases is probably true in all cases which are sufficiently similar to them. The likelihood of this event increases with the number of cases for which the truth is known and decreases with the number of apparent contradictions.

3. A Model Fact Retrieval System

To facilitate the development of inductive inference procedures, a model fact retrieval system was constructed using LISP on the CDC 6600. No attempt was made to store and retrieve information efficiently in the data base of the model system, but rather to use a very general data structure which is obviously compatible with structures used for the efficient management of large data bases.

3.1 Data Structures

The basic data structure for the system consists of an unordered list of entities, which may represent objects, events, or concepts. Each entity is described by a property list (a list of attribute/value pairs). There are no requirements for particular characteristics to be used to describe any entity. Any attributes deemed pertinent for a given entity may be used, with values being either numerical or nonnumerical. Binary relations between entities are represented by treating the relation as an attribute of the subject entity with the name of the object entity as its value. This is a quite general representation for data, allowing any concept to be described by its properties or in terms of its relations to other concepts.

Since it is frequently desirable to access the data by property rather than by entity, a simple inverted file directory is used which contains for each property a list of the entities possessing it.

3.2 Formal Query Language

A simple formal language was chosen for queries and responses. No attempt was made to translate from natural language to the formal language within the system.

Four basic questions may be asked of the system:

1. Is it true that a given entity has a given property (attribute/value pair)?
2. What entity has a given property?
3. For a given entity, what attribute has a given value?
4. For a given entity, what is the value of a given attribute?

When the attribute represents a relation and the value represents another entity, the questions might be phrased as:

1. Is it true that two given entities stand in the given relationship?
2. What entity is the subject of a given relation with a given entity?
3. What relationship exists between two given entities?
4. What entity is the object of a given relation with a given entity?

In each case the formal query consists of a list whose first element is an entity name and whose second element is a list containing an attribute/value pair. In LISP notation this is written (entity (attribute value)). All elements of a query which requires a true/false answer are specified. When a query contains an unknown element, it is indicated by placing the symbol "U" in the appropriate position. For example, the question, "What is the length of the Enterprise?" would be represented by (Enterprise (length U)). The question, "Who is the wife of John Brown?" would be represented by (U(wife of John Brown)) and "Is Mary Brown the wife of John Brown?" by (Mary Brown (wife of John Brown)).

A query is only meaningful to the system if no more than one element is left unspecified. For the sake of simplicity the system is designed to return only one answer. The query structure (and the procedure for retrieval and inference) could be easily modified to allow for several answers to a query, where more than one might exist. "Multiple choice" questions may be represented by attaching a list of possible answers to the symbol U just prior to the query.

This formal language is by no means rich enough to represent all questions which can be asked in English without some rather bizarre interpretations of what may constitute an attribute or a value. However, a great variety of interesting questions can be asked using sequences of one or more queries in the language.

4. Inductive Inference in the Model

4.1 Similarity Structure

Since the procedures for performing inductive inference depend upon the degree of similarity between objects or properties, an appropriate measure of similarity must be developed. In the entity/property-list data representation, the similarity of entities may be based on the number (or proportion) of properties which they have in common. Likewise, the similarity of properties may be judged by the number (or proportion) of en-

tities on whose property lists they co-occur. If the assessment of similarity were made each time in response to a query, the inference time would increase sharply with increasing size of the data base. For this reason, it is desirable to initially impose a similarity structure upon the data so that only that portion of the data base which is expected to have a direct bearing on the question at hand need be examined in attempting to perform an inference.

A variety of clustering techniques are in use which attempt to organize a set of data into clusters, the elements of each being closely related in some way. If a clustering procedure is to be used to achieve a similarity structure appropriate for the inductive inference paradigm given above, it must have these characteristics:

—It must operate upon nonnumerical data.

—The resulting clusters must be overlapping to reflect that two objects or properties may be similar in more than one way.

—It must provide a numerical measure of similarity for the items within a cluster to help establish a measure of confidence in the inference.

The clustering method used in our experiments is that reported by N. Dale [5] for finding K-clumps. K-clumps are clusters of elements such that each element in a cluster is related to every other element in the cluster by an amount greater than some threshold value K. These clusters correspond to the cliques discussed by Sparck Jones [6, 7]. Clusters formed this way have proven very effective for our inductive inference procedures, but other clustering methods which satisfy the above requirements are being investigated for future use.

The basic data for the clustering process are a set of entities and a set of properties which describe them. A binary incidence matrix B can be formed where each column corresponds to an entity and each row to a property. $B_{ij} = 1$ if the ith property describes the jth entity; $B_{ij} = 0$ otherwise. A connection matrix C is then computed where both rows and columns correspond to entities. The following symmetric nearness measure is used to compute C:

$$C_{ij} = 1(i,j)/(1(i) + 1(j) - 1(i,j))$$

where $1(i)$ and $1(j)$ are the number of 1's in columns i and j of the incidence matrix, respectively, and $1(i, j)$ is the number of 1's in the logical product of columns i and j.

The clusters which result from the process are sets of entities such that each pair in a set has connection greater than some threshold K and no entity not in the set has connection greater than K to each member of the set. The same process is used for clustering properties, operating upon the transpose of the original incidence matrix.

Elements of an entity-cluster may be thought of as similar to one another, with the clustering threshold as an indication of the cohesion of the set. The data may be clustered several times using a range of threshold values. Since a given cluster may exist for a number of threshold values, the highest value at which a cluster is located is recorded as the threshold which characterizes the cluster.

Elements of a property-cluster may be regarded as predictors of one another, with the threshold as a measure of confidence in the prediction. That is, if property i is on the property list of an entity and a cluster with threshold K contains property i and property j, then one can predict with confidence proportional to K, that property j also belongs on the property list of that entity.

The clusters are found by pre-processing of the basic data prior to creating the data structure for the model fact retrieval system. The data base is then augmented by the clusters, each of which is itself represented as an entity whose description consists of lists of its members and a threshold value.

4.2 Inference Procedures

Inductive inference in the model fact retrieval system is accomplished by the use of eight basic procedures, one for each of the query types described in Section 3.2. A response to an input query consists of a repetition of the query with any unknown element replaced by the retrieved or inferred answer. In addition, a measure of the support for that answer contained in the data base is appended to the response. If the answer is found by retrieval, a "T" is inserted in place of the support measure. If no answer can be found by retrieval or inference, an "F" is inserted. Thus, for a typical query

(Dallas (located in state of U))

the possible responses are

(Dallas (located in state of Texas) T)	retrieval
(Dallas (located in state of Texas) .73)	inference
(Dallas (located in state of U) F)	no answer

The inference procedures are heuristic in nature. There is no assurance that an answer will always be found nor that a chosen answer is the best possible one which can be inferred from the data base. However, in a series of experiments in which the answer to a query was removed from the data base, that answer was inferred by the procedures in the great majority of cases.

The particular heuristics used in the inference procedures represent an attempt at a practical implementation of the inductive inference paradigm stated in Section 2. In each case, the similarity structure obtained through clustering is used to isolate those entities or properties which are similar to the entity or property presented in the query. The characteristics of the similar entities or properties are then extended to infer the response to the query.

To illustrate in detail how one of the procedures works, consider the query

(Salem (type designation U))

given to the system with a data base extracted from

Jane's Fighting Ships [5]. Stated in English, the query is, "What type of ship is the USS Salem?" The correct response is

(Salem (type designation heavy cruiser)).

The data base contains an entry for the entity Salem with a property list of some 25 properties, such as

builders	Bethlehem Steel Co.
construction site	Quincy, Massachusetts
propulsion	steam
full load displacement	20,000–30,000 tons

The attribute "type designation" is not on the Salem property list. It is, however, on the property lists of many other ships described in the data base. Associated with it are a variety of values such as "attack aircraft carrier," "heavy cruiser," and "guided missile light cruiser." Each of the possible values of "type designation" is considered as a candidate for the response and the degree to which each value is supported by the data base is computed.

The level of support which is found in the data base for a particular value is derived from the similarity structure which has already been imposed on the data. In general, each property on the property list of the query-entity may belong to several clusters which also contain the property under consideration. For each property on the list, the maximum threshold among these clusters is taken as the degree of support provided by the presence of that property for the property under consideration as a candidate for the response. The total support for the property, which is called the level of evidential support, is the sum of these threshold values, divided by the sum of the maximum threshold values for all properties which cluster with the candidate property. The level of evidential support found this way ranges from zero when no property which belongs to the query-entity shares a cluster with the property under consideration, to unity when every property which clusters with the property under consideration is on the property list of the entity given in the query.

If at least one possible value for the attribute "type designation" has a level of support greater than some acceptable value, which has been input by the user, the value with the greatest support is chosen and is injected into the response along with its support value.

If sufficient support for a response is not provided by the entries on the property list when examined individually, support may be sought from the properties when considered in pairs or higher order groupings. It might be, for example, that neither "(builders Bethlehem Steel Co.)" nor "full load displacement 20,000–30,000 tons)" is a good predictor of "(type designation heavy cruiser)" when considered individually, but the presence of both on a property list might provide a very good predictor.

In theory it would be possible to form clusters in preprocessing similar to those already formed but with each pair of properties in turn considered as a single property. These clumps could then be compared with the query-entity and the level of support for each candidate response computed as in the first part of the procedure. However, since there are $n(n-1)/2$ possible pairs, where n is the number of properties being considered, it is not feasible to perform clumping of all such pairs. Instead, a process analogous to forming the clumps and then computing the support level is performed at inference time, when the number of properties to be considered can be greatly reduced. Only those properties on the property-list of the query-entity, when considered in pairs, can provide support for the candidate property. To further reduce the number of pairs to be examined, only those properties which share clump membership with the candidate property are considered. The degree of support provided by each pair of the latter set of properties is computed by examining the property-lists of all entities in the data base and determining the quotient

$$N(pq)/(N(p) + N(q) - N(pq))$$

where: $N(pq)$ is the number of entities possessing both the pair of properties and the candidate property; $N(p)$ is the number of entities possessing the pair; and $N(q)$ is the number of entities possessing the candidate. The level of evidential support is then the sum of such quotients for all pairs which are on the property-list of the query-entity divided by the sum of such quotients for all pairs being considered.

In principle, if an acceptable level of support were not provided by any pair, then all triplets, quadruplets, etc., could be tried. However, the marginal return which might be expected from implementing such procedures was not considered worthwhile for the experimental system. In fact, although this second part of the procedure enables the system to make correct inferences in certain cases where it could not otherwise do so, the time required is proportional to the size of the data base. Since the experiments performed with test data indicate that the great majority of queries which can be answered by this procedure can be answered without resorting to pairs, an operational system might be more efficient using only the first part of the procedure. Alternatively, one might wish to analyze the data prior to clustering and select a small number of pairs which are likely to act as good predictors of other properties. Each of these pairs could then be treated as a single additional property during the clustering process.

Each of the other query forms is treated by a heuristic procedure similar to this one. When the query contains an *attribute* in the second position, as in this example, the property clusters are used in the inference procedure. When a *relation* is in the second position, as in the query

(Cincinnati (in same state with Dayton)),

the entity clusters are used.

5. Inference Experiments

5.1 Program of Experiments

To examine the effectiveness of this approach to inductive inference, the model system was exercised with two data bases and 48 queries (24 with each data base). The program of experiments was designed to test the ability of the system to develop correct inferences from each data base under three sets of conditions:

1. Full data base—a data base with a high degree of commonality of attributes among the entities, as well as a high degree of redundancy in the data.
2. Incomplete data base—the same data base with progressively greater portions of the data randomly deleted.
3. Inconsistent data base—the full data base with progressively greater portions replaced by false and conflicting data.

The experiments with the inconsistent data bases were performed to determine to what degree the inductive inference procedures can function when there are errors and conflicting statements within the data base, a condition which would sharply limit the use of deductive procedures.

The two distinct data bases used in the experiments are:

1. Ship data base—a set of input data on United States naval vessels, extracted from *Jane's Fighting Ships*, 1962–63 [8]. The data set contains 74 entities (ships) each described by about 25 properties. Much of the data is numerical and so in each case the range of values has been divided into a number (usually 10 to 20) of discrete intervals. This effectively converts continuous information to discrete form, which is more suitable for the clumping procedures. The properties consist primarily of attribute/value pairs with only a small number of relation/entity pairs.
2. City data base—a set of input data on the nation's largest cities extracted from *Statistical Abstract of the United States: 1968* [9]. The set contains 55 entities, each described by about 10 attribute/value properties and about 30 relation/entity properties which have been deduced from the attribute/values. Numerical data have been treated in the same way as for ships.

A series of fact retrieval experiments was performed to validate the inference procedures and to explore their capabilities and limitations. In each case the fact which would answer the query by retrieval was deleted from the data base. Thus each response could be evaluated as correct or incorrect. The program of experiments was:

1. Ship data base: (a) 24 queries (three of each of the eight types described in Section 3.2) using the full data base (only the answer to the query removed); and (b) the same 24 queries with 10 percent of the data (randomly chosen) deleted from the data base in addition to the particular query answers.
2. City data base: (a) 24 queries (three of each type) using the full data base; (b) the same queries with progressively larger portions of the data base deleted, starting with 10 percent, 20 percent, etc., until satisfactory answers could no longer be obtained; and (c) the same 24 queries with progressively larger amounts of false data introduced into the data base, starting with 10 percent, 20 percent, etc., until satisfactory answers could no longer be obtained. The false data were chosen to be as far from correct as possible without introducing new properties or relations into the data base.

Table I. Experiments with Ship Data Base

query	Full Data Base	Deletions 10%
(1)	0.87	0.68
(2)	1.00	0.83
(3)	0.67	0.41
(4)	—	—
(5)	0.20	0.19
(6)	—	—

Table II. Experiments with City Data Base

query	Full Data Base	Deletions 10%	Deletions 20%	Deletions 30%	False Data 10%	False Data 20%	False Data 30%
(7)	0.41	0.38	0.32	—	0.32	0.30	—
(8)	0.75	0.54	0.54	0.49	0.61	0.55	0.42
(9)	0.50	0.45	0.39	0.33	0.44	0.38	0.35
(10)	0.05	0.03	0.02	0.01	0.05	0.03	0.02
(11)	0.03	0.02	0.01	—	0.02	0.01	—
(12)	0.02	0.02	0.01	—	0.02	0.01	—

The queries submitted to the systems with the ship data base were the four different forms of each of the following:

(1) (Forrestal (full load displacement 70,000–80,000 tons))
(2) (Salem (type designation heavy cruiser))
(3) (Biddle (cost 25,000,000–50,000,000 dollars))
(4) (America (larger than Forrestal))
(5) (Saratoga (flight deck longer than Coral Sea))
(6) (America (radar system improved over Enterprise))

For the city data base, the queries were the four different forms of each of the following:

(7) (Louisville (population density 4500–6750 per sq. mile))
(8) (Cincinnati (average minimum temperature 45–50 degrees))
(9) (San Francisco (located in the state of California))
(10) (Houston (warmer in winter than Memphis))
(11) (Cincinnati (in the same state as Dayton))
(12) (Baltimore (larger population than New Orleans))

5.2 Results of the Experiments

Tables I and II show the level of evidential support computed for the inferred response (i.e. the one with the highest level of support) to each query under each set of conditions. The level of support for a given response is independent of the form in which the query was posed. A dash indicates that no answer with an acceptable level of support could be inferred. The response inferred by the system was the correct answer to the query in every case except query (8) (Cincinnati (average minimum temperature 45–50 degrees)). For this query, the inferred response, under all conditions of the data base, was (Cincinnati (average minimum temperature 40–45

degrees)) with the correct answer having the next highest level of support. This would indicate that, in the limited data base being used, the characteristics of Cincinnati associated with temperature more nearly implied a slightly colder average minimum than actually occurs.

The ship data consist primarily of attribute/value pairs with only a small number of relation/entity pairs. There were not sufficient instances of any relation to enable the system to perform inductive inference for queries (4) and (6), even with the full data base. In the city data base, which is rich in relations, all the queries could be answered with the full data base.

A quite respectable level of support was obtained from both the ship data base and the city data base for queries of the (entity (attribute value)) type. Queries (1), (2), (3), (7), (8), and (9) are of this type. For these queries the acceptable level of support was set at 0.30 and queries generating a response with level of support lower than this acceptable level were considered unanswerable by the system.

For queries of the (entity (relation entity)) type, the computed level of support was much lower, causing the acceptable level to be set at 0.01. A data base in which many more relations among the entities are given is needed to provide a high level of support for responses to queries of this type. In our experiments with the limited data bases, each time a response was inferred, it was correct, but the level of support was very low.

The experiments with successively larger portions of the original data deleted indicate a proportionate decrease in the level of support for the chosen response, but in each case, the same response was inferred as for the full data base.

The experiments with successively larger portions of the data replaced by false and, as far as possible, contradictory data are perhaps the most interesting in the program. The system was still able to infer the correct response in all the same cases, with levels of support slightly lower than when a corresponding portion of the data was simply deleted. It is evident that the system is not heavily influenced by facts which contradict the bulk of supporting evidence.

The computational work involved in performing the inferences is divided into two parts: (1) the initial clustering of the data to form the similarity structure; and (2) execution of a heuristic procedure which uses the similarity structure to develop the inferred response. The initial clustering requires a significant amount of computation, but is performed only once. (Techniques have been developed for updating the data base without completely reclustering [10].) Once the similarity structure has been established, the response to a query can be inferred with relatively little computation. For example, in the experiments with the complete ship data base (using LISP on the CDC 6600), less than one second of computing time was needed to infer a response to queries with all elements specified, while queries with an omitted element averaged about two seconds.

6. Potential Uses

Inductive inference performed in this way can be especially useful for systems with large data bases compiled from several sources, where particular items of information may not be available for any given entity, and where a certain number of errors and inconsistencies can be expected to exist. In a total fact retrieval system inductive inference can complement deductive inference techniques, being called into play when deduction is either impractical to attempt or does not succeed. Alternatively, as Simmons has suggested for statistical induction techniques [11], it can act as a first stage filter to quickly isolate the most likely answers to a query. Deductive procedures can then be used to select the precise answer from this small set.

7. Summary

A practical approach to performing inductive inference in fact retrieval systems has been presented. The approach is useful for nonquantifiable data bases which can be expressed as entities described by properties and by relations to other entities. The inference procedures are heuristic and depend upon initially clustering both the entities and the properties within the data base. Results of experiments with relatively small data bases are quite encouraging, indicating that correct inferences can be made under a variety of conditions of incomplete and inconsistent data.

Acknowledgments. The guidance and encouragement by Alfred G. Dale, Department of Computer Sciences, The University of Texas at Austin, were especially helpful in conducting this research.

Received May 1973; revised August 1974

References

1. Kyburg, Henry E. Recent work in inductive logic. *Amer. Philosoph. Quart. 1* (Oct. 1964), 249–287.
2. Katz, Jerrold J. *The Problem of Induction and Its Solution.* Univ. of Chicago Press, Chicago, 1962.
3. Mill, J.S. *A System of Logic.* Longmans, Green and Co., London, 1868.
4. Simmons, R.F. Oral communication with the author.
5. Dale, N. *Automatic Classification System Users' Manual,* LRC 66 TS-I. The University of Texas Linguistics Research Center, Austin, 1966.
6. Jones, K.S., Automatic term classification and information retrieval, *Proc. IFIP Cong. 1968,* North Holland Pub. Co., Amsterdam, pp. 1290–1295.
7. Jones, K.S., and Jackson, D.M. The use of automatically-obtained keyword classifications for information retrieval. *Inform. Stor. Retr. 5* (1970), 175–201.
8. Blackmun, Raymond V.B. (Ed.), *Jane's Fighting Ships 1962–63.* Sampson, Low, Marston and Co., London, 1962.
9. U.S. Bureau of the Census. *Statistical Abstract of the United States: 1968,* Washington, D.C., 1968.
10. Skinner, C.W. Inductive inference in fact retrieval systems. Ph.D. Diss., U. of Texas, Austin, 1970.
11. Simmons, R.F. Natural language question-answering systems: 1969, *Comm. ACM 13,* 1 (Jan. 1970), 15–30.