Ontology-Driven Information Extraction and Knowledge Acquisition from Heterogeneous, Distributed, Autonomous Biological Data Sources

Vasant Honavar, Carson Andorf, Doina Caragea, Adrian Silvescu, Jaime Reinoso-Castillo, and Drena Dobbs
Artificial Intelligence Research Laboratory
Department of Computer Science
226 Atanasoff Hall, Iowa State University
Ames, Iowa 50011-1040
Honavar@cs.iastate.edu
www.cs.iastate.edu/~honavar/aigroup.html

Abstract

Scientific discovery in data rich domains (e.g., biological sciences, atmospheric sciences) presents several challenges in information extraction and knowledge acquisition from heterogeneous, distributed, autonomously operated, dynamic data sources. This paper describes these problems and outlines the key elements of algorithmic and systems solutions for computer assisted scientific discovery in such domains. These include: ontology-assisted approaches to *customizable* data integration and information extraction from heterogeneous, distributed data sources; *distributed* data mining algorithms for knowledge acquisition from large, distributed data sets which obviate the need for transmitting large volumes of data across the network; ontology-driven approaches to *exploratory data analysis* from alternative ontological perspectives; and modular and extensible agent-based implementations of the algorithms within a platform-independent agent infrastructure. Prototype implementations of the proposed system are being used for discovery of macromolecular structure-function relationships in computational biology and distributed coordinated intrusion detection in computer networks.

Challenges in Integration and Analysis of Heterogeneous Distributed Data

Development of high throughput data acquisition technologies in biological sciences, together with advances in digital storage, computing, and communications technologies have resulted in unprecedented opportunities for large scale, computer assisted, data-driven scientific discovery [Baxevanis et al., 1999]. Data sets of interest to computational biologists are often heterogeneous in structure, content, and semantics. Examples include sequence data (DNA, RNA, and protein sequences, expressed sequence tags) [Benson et al., 1997; Boguski et al., 1997]; numeric measurements (e.g., gene expression data); symbolic data describing relations among entities; structured or semi-structured text (e.g., annotations associated with DNA sequences, protein structures, and gene expression data); temporal data (e.g., gene expression time series); structures containing numeric as well as symbolic information (e.g., 3-dimensional protein structures); and results of various types of analysis [Baxevanis, 2000; Discala et al., 2000]. They currently include data stored in flat files, relational databases, and object-oriented databases. The term biological database is used loosely to refer to a biological data collection in any of these forms. How best to organize genome data is still a matter of debate [Frenkel, 1991; Gelbart, 1998] although several object-oriented databases and have been proposed in recent years [Gray, 1990; Goodman, 1995; Ghosh, 1999; Durbin, 1991]. Applications such as characterization of macromolecular structure function relationships and inference of genetic regulatory pathways require selection and extraction of relevant information from such data (e.g., features from sequences, counts and statistical summaries from measurements, structured representation of relevant information from textual annotations). They also call for data integration from multiple sources into a coherent form that lends itself to further analysis (e.g., data mining) by bridging syntactic and semantic gaps among them. Typical data analysis tasks that arise in computational biology are difficult to express using standard query languages and thus application programs have to be constructed using program libraries. While queries expressed in declarative languages like SQL are still useful in biological databases, the use of programming interfaces is unavoidable for many types of data analysis (e.g., data mining). This follows from the fact that the same set of data may have to be analyzed in different ways depending on the information extraction and knowledge acquisition objectives of the user. It is impossible to foresee all the potential uses of data when designing data repositories or data analysis services.

The data sources of interest in computational molecular biology are large, diverse in structure and content, and typically autonomously maintained [Fasman, 1994]. Transforming these data into useful knowledge (e.g., inference of genetic networks from gene expression data, building predictive models of protein function from protein sequence) calls for algorithmic and systems solutions for computer assisted knowledge acquisition and data and knowledge visualization. Machine learning algorithms [Mitchell, 1997] currently offer one of the most cost effective approaches to data-driven knowledge acquisition (discovery of features, correlations, and other complex relationships and hypotheses that describe potentially interesting regularities from large data sets) in increasingly data rich domains such as computational biology [Baldi and Brunak, 1998]. However, application of machine learning algorithms to large scale knowledge discovery from heterogeneous distributed data presents several challenges [Thrun et al., 1999]. Our work is aimed at addressing some of these challenges. Most currently available learning algorithms are *batch algorithms* in that they assume that the learner has access to the entire data set before data driven knowledge acquisition can proceed. Given the large size and distributed and dynamic nature of data sets encountered in computational molecular biology, it is neither feasible nor desirable to gather all of the data in a centralized database or exchange entire data sets among different sites. This goal is not always achievable (e.g.. when information required from two databases is the result of performing the relational join operation on the two databases). However, whenever feasible, it is desirable to develop information extraction and data integration techniques that can operate on multiple distributed data sets without collecting all the data in a centralized location, thereby obviating the need to transmit large amounts of data between different data repositories or between data sources and users. Thus it is necessary to develop distributed learning algorithms that can acquire knowledge from distributed data sets without collecting all of the data in a centralized location, or the need to transmit large amounts of data between different data repositories or between data sources and users [Honavar et al., 1998; Honavar et al., 2001; Caragea et al., 2001a]. When both the data and computational resources needed are not available at the same location, it is necessary to establish processing centers e.g., computation servers, and data warehouses where data from multiple sources can be integrated and analyzed. Thus, it is highly desirable in a distributed environment, for users to be able to supply computational procedures that can be executed on data from remote sites. Since the data repositories reside on heterogeneous hardware and software platforms, this requires a platform-independent execution environment for user-developed programs. The distributed computing paradigm based on transportable procedures or *mobile agents* [White, 1997] supports such agent-based systems made of agents (software entities capable of flexible, autonomous action within the constraints imposed by their environment so as to achieve their design objectives) [Jennings et al., 2000]. Mobile agent infrastructures (e.g., the SMART system developed in our lab, or the commercially available Voyager (www.objectspace.com/) system), allow users to not only call procedures on remote computers, but to also dynamically supply the procedures (mobile agents) to be executed on remote computers in a platform independent execution environment.

Many biological databases are dynamic: they constantly accumulate new data (and perhaps less frequently, undergo updates of existing data e.g., due to correction of previously supplied erroneous annotations). Thus it is necessary to develop incremental learning algorithms that discover interesting regularities and refine existing models as new data become available. Similarly, there is a need for data visualization techniques for dealing with large, distributed data sets.

In many applications, the individual data sets are autonomously owned and maintained. Consequently, access to the raw data may be limited and only summaries of the data (e.g., number of instances that match some criteria of interest) maybe made available to users. *In such cases, there is a need for distributed learning algorithms that can operate within the data access constraints imposed on them.*

Biological Ontologies

Task or context-specific analysis of biological data requires exploiting the relations between terms used to specify the data, to extract the relevant information and integrate the results in a coherent form. For example, assignment of a biological function to a putative protein from a genomic sequence involves relations between terms such as nucleotide, gene, amino acid, protein, motif, domain, tertiary structure). Ontologies, [Sowa, 1999; Uschold and Jasper, 1999] specify terms; relationships among terms (e.g., father and mother are both parents). Different ontologies can provide different perspectives on the same domain of discourse. A number of ontologies designed to support machine-readable annotations of biological data are currently under development [Ashburner et al., 2000; Karp, 2000]. (http://www.geneontology.org, http://smi-web.stanford.edu/projects/bio-ontology/). Such ontologies also facilitate sharing of data and knowledge among computational biologists [Karp, 1996; Schultz-Kremer, 1997]. Types of ontologies that are commonly encountered in biology include:

- *Taxonomies* which correspond to the familiar *isa hierarchies* used for knowledge representation in artificial intelligence; Examples include the *genetic code* which specifies the mapping between gene sequences that use a 4-letter alphabet of nucleotides and the corresponding protein sequences defined over a 20-letter alphabet of amino acids; classification of proteins into functional families; classification of amino acids into classes based on specific properties of interest e.g., hydrophobicity or charge; *phylogenetic trees* which represent evolutionary relationships among organisms;.
- *Part-whole* relationships which correspond to the *part-of* hierarchies used for knowledge representation in artificial intelligence; Examples include specification of functionally significant *motifs* or sequence patterns that occur in a protein sequence; specification of recognizable *domains* of a 3-dimensional structure of a protein; specification of biologically relevant parts of the DNA sequence such as introns, exons, promoters, open reading frames, etc.

Ontologies are typically encoded using a *declarative* form of knowledge representation (e.g., first order logic or its variants). In order for ontologies to be useful for generating alternative representations of the data from different ontological perspectives or for integrating *data* from multiple sources, the definition of terms and relationships among terms need to be augmented with *functions* for computing values of some subset of terms from the known values of related terms (e.g., prediction of membrane spanning domain of a protein based on hydrophobicity of amino acids). This requires the use of domain, application, or even data source specific rules and transformations for integrating data and metadata. In what follows, we use the term ontologies in this broader sense.

Role of Ontologies in Data Driven Knowledge Acquisition

The following simple example illustrates how different *ontologies* can drive the analysis of the available data by scientists in different *contexts*, under different sets of assumptions, to explore different scientific hypotheses. Consider an attempt to discover predictive relationships between sequence regularities and protein structure (and possibly function). There are multiple ways to represent protein sequences (depending on the investigator's ontological perspective). For instance, sequences could be represented using the 20-letter alphabet of amino acids, or they could be represented using a 2-letter alphabet {C, U} with C denoting a charged amino acid and U denoting an uncharged (U) amino acid. Sequence regularities that are not readily apparent in the former case often pop out dramatically in the latter case (Figure 1). Note that in rhodopsin, the segregation of charged residues (denoted by small circles) towards the ends of the molecule, together with presence of uncharged residues in the center, `explains' the fact that it is a membrane spanning protein whereas myoglobin, which has a more even distribution of charged residues, is not.

Our colleagues have demonstrated that apparent inconsistencies between evolutionary trees based sequence data and those based on morphology can be reconciled by representing sequences using a two letter alphabet {H, P} where H stands for a *hydrophobic* amino acid and P denotes a *hydrophilic* amino acid [Naylor and Brown, 1998].

The preceding examples demonstrate how (implicit or explicit) *ontologies* of scientists facilitate scientific discovery by imposing different ontological perspectives on the data. Exploratory data analysis in science often involves search for regularities or potentially explanatory patterns in large data sets from different ontological perspectives. Thus, environments for computer-assisted knowledge acquisition need to support tools for generating alternative representations or *views* of biological data sets using different user-specified ontologies.

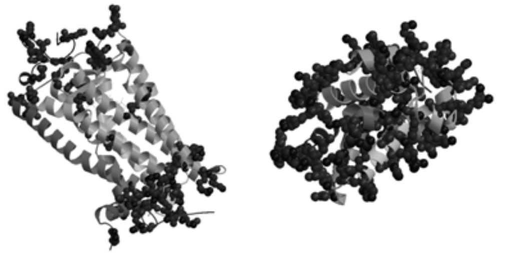

Figure 1: The dramatic differences in the distribution of charged amino acids (denoted by dark circles) on rhodopsin (left) and myoglobin (right) molecules offers insights into their respective functions

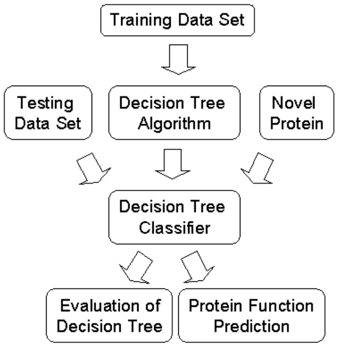

Figure 2: Generating decision trees for assigning protein sequences to functional families

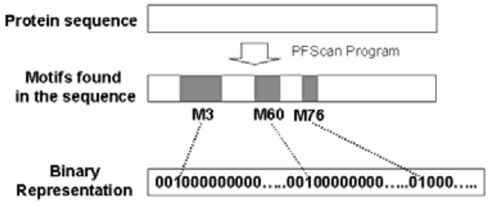

Figure 3: Generation of motif based fixed length binary representation of a protein sequence

Data sets of interest in computational biology are heterogeneous in structure and content and are often distributed across multiple, autonomously maintained databases that are accessible through the Internet. Computational biologists typically access the data sets of interest to them by querying the relevant database(s) and use their knowledge of the domain and specific application context to extract the relevant information manually or by executing a set of programs. Consider for example, data-driven construction of decision trees to classify protein sequences into functional families [Wang et al., 2001] shown schematically in Figure 2. First, we retrieve a set of protein sequences along with the corresponding sequence identifiers and function labels from the PROSITE database. We submit the sequence identifier to the Profilescan program to obtain a list of sequence *motifs* (relatively short potentially functionally significant sequence patterns) present in each sequence. The sequence identifiers and the list of motifs associated with each sequence are stored in a file. We collect the list of motifs to form a *motif set W* with |*W*| distinct motifs (Figure 3). Each sequence is then represented as a binary pattern of |*W*| 1s and 0s (with a 1 in a given position denoting the presence, and 0 denoting the absence, of the corresponding motif in the protein sequence in question). Each resulting binary pattern is labeled with the corresponding function (obtained from the annotation for the sequence). A subset of the resulting data set of binary patterns and associated function labels is provided as input to a decision tree learning algorithm to generate a decision tree for classifying proteins into functional families. The resulting decision tree represents in a compact form the presence and/or absence of specific motifs that are good predictors of protein function. The resulting decision tree is then evaluated on an independent test set. If it has satisfactory accuracy on the test set, it can be used to assign sequences with unknown function to one of the families. We then use additional programs to display the 3-dimensional structure of selected proteins. We visualize the motifs used by the decision tree to assign that protein to a functional class by overlaying the motifs at the corresponding positions on the sequence and 3-dimensional structure. The resulting visual representation helps biologists to explore the biological significance, if any, of the acquired knowledge. The preceding example illustrates how *implicit* ontologies of the scientists drive the information extraction and data integration procedures used in knowledge acquisition from data. For instance, the representation of protein sequences in terms of the motifs found in them is based on an (implicit) application of an ontology consisting of *part-whole* relationships between protein sequences and sequence motifs (ignoring in effect, the order in which motifs appear along the sequence). This scenario is typical of data analysis tasks encountered in computational biology. Ontologies thus provide an important (but often underutilized) source of *background knowledge* or assumptions that set up the *context* for data-driven knowledge discovery (Figure 4). Ontologies that take the form of taxonomies over attribute values can facilitate discovery of regularities from data at multiple levels of abstraction [Dhar and Tuzhilin, 1993; Taylor, Stoffel, and Hendler, 1997]. For instance, a taxonomy of functional families can be used to guide data-driven discovery of classifiers that assign protein sequences to functional families at different levels of abstraction. This underscores the need for integrating domain-specific as well as user-supplied (e.g., application specific) ontologies with distributed learning algorithms.

It underscores the need for selectively extracting and integrating heterogeneous data using user-supplied ontologies into a form that is expected by the applications that will process the data (e.g., decision tree learning algorithm described in the example above).

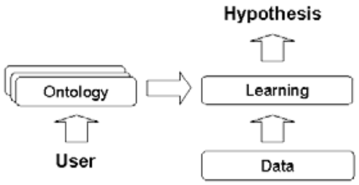

Figure 4: Ontology assisted knowledge acquisition

Integration of Distributed Heterogeneous Data

Approaches to processing heterogeneous data sources can be broadly classified into two categories: multi-database systems [Sheth, 1990; Bright et al., 1992; Barsalou and Gangopadhyay, 1992] and mediator systems [Wiederhold, 1997]. The multidatabase systems approach typically focuses on data integration across relational databases using relational or object-oriented *views* [Bertino, 1991; Yen et al., 1998; Miller et al., 1998] to provide integrated access to distributed databases. Database views provide a means of selectively presenting the relevant data from the users' perspective. For instance, a view can be defined across two data sets stored in different relational databases so that the user is able to interact with the data as though the data reside in a single (virtual) database. In this case, the relational view essentially executes a join operation on the two data sets, assuming that the terms in the two databases have the same syntax (names) and semantics (meaning). The basic multidatabase approaches to data integration have been developed to provide a data model and a common query language for information integration in the TSIMMIS project at Stanford University [Garcia-Molina et al., 1996; Chang and Garcia-Molina, 1999]. Similar techniques have been used for data integration from structured (but not necessarily relational) databases in the SIMS project [Arens et al., 1993] and data integration from unstructured web sources using wrappers in the Ariadne project [Knoblock et al., 2001] at the University of Southern California. Some important aspects of data integration have been formalized in the context integration of knowledge bases containing relational, spatial, and textual information in the Hermes project at the University of Maryland [Subrahmanian et al., 2000]. We have developed object-oriented data warehouses for information extraction and data integration from multiple relational or object-oriented databases [Honavar et al., 1998; Miller et al., 1998; Wu et al., 2000; Miller et al., 2000]. The resulting system allows users to interact with databases and generate, manipulate, and interact with *views*. Users can generate queries interactively using a Java interface. The resulting *object-oriented views* are represented as Java objects. The system supports adding and deleting *methods* associated with particular Java classes. Conversion between Java Classes and abstract data types defined by the database are automated by the system. The user is able to interact with the object-oriented view by executing the methods at run time.

Ontology Assisted Distributed Information Extraction and Data Integration

Data integration in computational biology requires a cascade of data transformations that bridge the syntactic and semantic gaps across data sources. Recently, we have developed a prototype system for integration of data from multiple biological data sources used in the protein function classification example described above. The resulting object-oriented data warehouse automates all of the information extraction and data transformations needed for constructing and using decision trees for protein function classification. The resulting system is able to extract the relevant data from Swissprot (www.expasy.ch/sprot/sprot-top.html) and Prosite (www.expasy.ch/prosite/) databases, transform the data into the desired form, and store it in local data warehouse for further analysis. Object-oriented views are particularly well-suited for information extraction from biological databases because the description of genomic, structural, functional, and organizational information involves many object types and relationships. One limitation of the current design is that the ontologies are not explicitly specified, but are implicitly hardwired into the procedures that transform the data. Work in progress is aimed at replacing the implicit ontologies with explicitly specified ontologies to design modular and customizable software that allows scientists to adapt and use ontologies of their choice to drive data transformation and data integration from heterogeneous distributed databases to enable rapid prototyping of systems solutions for scientific discovery through exploratory data analysis.

The development of algorithms and software for generating alternative representations of the data from different user-specified ontological perspectives is a key component of the proposed research. The preceding examples show that information extraction and data transformation operations involved in data-driven knowledge acquisition in computational biology can be decomposed into two components:

- *Ontologies which specify the relevant relationships among entities in our universe of discourse* (e.g., protein sequences, motifs, structures, functions). This component can be further partitioned into a database independent component (e.g., the ontology that captures the relevant part-whole relationships between protein sequences and motifs) and a database dependent component.
- *The processes that use this ontology to extract relevant information and knowledge from the available data sets*. These can be divided into database specific operations (e.g., querying a specific database using the interface provided), database independent (but ontology dependent) operations, and data structure dependent operations. An example of an ontology dependent operation is the transformation of a protein sequence into a representation that captures the presence or absence of particular motifs in a sequence. The resulting representation is constrained by the allowed data structures (e.g., bit patterns, lists, etc.) and the associated methods (procedures available to manipulate the data).

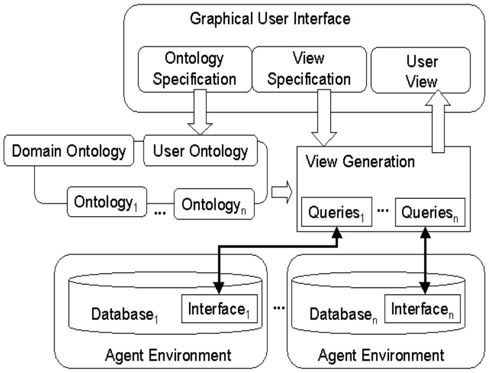

Figure 5: Ontology-based view generation from distributed databases. Ontology_I is an explicit specification of the possibly implicit ontology associated with Database_I with corresponding interface Interface_I. Extracted information is presented in User view.

Such decomposition allows us to develop tools that facilitate rapid design and prototyping of domain and application specific views over heterogeneous distributed data sources (Figure 5). Scientists will be able to adapt existing ontologies or define new ontologies to generate alternative representations (views) of data, at different levels of abstraction, in different contexts. Note that our goal is not to develop ontologies that exhaustively capture all of the relevant relationships in our universe of discourse (i.e., computational biology) but to demonstrate a theoretically well-founded modular approach to using user-supplied ontologies for information extraction and data integration.

Distributed and Incremental Data Mining Algorithms for Data Driven Knowledge Acquisition

The problem of learning from distributed data sets can be summarized as follows: the data are distributed across multiple sites and the learner's task is to discover useful knowledge. For example, such knowledge might be expressed in the form of a decision tree or a set of rules for pattern classification. As noted in the previous section, there are at least two ways in which the data set may be fragmented in a distributed setting [Sharma, et al., 2000; Caragea et al., 2001b]:

- *Horizontal fragmentation* wherein subsets of the data set are distributed across multiple sites; and
- *Vertical fragmentation*, wherein values for different subsets of attributes of the data set (perhaps gathered by different laboratories) are distributed across different sites (e.g., in independently maintained databases).

In relational databases, the data set is a set of tuples of attribute values. The preceding definitions of fragmentation of data sets can be extended in the case of other types of databases.

Given the large size and distributed and dynamic nature of data sets encountered in computational molecular biology, it is neither feasible nor desirable to exchange entire data sets among different sites. Consequently, the learner has to rely on the information extracted from the sites. One approach to learning from distributed data sources is to have a learning agent visit the different sites to gather the information it needs to generate the desired model from data. Alternatively, different sites can transmit the information necessary for inferring the model to the learning agent situated at a given location.

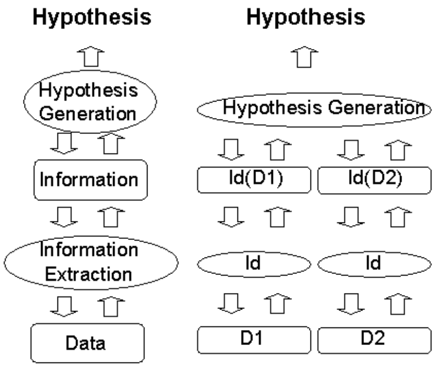

Figure 6: Decomposition of learning into information extraction and hypothesis generation components in the centralized (left) and distributed (right) scenarios.

A distributed learning algorithm L_D is said to be *exact* with respect to the hypothesis inferred by a learning algorithm L if the hypothesis produced by L_D using distributed data sets D_1 through D_n is the same as that obtained by L when it is given access to the complete data set D which can be constructed (in principle) by combining the individual data sets D_1 through D_n. For example, an exact distributed decision tree learning algorithm for vertically fragmented data is guaranteed to generate exactly the same decision tree in the distributed setting as it would when it is given access to the entire data set. Similarly, we can define exact distributed learning with respect to other criteria of interest (e.g., expected accuracy of the learned hypothesis). More generally, it is useful to consider *approximate* distributed learning in similar settings.

The incremental learning problem can be formulated as follows: the learner incrementally refines a hypothesis or a set of hypotheses as new data become available. Because of the large volume of data involved, it may not be practical to store and access the entire data set during learning. Thus, the learner may not have access to previously analyzed data (with the possible exception of a relatively small subset of critical examples stored by the learner). We assume that (sub) sets D_1 through D_n that correspond to horizontal or vertical fragments of a data set become available over discrete intervals in time. The learner starts with an initial hypothesis that constitutes prior knowledge of the domain. We assume that the learner has limited access to previously processed data in its raw form. Thus, the learner can maintain only the minimal information necessary for accurately updating its hypothesis as new data become available. Exactness of incremental learning can be defined in a manner analogous to that of distributed learning. More generally, it is useful to consider approximate incremental learning.

One approach to distributed learning is based on a decomposition of the learning task into information extraction and hypothesis generation components (Figure 6). This involves identifying the information requirements of the learning algorithm and designing efficient means of providing the needed information to the hypothesis generation component while avoiding the need to transmit large amounts of data. This offers a general strategy for transforming a batch learning algorithm (e.g., a traditional decision tree induction algorithm) into an exact distributed learning algorithm [Caragea et al., 2000; Caragea et al., 2001a]. Thus, we decompose the distributed learning task into distributed information extraction and hypothesis generation components (Figure 6). In this approach to distributed learning, only the information extraction component has to effectively cope with the distributed nature of data in order to guarantee provably exact learning in the distributed setting in the sense discussed above. Suppose we decompose a batch learning algorithm L in terms of an information extraction operator I that extracts the necessary information from data set and a hypothesis generation operator H that uses the extracted information to produce the output of the learning algorithm L. That is, $L(D) = H(I(D))$. We can define a distributed information extraction operator I_d that generates from each data set D_i, the corresponding information $I_d(D_i)$, and an operator C that combines the information obtained from the data sets to produce $I(D)$. That is, the information extracted from the distributed data sets is the same as that used by L to infer a hypothesis from the complete data set D. That is, $C[I_d(D_1), I_d(D_2), .. I_d(D_n)] = I(D)$. Thus, we can guarantee that L_d will be exact with respect to L. The feasibility of this approach to exact distributed learning depends on the information requirements of the batch algorithm L under consideration and the (time, memory, and communication) complexity of the corresponding distributed information extraction operations. We have used this approach of decomposing distributed learning into distributed information extraction and hypothesis generation to construct provably exact algorithms for decision tree learning from horizontally as well as vertically fragmented distributed data sets [Caragea et al., 2001b]. It turns out a large family of split criteria used to build decision trees, including information gain [Quinlan, 1986] and Gini index [Breiman et al., 1984] etc., can be expressed in terms of relative frequencies of instances that satisfy certain constraints on the values of their attributes. Indeed, in this case, these relative frequency estimates constitute *sufficient statistics* [Casella and Berger, 1990] for these split criteria [Caragea et al., 2001b]. (A sufficient statistic for a parameter θ (e.g., mean of a distribution), in a certain sense, captures all of the information about θ contained in the data set. For example, sample mean is a sufficient statistic for the mean of the distribution.).

We have shown that the information necessary for decision tree construction can be efficiently obtained from horizontally or vertically fragmented distributed data sets, thereby yielding provably exact algorithms for decision tree induction from horizontally or vertically fragmented distributed data sets. This approach to learning decision trees from distributed data based on a decomposition of the learning task into a distributed information extraction component and a hypothesis generation components provides an effective way to deal with scenarios in which the sites provide only statistical summaries of the data on demand and prohibit access to raw data. Even when it is possible to access the raw data, the distributed algorithm compares favorably with the corresponding centralized algorithm which needs access to the entire data set whenever its communication cost is less than the cost of collecting all of the data in a central location. Let $|D|$ be the total number of examples in the distributed data set; $|A|$, the number of attributes; V the maximum number of possible values per attribute; n the number of sites across which the data set D is distributed; M the number of classes; and size(T) the number of nodes in the decision tree. Our analysis has shown that in the case of horizontally fragmented data, the distributed algorithm has an advantage when MVn size(T) < |D|. In the case of vertically fragmented data, the corresponding conditions are given by size(T) < |A|. The distributed decision tree learning algorithms have been implemented in our laboratory using mobile software agents [Sharma et al., 2000; Caragea et al., 2001b]. Our experiments have shown that these conditions are often met in the case of large, high-dimensional data sets that are encountered in computational biology (e.g., construction of decision trees for classification of protein sequences into functional families) [Wang et al., 2001]. Work in progress focuses on techniques which trade off exactness of the algorithm (and hence possibly the accuracy of the learned model) for increased efficiency (in terms of computation and communication costs).

Space does not permit a detailed discussion of our results on incremental learning. Hence, we simply note some of our results on incremental learning. We have designed and implemented a semi-incremental algorithm [Polikar et al., 2000] for training neural network classifiers based on recent computational learning theoretic results on *accuracy boo*sting [Schapire, 1999; Freund and Schapire, 1997]. The resulting algorithm has been successfully applied to train an *electronic nose* for odor recognition [Polikar et al., 2001].

The results summarized above suggest steps toward the development of a framework for distributed and incremental learning. Work in progress is aimed at generalizing the treatment of distributed learning outlined above by introducing a family of learning and information extraction and information composition operators and establishing sufficient conditions for provably exact (and in some cases, approximate) distributed or incremental learning in terms of general algebraic properties of the operators [Caragea et al., 2001a]. This framework provides a means of unifying a diverse body of recent work related to:

- Distributed learning approaches based on combining multiple models learned from disjoint data sets [Davies et al., 2000; Prodromidis et al., 2000]; parallel formulations of decision tree learning algorithms [Srivastava et al., 1999]; techniques for scaling up distributed learning algorithms [Provost et al., 1999]; collective data mining algorithms [Kargupta et al., 2000]; and the computation of *sufficient statistics* [Casella and Berger, 1990; Moore and Lee, 1997].
- Incremental learning including approaches based on *online* algorithms developed within the *mistake bound* model [Littlestone, 1994], and *lazy learning* e.g., nearest neighbor techniques, and locally weighted regression [Atkeson et al., 1997] and accuracy boosting [Freund and Schapire,1997].

This approach to distributed learning and incremental learning through a decomposition of distributed (or incremental) learning into distributed (or incremental) information extraction interleaved with hypothesis generation provides an attractive approach to developing data-driven knowledge acquisition learning algorithms in the distributed (or incremental) setting. It should be noted that a large body of theoretical results (e.g., sample complexity, error bounds, etc.) derived in the batch learning scenario carry over naturally to incremental and distributed learning scenarios if we can establish that the distributed (or incremental) information extraction component provides the same information to the hypothesis generation component as that available in the batch setting. The boundary that defines the division of labor between information extraction and hypothesis generation components depends on the hypothesis class used for learning, and decomposition used. At one extreme, if no information extraction is performed, the hypothesis generation component needs to access the raw data. An example of this scenario is provided by distributed instance based learning of nearest neighbor classifiers from a horizontally fragmented data set. Here, the data set fragments are simply stored at the different sites. Classification of a new instance is performed by the hypothesis generation component which classifies a new instance according to the classification assigned to the nearest neighbor of the instance to be classified (based on some specified metric which measures the similarity between any two instances). At the other extreme, if the information extraction component does most of the work, and the task of the hypothesis generation component becomes trivial. This argues for a more systematic exploration of the design space of distributed learning algorithms. Hence, work in progess is aimed at building on these results to design and analyze distributed and incremental learning algorithms and software with emphasis on data-driven knowledge acquisition from large, dynamic, high-dimensional biological data sets (DNA sequence data, protein sequence data, protein structures, gene expression data).

Summary

The recent advances in high throughput data acquisition technologies, coupled with advances in computing, digital storage and communication technologies, presents unprecedented challenges as well as opportunities in large-scale data-driven knowledge acquisition in several domains. Examples of such domains include biological sciences, atmospheric sciences, medical sciences, and social sciences, among others. In this paper, we have described some of the problems that need to be addressed in order to translate the tremendous advances in our ability to gather and store data in increasing volumes at rapidly increasing rates into fundamental scientific advances and economic and social benefits through data-driven discovery. We have presented some of the key elements of algorithmic and systems solutions for computer assisted knowledge acquisition. These include: ontology-assisted approaches to *customizable* data integration and information extraction from heterogeneous, distributed data sources; *distributed* data mining algorithms for knowledge acquisition from large, distributed data sets which obviate the need for transmitting large volumes of data across the network; ontology-driven approaches to *exploratory data analysis* from alternative ontological perspectives; and modular and extensible agent-based implementations of the algorithms within a platform-independent agent infrastructure. Prototype systems that incorporate the key components of proposed solution are being used for discovery of macromolecular structure function relationships in biological sciences and coordinated intrusion detection in distributed computing and communication networks. We conclude with a brief mention of some important current and future research directions:

- Further development of the prototype systems and their application to Knowledge acquisition problems in computational biology including: Problems which involve the analysis of a single element (e.g., gene, protein) or a small set of elements that vary across different organisms and problems in which information to be analyzed reflects interactions among diverse elements within a single organism
- Development of a small set of data source independent ontologies (including not only the terms and relations among terms, but also the associated functions for data transformation) to support information extraction and exploratory data analysis in the context of representative problems in computational biology that are identified in section B-III. The resulting toolbox of ontologies will include generic data transformation procedures for handling common types of ontologies such as taxonomies, phylogenetic trees, part-whole relations, among others.
- Development of a set of data source specific ontologies to support information extraction from a set of biological databases (e.g., the Swissprot protein database, some common motif databases, and sequence databases. This will build on a subset of the ontologies that are being developed as part of the gene ontology (GO) project (www.geneontology.org).

- Specification, design, and implemention of a software tool that allows biologists to create and edit ontologies (including the associated computational procedures for ontology-driven transformation of data) that would then be used to drive information extraction and data transformation operations in specific contexts.
- Specification, design, and implemention of software tools that would enable biologists to interactively select data sources, ontologies, and data analysis and visualization programs (e.g., decision tree learning algorithm) using a graphical user interface to instantiate specific data analysis procedures (e.g., building motif-based classifiers of protein function). The system will also include pre-specified views of the data to provide some of the most useful representations of the data.
- Development of an object-oriented data warehouse [Miller et al., 1998; Miller et al., 2000; Wu et al., 2001; Silvescu et al., 2001b] to support applications that require local storage of information extracted from remote data sources for further analysis. The data warehouse essentially maintains materialized (instantiated) views across multiple data sources to provide rapid access to pre-integrated data in a desired form. It also supports analysis of data from data sources that do not provide facilities for execution of user-supplied analysis programs. In such cases, setting up data servers that store the relevant data in an integrated form for specific communities of users and computation servers that are tightly coupled to such data servers would significantly increase the utility of such data sources.
- Development of approaches to perform ontology-driven information extraction from multiple databases [Honavar et al, 1998; White, 1997] where the desired views are computed in a distributed fashion (as an integral part of information extraction data mining from distributed data sets) without gathering all of the data at a central location.
- Exploration of the design space of exact and approximate distributed learning algorithms based on decomposition of distributed learning into distributed information extraction and hypothesis generation components for variety of hypothesis classes for classification and function approximation. This will include an examination of alternative decompositions resulting in different ways of dividing the work between information extraction and hypothesis generation components and distributed learning algorithms.
- Implementation of a modular and extensible and platform independent agent-based system for distributed data mining for execution within a loosely coupled distributed environment. This builds on our ongoing work on distributed software environments for data integration and data mining from distributed data sets [Honavar et al., 1998; Sharma, 2000; Caragea et al., 1991b; Honavar et al., 2001].
- Exploration of the design space of exact and approximate incremental learning algorithms based on decomposition of incremental learning into incremental information extraction and hypothesis generation components for variety of hypothesis classes for classification and function approximation.
- Further elucidation of the necessary and sufficient conditions that guarantee the existence of efficient exact, and approximate distributed and incremental learning algorithms in terms of properties of hypothesis and data representations and available learning operators (e.g., operators for generating hypotheses from data; operators for combining multiple hypotheses, etc.).
- Integration of distributed and incremental learning with visualization algorithms to facilitate interactive exploratory analysis of large, distributed, high-dimensional biological data sets, with emphasis on techniques for data integration from diverse data sets, dimensionality reduction, automated feature selection and construction, identification of outliers and departures from trends, and knowledge visualization.
- Development of techniques for incorporating ontologies (in particular, *part-whole hierarchies* and *taxonomies*) with data integration and data mining algorithms to support data-driven knowledge acquisition at multiple levels of abstraction from large, distributed, heterogeneous biological data sets.

Bibliography

1. Arens,Y., Chee, C., Hsu, C., and Knoblock, C. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Intelligent and Cooperative Information Systems*. Vol. 2, No. 2. Pp. 127-158, 1993.
2. Ashburner et al. (2000) Gene Ontology: Tool for the Unification of Biology. Nature Genetics. Vol. 25. No. 1. pp. 25-29.
3. Atkeson, C.G., Schaal, A. and Moore, A. (1997) Locally weighted regression AI Review, Vol. 11, pp. 11-7 1997
4. Baldi, P.and Brunak, S. (1998). Bioinformatics: The Machine Learning Approach. Cambridge, MA: MIT Press.
5. Barsalou, T. and D. Gangopadhyay. (1992). M(DM): An open framework for interoperation of Multimodel Multidatabase Systems. IEEE Data Engineering pp. 218-227.
6. Baxevanis, A. and Ouellette, B.F.F. (ed.) (1998). Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. New York: Wiley.
7. Baxevanis, A.D. (2000). The Molecular Biology Database Collection: an online compilation of relevant database resources. Nucleic Acids Research 28(1): pp. 1-7.
8. Benson, D.A., Boguski, M.S., Lipman, D.J. and Ostell, J. (1997). GenBank. Nucleic Acids Research 25(1): pp. 1-6.
9. Bertino, E. (1991). Integration of heterogeneous data repositories by using object-oriented views. 1st Int. Workshop on Interoperability in Multidatabase Systems: pp. 22-29.

10. Boguski, M.S., Lowe, T.M.J. and Tolstoshev, C.M. (1993). Dbest - database for expressed sequence tags. Nature Genetics 4(4): pp. 332-333.
11. Breiman, L. (1984). *Classification and Regression Trees*. New York: CRC Press.
12. Bright, M.W., A.R. Hurson and S.H. Pakzad. (1992). A taxonomy and current issues in multidatabase systems. Computer. Vol. 25. No. 3. pp. 50-60.
13. Caragea, D., Silvescu, A., and Honavar, V. (2000). Towards a Theoretical Framework for Analysis and Synthesis of Distributed, Parallel, and Incremental Learning Algorithms. In: Proceedings of the KDD 2000 Workshop on Distributed and Parallel Knowledge Discovery. Boston, MA.
14. Caragea, D., Silvescu, A., and Honavar, V. (2001a). Analysis and Synthesis of Agents that Learn from Distributed Dynamic Data Sources. Invited chapter. In*:* Wermter, S., Willshaw, D., and Austin, J. (Ed.). *Emerging Neural Architectures Based on Neuroscience.* Springer-Verlag. In press.
15. Caragea, D., Silvescu, A., Andorf, C., Reinoso-Castillo, J. Honavar, V. (2001b). Learning Decision Tree Classifiers from Distributed Databases. Tech report ISU-CS-TR 2001-09. Department of Computer Science, Iowa State University, Ames, Iowa 50011, USA. Submitted for conference publication.
16. Casella, G., and Berger, R. (1990) *Statistical Inference*. Belmont, CA: Duxbury Press. pp. 247-264.
17. Chang, C.-C.K. and Garcia-Molina, H. (1999). Mind your Vocabulary: Query Mapping across Heterogeneous Information Sources. ACM SIGMOD Int. Conference on Management of Data 1999, Philadelphia: pp. 335-346.
18. Discala, C., Benigni, X., Barillot, E. and Vaysseix, G. (2000). DBcat: a catalog of 500 biological databases. Nucleic Acids Research 28(1): pp. 8-9.
19. Durbin, R. and Mieg, J. (1991) A *C.elegans* Database. www.sanger.ac.uk/Software/Acedb/
20. Fasman, K. (1994). Restructuring the Genome Data Base: A model for a federation of biological databases. Journal of Computational Biology **1**(2): pp. 165-171.
21. Frenkel, K.A. (1991). The Human Genome Project and Informatics. Communications of the ACM 34(11): pp. 40-51.
22. Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer Science and System Sciences*, 55(1):119--139, 1997.
23. Gelbart, W.M. (1998). Databases in Genomic Research. Science **282**: pp. 659-661.
24. Garcia-Molina , Y. Papakonstantinou , D. Quass , A. Rajaraman , Y. Sagiv , J. Ullman , V. Vassalos , J. Widom (1996). The TSIMMIS approach to mediation: Data models and Languages. *Journal of Intelligent Information Systems*.
25. Ghosh, D. Object-oriented Transcription Factors Database ooTFD: An Object-oriented successor to TFD. *Nucleic Acids Research* 26 pp. 360-362.
26. Goodman, N. (1995). An object-oriented DBMS War Story: Developing a Genome Mapping Database in C++. In: *Modern Database Systems*, W. Kim (Ed.), Reading, MA: Addison-Wesley.
27. Gray, P., Paton, N., Kemp, G., Fothergill, J. (1990) An object-oriented database for protein structure analysis. *Protein Engineering* 4-3 pp. 235-243.
28. Hendler, J, Stoffel, K., and Taylor, M. Advances in High Performance Knowledge Representation. University of Maryland Institute for Advanced Computer Studies Dept. of Computer Science, Univ. of Maryland, July 1996. CS-TR-3672 (Also cross-referenced as UMIACS-TR-96-56)
29. Honavar, V., Miller, L., and Wong, J. Distributed knowledge networks. In: Proceedings of IEEE Information Technology Conference. pp. 87--90, IEEE Press, 1998.
30. Honavar, V., Andorf, C., Caragea, D., Silvescu, A., and Sharma, T. (2001). Agent-Based Systems for Data-Driven Knowledge Discovery from Distributed Data Sources: From Specification to Implementation. In: *Intelligent Agent Software Engineering*. Plekhanova, V., and Wermter, S. (Ed.), London: Idea Group Publishing. To appear.
31. Jennings, N.R., Wooldridge, M.J. (2000). Agent-Oriented Software Engineering in: J. Bradshaw (ed.), Handbook of Agent Technology, AAAI/MIT Press, 2000.
32. Kargupta, H., Park, B., Hershberger, D., and Johnson, E. (2000) Collective Data Mining: A New Perspective Toward Distributed Data Mining. . In: *Advances in Distributed and Parallel Knowledge Discover*y. Kargupta, H., and Chan, P. (Ed). Cambridge, MA: MIT Press.
33. Karp, P.D. (1996). A Strategy for Database Interoperation. Journal of Computational Biology 2(4): pp. 573-583.
34. Karp, P.D**.**, (2000) An ontology for biological function based on molecular interactions. Bioinformatics 16(3):269-85.
35. Knoblock, C., Minton, S., Ambite, J., Ashish, N., Muslea, I., Philpot, A., and Tejada, S. (2001). The Ariadne approach to Web-based information integration. *International Journal on Cooperative Information Systems.* Forthcoming.
36. Littlestone, N. (1994). The weighted majority algorithm. *Information and Computation*, 108:212-261, 1994.
37. Miller, L., Honavar, V. and Wong, J. (1998). Object-oriented data warehouses for information fusion from heterogeneous distributed data and knowledge sources. In Proceedings of the IEEE Information Technology Conference, Syracuse, NY. IEEE Press. 1998.
38. Miller, L.L., Lu, Y, Zhou, Y. Hurson, A.R. (2000). A Data Warehouse Based on Materializing Object-Oriented Views. *International Conference on Computers and Their Applications*. New Orleans, LA. pp. 68-71.
39. Mitchell, T. (1997). Machine Learning. New York: Addison-Wesley.

40. Moore, A.W. and Lee, M.S. (1997). Cached Sufficient Statistics for Efficient Machine Learning with Large Data Sets. *Journal of Artificial Intelligence Research*, Vol. 8., pp 67-91.
41. Polikar, R., Udpa, L., Udpa, S., and Honavar, V. (2000). Learn++: An Incremental Learning Algorithm for Multilayer Perceptron Networks. In: Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2000. Istanbul, Turkey.
42. Polikar, R., Shinar, R., Honavar, V., Udpa, L., and Porter, M. Detection and Identification of Odorants Using an Electronic Nose. In: Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2001. Salt Lake City, Utah, USA.
43. Prodromidis, A., Chan, P., and Stolfo, S. (2000). Meta-learning in distributed data mining systems: Issues and Approaches. In: *Advances in Distributed and Parallel Knowledge Discover*y. Kargupta, H., and Chan, P. (Ed). Cambridge, MA: MIT Press.
44. Provost, F. and V. Kolluri, A Survey of Methods for Scaling Up Inductive Algorithms. *Data Mining and Knowledge Discovery* 3 (1999).
45. Quinlan, J. (1986). Induction of Decision Trees. *Machine Learning* vol. 1, pp 81-106, 1986.
46. Sharma, T., Silvescu, A., Andorf, C., Caragea, D., and Honavar, V. (2000). Learning Classification Trees from Distributed, Horizontally and Vertically Fragmented Data. Technical Report 00-10. Department of Computer Science. Iowa State University.
47. Schulze-Kremer, S. (1997). Adding Semantics to Genome Databases: Towards an Ontology for Molecular Biology. 5th Int. Conf. on Intelligent Systems for Molecular Biology, Halkidiki, Greece, AAAI Press, Menlo Park: pp. 272-275.
48. Schapire, R.E. (1999). A brief introduction to boosting. In Proceedings of International Joint Conference on Artificial Intelligence, pages 1401--1405, 1999.
49. Sowa, J. (1999) Knowledge Representation: Logical, Philosophical, and Computational Foundations. New York: PWS Publishing Co.
50. Thrun, S., Faloutsos, C., Mitchell, M., Wasserman, L. (1999). Automated Learning and Discovery: State-of-the-art and research topics in a rapidly growing field. *AI Magazine*, 1999.
51. Uschold, M. and Jasper, R. (1999). A Framework for Understanding and Classifying Ontology Applications. In: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods. V.R. Benjamins, B. Chandrasekaran, A. Gomez-Perez, N. Guarino, and M. Uschold (Ed.).
52. Wiederhold, G. and M. Genesereth (1997) The Conceptual Basis for Mediation Services, IEEE Expert, Vol.12 No.5 pp. 38-47.
53. Wang, D., Wang, X., Honavar, V., and Dobbs, D. (2001). Data-Driven Generation of Decision Trees for Motif-Based Assignment of Protein Sequences to Functional Families. In: Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems & Technology. In press.
54. White, J.E. (1997). Mobile Agents. In: Bradshaw, J.M. (ed.), Software Agents, Cambridge, MA: MIT Press.
55. Wong, J., Helmer, G., Honavar, V., Naganathan, V., Polavarapu, S., and Miller, L. SMART Mobile Agent Facility. Journal of Systems and Software. Vol. 56. pp. 9-22.
56. Wu, L., L.L. Miller, and S. Nilakanta. (2001). Design of Data Warehouses using Metadata. To appear in *Information and Software Technology*.
57. Yen, C.H., L.L. Miller, A. Sirjani and J. Tenner (1998) Extending the object-relational interface to support an extensible view system for multidatabase integration and interoperation. International Journal of Computer Systems Science and Engineering. Vol. 13 pp. 227-240.

.