Mining Scientific Data*

Naren Ramakrishnan
Department of Computer Science
Virginia Tech, VA 24061
Tel: (540) 231-8451
Email: naren@cs.vt.edu

Ananth Y. Grama
Department of Computer Sciences
Purdue University, IN 47907
Tel: (765) 494-6964
Email: ayg@cs.purdue.edu

Abstract

The past two decades have seen rapid advances in high performance computing and tools for data acquisition in a variety of scientific domains. Coupled with the availability of massive storage systems and fast networking technology to manage and assimilate data, these have given a significant impetus to data mining in the scientific domain. Data mining is now recognized as a key computational technology, supporting traditional analysis, visualization, and design tasks. Diverse applications in domains such as mineral prospecting, computer aided design, bioinformatics, and computational steering are now being viewed in the data mining framework. This has led to a very effective cross-fertilization of computational techniques from both continuous and discrete perspectives. In this chapter, we characterize the nature of scientific data mining activities and identify dominant recurring themes. We discuss algorithms, techniques, and methodologies for their effective application and present application studies that summarize the state-of-the-art in this emerging field. We conclude by identifying opportunities for future research in emerging domains.

*This work was supported in part by National Science Foundation grants EIA-9974956 and EIA-9984317 to Ramakrishnan and National Science Foundation grants EIA-9806741, ACI-9875899, and ACI-9872101 to Grama.

Contents

1 Introduction

Computational simulation and data acquisition in scientific and engineering domains have made tremendous progress over the past two decades. A mix of advanced algorithms, exponentially increasing computing power, and accurate sensing and measurement devices have resulted in terabyte-scale data repositories. Advances in networking technology have enabled communication of large volumes of data across geographically distant hosts. This has resulted in an increasing need for tools and techniques for effectively analyzing scientific datasets with the objective of interpreting the underlying physical phenomena. The process of detecting higher order relationships hidden in large and often noisy datasets is commonly referred to as *data mining*.

The field of data mining has evolved from its roots in databases, statistics, artificial intelligence, information theory, and algorithmics into a core set of techniques that have been successfully applied to a range of problems. These techniques have come to rely heavily on numerical methods, distributed and parallel frameworks, and visualization systems to aid the mining process. In this chapter, we characterize the nature of scientific data mining activities and identify dominant recurring themes. We discuss algorithms, techniques, and methodologies for their effective application, and present application studies that summarize the state-of-the-art in this emerging field.

The diverse and rich background of data mining is reflected in the many distinct views taken by researchers of this evolving discipline. Some of these dominant views are summarized here:

Mining is Induction

Perhaps the most common view of data mining is one of induction *i.e.*, proceeding from the specific to the general. This basis finds its roots in the artificial intelligence and machine learning communities and relies on techniques ranging from neural networks [Fu, 1999] to inductive logic programming [Muggleton, 1999]. Systems such as PROGOL (not PROLOG), FOIL, and Golem view induction as reversing the deduction process in first-order logic inference. Barring any specific constraints on the nature of induced generalizations, one way to generate (mine) them is to systematically explore and analyze a space of possible *patterns*. This idea of 'generalization as search' was first proposed by Mitchell [Mitchell, 1982] and has since resurfaced in various domains, most notably the association-rules mining algorithm of [Agrawal et al., 1993] used in commercial market basket analysis. The key idea in this research is to use partial orderings induced by subsumption relations on a concept space to help prune the search for possible hypotheses. Areas that deal most directly with this perspective include Sections 3 and 5.

Mining is Compression

The process of learning a *concept* from an enumeration of training sets typically leads to a very large space of possible alternatives. Often, the most desirable of these concepts is the one that is most succinct or that is easiest to describe. This principle, known as Occam's Razor, effectively equates learning to compression (where the learned patterns are, in some sense, "smaller to describe" than exhaustively enumerating the original data itself). The emergence of computational learning theory in the 1980s and the effectiveness of models such as MDL (the Minimum Description Length principle) have provided a solid theoretical foundation to this perspective. Several commercial data mining systems employ this aspect to determine the feasibility of the mined patterns. For example, if the description of a defective machine part is longer than an enumeration of possible defective parts, then the learned concept is not very useful.

Mining is Querying

The view of mining as intelligently querying a dataset was first projected in the database systems community. Since most commercial and business data resides in industrial databases and warehouses, mining is often viewed as a sophisticated form of database querying. SQL, the de-facto standard for database query languages, currently, does not support queries of the form "find me all machine parts in the database that are defective." The querying aspect allows embedding of mining functions as primitives into standard query languages. As more scientific and engineering data finds its way into databases and XML formats, this form of querying will become more important. It is conceivable that systems might support standard queries of the form 'highlight all vortices within specified flow data' in the near future. This perspective is currently most popular in the commercial and economic mining community. Its implications for mining scientific data are discussed in Section 5.

Mining is Approximation

Mining is often also thought of as a process of systematic approximation. Starting from an exact (lossless) data representation, approximations are introduced in the hope of finding some latent/hidden structure to the data. Such approximations might involve dropping higher-order terms in harmonic approximations, adaptive simplification of geometries, or rank reduction in attribute matrices. A very popular technique that has this flavor is called Latent Semantic Indexing [Berry et al., 1999, Berry et al., 1995, Berry and Fierro, 1996, Jiang et al., 1999, Letsche and Berry, 1997], an algorithm that introduces approximations using singular value decompositions of a term-document matrix in information retrieval to find hidden structure. This has parallels in Karhunen-Loeve expansions in signal representation and principal component analysis in statistics. A surview of such *data reduction techniques* appears in [Barbara et al., 1997]. Section 4 most directly deals with this perspective.

2 Motivating Domains

As the size and complexity of datasets gathered from large scale simulations and high resolution observations increases, there is a significant push towards developing tools for interpreting these datasets. In spite of its relative infancy, the field of scientific data mining has been applied with significant success in a number of areas. In this section, we outline some key application areas and the data mining tasks within these with a view to motivating the core techniques underlying common data mining tasks. The survey presented is by no means comprehensive. Other applications abound in areas such as computational biology, scientific visualization, robotics, and wireless communications.

2.1 Data Mining in Geological and Geophysical Applications

Data mining applications in geology and geophysics were among the first to be pursued and have achieved significant success in such areas as weather prediction, mineral prospecting, ecological modeling, landuse analysis, and predicting earthquakes from satellite maps. An interesting aspect of many of these applications is that they combine spatial and temporal aspects, both in the data as well as the phenomena being mined.

Datasets in these applications come both from observations and simulation. A prototypical example from weather forecasting relies on data that is typically defined over a grid of observation stations or finite difference simulation models. The underlying variables defined over these discretizations include horizontal velocities, temperature, water vapor and ozone mixing ratio, surface pressure, ground temperature, vertical velocities, precipitation, cloudiness, surface fluxes of sensible and latent heat, surface wind stress, and relative heating. Such data is typically captured at periodic snapshots. For example, in the Atmospheric Global Circulation Model (AGCM), at the lowest grid resolution (4 degree latitude × 5 degree longitude × 9 atmospheric levels), storing data at periods of 12 hours results in over 5 GB of data per year.

Data analysis proceeds with respect to a variety of atmospheric phenomena, such as cyclones, hurricanes, and fronts. The key challenge in analyzing atmospheric data is the imprecise definition of weather phenomena. For example, cyclones are typically defined based on a threshold level of vorticity in quantities such as atmospheric pressure at sea level. Other techniques for detecting cyclones rely on determination of local extrema of the sea level pressure [Huang and Zhao, 1998]. The outcome of an automated cyclone detection technique is a one-dimensional track in three dimensional space consisting of two space coordinates and one time coordinate. Other weather-related phenomena such as *blocking features* can also be detected using similar techniques. Blocking features are characterized by well defined time-scales (in the range of weeks) with a westerly flow that is split into two branches. These persistent anomalies are measured by examining the variation of geopotential height from the expected value.

The spatio-temporal nature of most of the analysis tasks makes them extremely challenging. The temporal aspect is typically modeled by characterizing the short-range and long-range dependence of processes on various forms of time series analysis. Spatial aspects are modeled using point-process techniques, particularly in weather-related phenomena. For example, storms are viewed as an agglomerate of rain-producing cells distributed over the area within which rain is occurring. These cells are assumed to be stationary and to be distributed in space either independently according to a Poisson process, or with clustering according to, say, a Neymann-Scott scheme. In addition, fits to empirical data are achieved by including random variables in the problem formulation to correlate between the durations of cells within a single storm.

The modeling, detection, and prediction of global climate phenomena has received much attention in the high performance computing community [Semtner, 2000]. Such applications involve multiple models at varying levels of fidelity (and often multi-disciplinary), parallel computing, and robust methodologies for ensuring that predictions agree with observed behavior. The extreme sensitivity of such phenomena (folklore has that, possibly incorrectly, ‘a bird flapping its wings in Switzerland can cause thunderstorms in the United States’) and interactions between ocean and atmospheric processes make this problem a promising

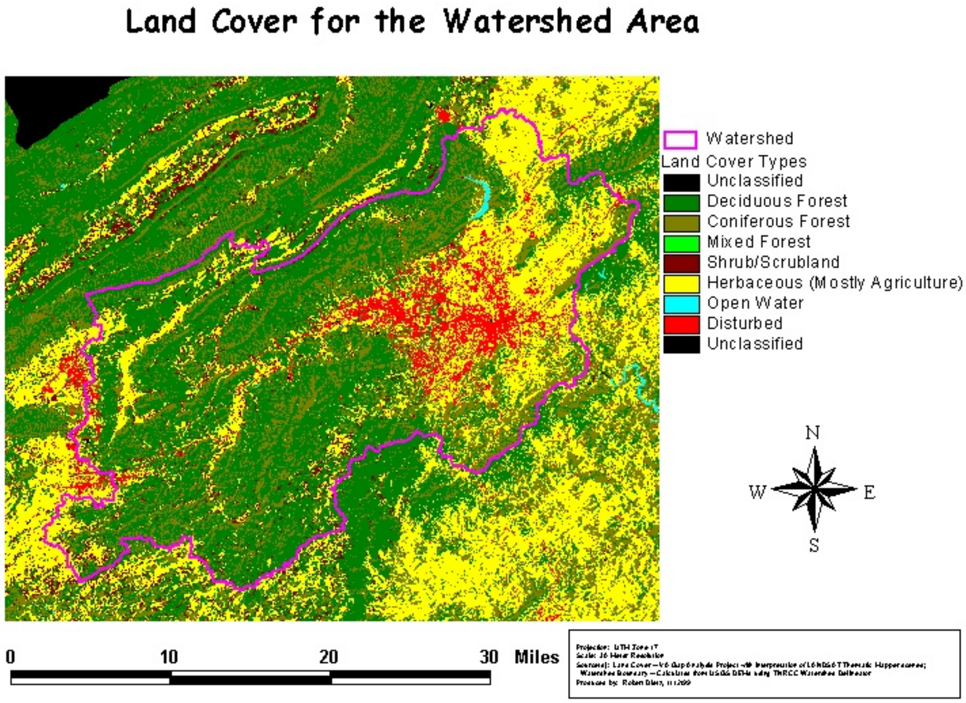

Land Cover for the Watershed Area
Watershed
Land Cover Types
Unclassified
Deciduous Forest
Coniferous Forest
Mixed Forest
Shrub/Scrubland
Herbaceous (Mostly Agriculture)
Open Water
Disturbed
Unclassified
N
W
E
S
0
10
20
30
Miles
Projection: UTM Zone 17
Scale: 30 Meter Resolution
Source(s): Land Cover—VA Gap Analysis Project with Interpretation of LANDSAT Thematic Mapper scenes;
Watershed Boundary—Calculated from USGS DEMs using TNRCC Watershed Delineator
Produced by: Robert Dietz, 11/2/99

Figure 1: Landuse segmentation of the Upper Roanoke River Watershed in Southwest Virginia, USA. Areas marked ‘Unclassified’ and ‘Mixed Forest’ pose difficulties for evaluating the effect of settlement patterns. Data mining can help identify mislabeled areas and correlate unlabeled regions with known land cover features. Figure courtesy R. Dymond (Virginia Tech).

application area for data mining. Effects ranging from warming of the oceans (e.g. El Niño) to the compensatory behavior of monsoons in tropical areas (e.g. the Southwest and Northeast monsoons in the Indian subcontinent) are important in assessing chances of drought, harsh winters, flooding, and in the management of water resources.

One of the early successful mining systems for scientific data was CONQUEST (CONcurrent Querying in Space and time) [Stolorz et al., 1995]. This system was designed to handle datasets with significant temporal coherence in events and to enable complex multimodal interactive querying and knowledge discovery. Han et al. [Koperski et al., 1998] provide a survey of techniques for mining geographical data, with specific emphasis on database perspectives and clustering, including their CLARANS algorithm for spatial mining [Ng and Han, 1994].

Rapid advances in remote sensing have also resulted in large repositories of high-resolution imagery that pose significant challenges for data handling and analysis. One of the analysis tasks that has received some attention is the prediction of the motion of surface faults during an earthquake [Preston et al., 2000]. QUAKEFINDER [Stolorz et al., 2000] is one such system that applies machine learning techniques to the measurement of plate displacements during earthquakes by comparing successive satellite images of a fault-laden area. Similar approaches are being applied to a number of applications relating to monitoring continuous and abrupt tectonic activity, land cover dynamics, and global climate changes.

In addition, datasets from remote sensing are typically characterized by mislabeled or unavailable information (see Fig. 1). While this problem is endemic to many experimental disciplines, it causes particular hardship for applications such as watershed assessment, where landuse distributions and settlement patterns are important drivers of change [Rubin et al., 2000]. They affect surface and groundwater flows, water quality, wildlife habitat, economic value of the land and infrastructure (directly due to the change itself such as building a housing development, and indirectly due to the effects of the change, such as increased flooding), and cause economic effects on municipalities (taxes raised versus services provided). Modeling such effects in a system requires the accurate and effective determination of landuse classifications; however, out-of-date field measurements, lack of knowledge of precise commercial and vegetation boundaries often result in mislabeled training data [Brodley and Friedl, 1999], thus posing bottlenecks in data mining. The more broader task of map analysis using geographical information system (GIS) data is important in identifying clusters of wild life behavior in forests [Berry et al., 1994], modeling population dynamics in ecosystems [Abbott et al., 1997], and socio-economic modeling [Berry et al., 1996].

An important application of data mining in geophysics relates to the detection of subsurface mineral deposits, oil reservoirs, and other artifacts. Typical datasets in these applications are collected from excitation and observation stations in borewells. For example, an agent is injected into one of the borewells and measurements are made at nearby observation bores. Such measurements are typically taken at regular intervals (and thus the temporal nature of data) over pre-specified observation points. Carefully examining this data for artifacts reveals presence (or absence) of phenomena being studied. An approach to mining porosity of prospect regions by automatically deriving analytic formulae is described in [Li and Biswas, 1995]. Such applications are characterized by great economic importance, and the accompanying need to ensure privacy and confidentiality in data mining. Other computational aspects of earth systems science can be found in the May-June 2000 Special Issue of IEEE/AIP *CiSE* on this topic [Rundle, 2000].

2.2 Data Mining in Astrophysics and Cosmology

The recent past has seen an explosion in the amount of data available to astrophysicists for analyzing a wide variety of phenomena. This data holds the key to such fundamental questions as the origins of the universe, its evolution, the structures that lie deep within, and the presence of extra-terrestrial lifeforms. A number of researchers are actively working on various high-profile projects relating to analysis of astrophysical data.

The main source of astrophysical data is in the form of surveys of the sky in different segments of the electromagnetic spectrum. We are rapidly getting to the point where surveys are generating more data than can be assimilated by semi-automated means. SKICAT (Sky Image Classification and Archiving Tool) was one of the early systems that recognized the need for automated analysis of large-scale sky surveys [Fayyad et al., 1993, Weir et al., 1995]. SKICAT was also instrumental in popularizing data mining in the scientific community. The goal was to provide an automated means for analyzing data from the Palomar Observatory Sky Survey (POSS-II), which consists of approximately 107 galaxies and 10^8 stars. The magnitude of this data clearly precludes manual analysis. The SKICAT system attempted to address the basic question of determining which of the objects in the survey belong to various classes of galaxies and stars. The system extracts a variety of features derived from image processing techniques and uses a tree classifier to classify celestial objects into one of several known classes. More recently the Sloan Digital Sky Survey [Szalay, 1999], on the order of terabytes, is expected to become the benchmark reference for knowledge discovery in computational cosmology. See [Szalay, 1999] and other articles in the March-April 1999 special issue of IEEE/AIP *CiSE* [Tohline and Bryan, 1999] for more information on such projects.

One of the key sources of data relating to the origin of the universe is believed to be cosmic background radiation (CMB). This radiation provides an excellent measure of the inhomogeneities in the distribution of matter in the universe over a period of billions of years. Fluctuations in photon intensity indicate the variations in density and velocity of radiation at a point when the universe was highly ionized. They also provide information about the amount of gravitational clustering during different epochs in the universe's history through which the photons pass [Hu, 1996]. Telescopes such as Hubble and Keck have made visible galaxies that are at much earlier stages of their evolution than the Mikly Way. This will potentially enable us to chart our own evolution if we can effectively analyze radiation data. Large scale CMB data is available from a variety of astrophysical experiments such as COBE [Bennett et al., 1996, Leisawitz, 1999], BOOMERANG [de Bernardis et al., 2000], MAXIMA [Lee et al., 1998], and Planck [Bersanelli et al., 1996]. A complete analysis of CMB requires large scale simulation and analysis of full-sky maps at very high resolutions.

Recently, a project by the name SETI@home (Search for Extraterrestrial Intelligence [Anderson et al., 1999]) gained prominence due, in large part, to its ingenious approach to harnessing the large computational resources of the Internet. This project analyzes data collected from the Arecibo Radio Telescope in Puerto Rico to search for patterns and anomalies indicating extraterrestrial intelligence. The data is parceled into packets of 330K and sent off to participating clients (to date, there have been more than two million client machines running the SETI analysis software). These clients look for interesting artifacts in the data and report back potential anomalies to the server. 'Interesting artifacts' in this context correspond to strong and steady signals in a narrow frequency band, pulsed signals, and continuous tones, among other features.

In addition to addressing the issue of analyzing data from various parts of the electromagnetic spectrum, research has also focused on supporting tools and techniques. These tools include compression techniques, improved information storage and retrieval structures, distributed frameworks, and data fusion. Data fusion can play an important role in astrophysical applications since emissions in different ranges of the spectrum may correlate to manifest interesting artifacts in data [Korn et al., 1997, Moore, 1998, Moore and Lee, 1998].

2.3 Data Mining in Chemical and Materials Engineering Applications

Data mining challenges in chemical and materials engineering can be broadly classified into the following categories: structure classification and prediction of properties, structure-activity relationships, and materials design.

The problem of predicting the behavior of materials under various physical conditions has been studied using regression analysis [Vashishta et al., 1996a, Vashishta et al., 1996b]. Extensive experimental and simulation data has been used to compute such higher order relationships as "thermal conductivity and Young's modulus of silicon nitride scale with the density as $\rho^{1.5}$ and $\rho^{3.6}$" and that "pores appear in silicon nitride as density

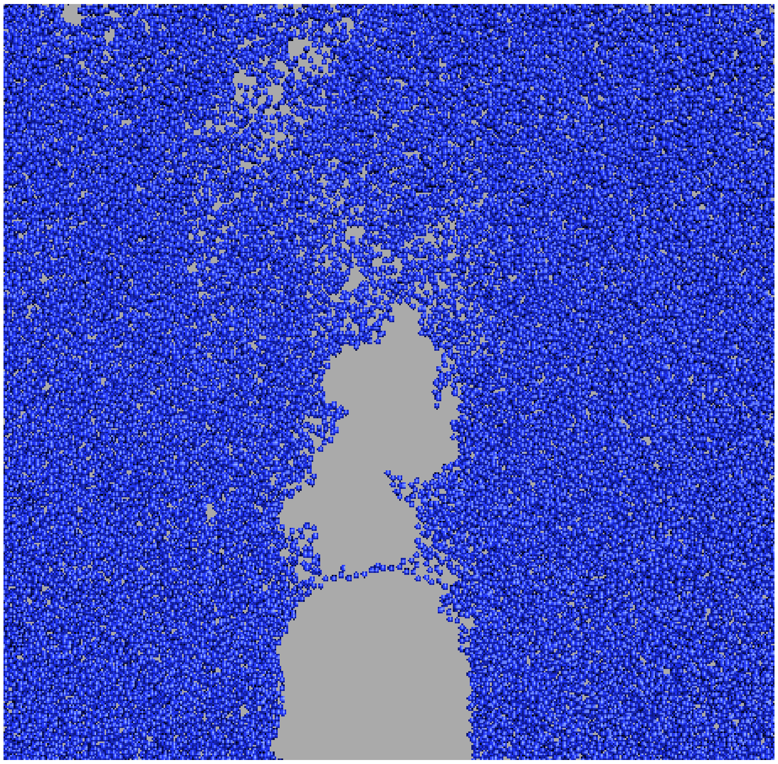


Figure 2: Molecular dynamics simulation demonstrating a crack in an amorphous silicon nitride film. Slow intra-microcrack propagation produces smooth surfaces, while coalescence of microcracks leads to rapid propagation and produces rough surfaces. The nature of crack surfaces is important in many application. The problem of extracting a crack surface dynamically from large scale molecular dynamics simulation data is an important analysis task. Figure courtesy A. Nakano (Louisiana State University).

reduces to 2.6 g/cc." [Nakano et al., 1995]. Physical processes such as crack propagation and fracture surface characteristics have also been investigated (Figure 2). Data handling and compression frameworks to support discovery of such relationships have been developed [Nakano et al., 1995, Yang et al., 1999].

The goal of designing materials with specified properties is a more complex one considering the fact that the forward problem of predicting material properties is itself not completely understood. Furthermore, applications involving life-expectancy based design require that all or most parts of a structural component fail simultaneously. For this reason, heuristics and machine learning approaches have a very important role to play in this area. Industrial applications of materials design include composites and blends, agricultural chemicals, refrigerants, solvents etc. With increased environmental awareness and concerns, greater emphasis is being placed on the design of novel materials.

Traditional approaches to design of materials have relied on a trial-and-error paradigm. The complexity of this paradigm lies both in the large dimensionality of the search space and the need to guide the search process with appropriate heuristics. Computer aided materials design (CAMD) has thus gained prominence over the recent past. CAMD fundamentally poses an inverse problem in which the structure must be inferred from function

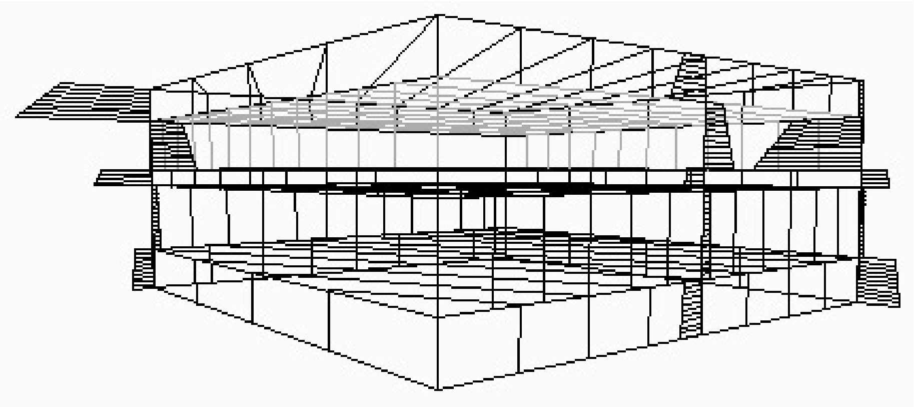


Figure 3: Wireframe model of a wood-based composite showing failed layers (gray) and active layers (black), and the orientation of fibers in each layer. In this figure, the second layer has failed. The horizontal protrusions are proportional in length to the magnitude of forces being applied to the layers at the time of failure. Data mining can help characterize failure criteria, by taking into account fatigue data and domain specific background knowledge. Figure courtesy C.A. Shaffer (Virginia Tech).

(or properties). A number of traditional approaches ranging from neural networks and genetic algorithms to tree classifiers have been explored by researchers. One such system — GENESYS [Venkatasubramanian et al., 1994, Venkatasubramanian et al., 1995], has been found to be useful in the design of several novel materials.

Crack propagation studies and failure analysis are now becoming increasingly sophisticated with the advent of parallel molecular dynamics [Nakano et al., 1999], nanoscale simulations [Nakano et al., 1995], and problem-solving environments (PSEs) [Goel et al., 1999]. Understanding how and why materials fail and fracture constitutes one of the grand challenge problems in computational materials science. Fig. 3 describes a materials analysis scenario involving wood-based composites [Goel et al., 1999]. The failure of one or more layers upon simulation of various forces, as shown, helps in assessing the strength properties of reinforced materials. Data mining in this context can help characterize interface phenomena from typical data sets which, in turn, can be utilized in adaptive control and dynamic analysis. In addition, technologies such as Micro-Electro Mechanical Systems (MEMS) open up the possibility of designing smart and intelligent structures [Berlin and Gabriel, 1997], embedded internet devices, and programmable vector fields for micromanipulators [Böhringer et al., 1997]. Such devices currently lack powerful programming abstractions and languages that can support distributed intelligence and large-scale autonomy.

At a broader level, multidisciplinary analysis and design (MAD) of complex devices such as gas turbine engines require knowledge and computational models from multiple disciplines. For example, the analysis of an engine involves the domains of *thermodynamics* (specifies heat flow throughout the engine), *reactive fluid dynamics* (specifies the behavior of the gases in the combustor), *mechanics* (specifies the kinematic and dynamic behaviors of pistons, links, cranks etc.), *structures* (specifies the stresses and strains on the parts) and *geometry* (specifies the shape of the components and the structural constraints). The design of the engine requires that these different domain–specific analyses interact in order to find the final solution. While these different domains might share common parameters and interfaces, each of them is governed by its own constraints and limitations. There are now thousands of well defined software modules for modeling various parts and behaviors or for supporting the simulation process. For most design aspects, there are multiple software modules to choose from. These embody different numerical methods (iterative or direct solvers), numerical models (standard finite differences, collocation with cubic elements, Galerkin with linear elements, rectangular grids, triangular meshes), and physical models (cylindrical symmetry, steady state, rigid body mechanics, full 3D time dependent physics). Data mining can reveal the most appropriate models and algorithms to choose in a particular situation, by mining benchmark studies involving batteries of standardized problems and domains [Drashansky et al., 1999]. Furthermore, such complex devices typically have tens of thousands of parts, many of which experience extreme operating conditions. Important physical phenomena take place at spatial scales from tens of microns (combustion, turbulence, material failure) to meters (gas flow, rotating structures) and at temporal scales from microseconds to months. Modeling such complex phenomena can benefit from techniques such as qualitative computing [Forbus, 1997] and order-of-magnitude reasoning [Murthy, 1998, Nayak, 1992]. An excellent survey of computational techniques in mechanics appears in [Noor, 1997].

2.4 Data Mining in Bioinformatics

Bioinformatics constitutes perhaps one of the most exciting and challenging application areas for data mining. This information-centric view of biology has ushered in a veritable revolution in genomics involving technologies such as DNA sequencing and linkage analysis. A compelling frontier is the design of software tools that can harness and mine the continuing amount of data generated by such research efforts. Recently, Celera Genomics, Inc., Maryland, USA and the federally-funded multi-nation Human Genome Project have announced the creation of a rough map of the entire human genome. These studies attempt to discover all of the approximately 35,000—100,000 human genes and make them accessible for further biological study and to determine the complete sequences of the 3 billion DNA subunits. Determining the structure of complex organic molecules, correlating the structure and function of the molecule, and engineering a molecule with desired function present key challenges in bioinformatics. The size and complexity of the molecules renders these tasks extremely difficult and computation-intensive.

Extensive research has been done on computational techniques for correlating structure and function of the gene. These techniques span the areas of machine learning (clustering, classification), algorithmics (sequence alignments, pattern matching), and statistics (Bayesian learning, Model fitting). An example of structure-function correlation is illustrated in the case of tumor protein $p53$ (TP_{53}). The human gene TP_{53} belongs to the family $p53$ of proteins. This family is responsible for suppressing the growth of tumors ('preventing cancerous mutiny' [Ridley, 2000]). Through extensive data analysis, it has been found that $p53$ is frequently mutated or inactivated in about 60% of cancers. Also, it has been found that $p53$ interacts with cancer-associated HPV ($e6$) viral proteins causing the degradation of $p53$. This requires an additional factor $e6 - ap$, which stably associates with $p53$ in the presence of HPV viral proteins.

In contrast to TP_{53} about which a fair deal is known, there are other proteins about which very little is known. For example, it is known that $BRC1\_HUMAN$ (Breast Cancer

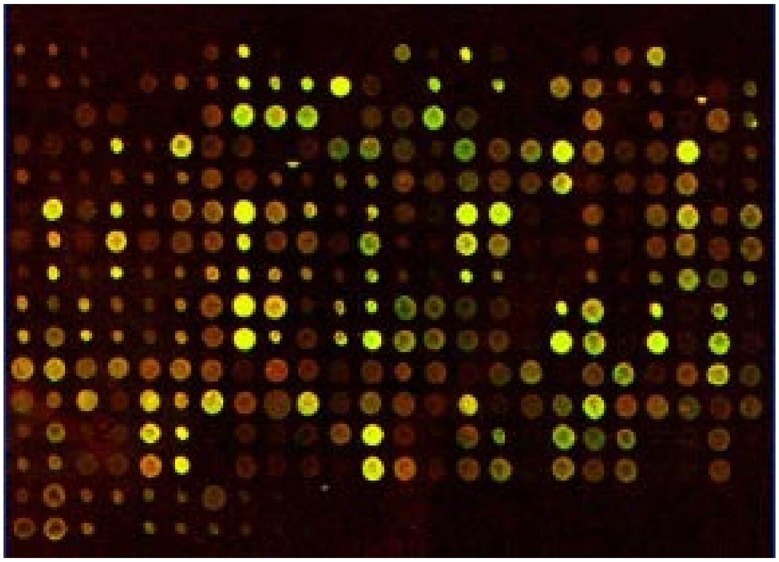


Figure 4: Biological variation of gene expression in Loblolly pine clones by microarray analysis. Data mining is used to identify clusters of genes that are co-expressed under similar stress/drought conditions, helping to hypothesize and/or update gene regulatory networks. The microarray is printed in four 24×16 sub-quadrants, one of which is shown here. Figure courtesy Y.-H. Sun (NCSU).

Type 1 Susceptibility Protein) is known to have high incidences of mutations in breast and ovarian cancer families (45% of inherited breast cancers). However, the precise function of this protein is not known, and is suspected to be responsible for regulating gene expression. Relatively little is known about the other proteins in this family. Clearly, the usefulness of this information is tremendous.

The rapid emergence of microarray technologies contributes another major opportunity for data mining. Microarrays provide an experimental approach to measure levels of gene expression changes at a system-wide scale, as subject to growth, stress, and other conditions. Levels of gene expression are measured by co-hybridization (of nucleic acid samples) and determining the ratio of signal expression by laser scanners. The intensity values (or ratios) thus determined are used to create an image file (see Fig. 4) that serves as the starting point for data mining. This domain is characterized by imperfections in the experimental process, lack of repeatability, and the high cost of obtaining experimental data. In addition, the sources and causes of experimental variability are not well-understood. The detection of such gene expression changes (for varying levels of stress, say) coupled with modeling time and/or position effects is an area of active research in the bioinformatics community. Clustering of gene expression levels should take into account apriori background knowledge for hypothesizing gene regulatory networks. Standardization of data formats, and experiment management support are also important for effective and seamless applications of data mining.

2.5 Data Mining in Flows

Finite element formulations of a variety of flow simulations and structures have been among the most compute-intensive applications of large-scale supercomputers. These techniques have been used to understand and design machine parts, airframes and projectiles, and non-invasive medical procedures for cauterizing tumors among other things. The underlying problems range from “how to stir vegetable soup in a pot so that each serving contains the same composition of ingredients” to “how to design stealth airframes to minimize scattering signatures and detection by enemy radar.”

Analyzing fluid flows, especially turbulent flows, remains an extremely difficult and challenging problem. Minimizing turbulence is at the heart of reducing drag on airfoils and automobiles, improving performance of jet engines, and maximizing convective cooling. At the same time, maximizing turbulence is important, for example, in complete mixing of charge in engine cylinders and the distance a dimpled golf ball travels on a drive. Emerging medical applications of flows rely on focused electromagnetic fields that generate eddy currents within specific regions of the body to cauterize tumors. Particulate flows are being investigated for understanding blockages in arteries and vascular athresclerosis.

Simulations and observations in this domain address the forward problem of predicting flow behavior and its impact, and the inverse problem of designing features that optimize desired flow behavior. For example, the presence of longitudinal grooves, also known as riblets, placed a few tens of microns apart on airframes can reduce viscous drag by as much as 6%. Similarly, active surfaces comprised of MEMS sensors and actuators can be controlled to minimize drag on airframes.

The input data for analysis of flows typically comes from large scale simulations. Scalar and vector fields such as pressure, potential, and velocity are defined over a finite element or a finite difference grid. Typical grids from large simulations can be in the range of 10^7 elements with over 100 bytes of data defined at each grid point per timestep [Sarin and Sameh, 1998]. This corresponds to roughly a GB of data at each timestep. The simplest analysis tasks correspond to extraction of artifacts such as vortices and saddle points from simulation data [Peikert and Roth, 1999, Sadarjoen and Post, 1999, Soria and Cantwell, 1992, Tanaka and Kida, 1993, Yates and Chapman, 1990]. This corresponds to a spatio-temporal mining process that detects, for instance, singularities in the flow field. More complex tasks in analyzing flows correspond to correlating flow structure with function, for example, predicting drag coefficient of airfoils. Ultimately, the goal is to develop structures that exhibit more desirable properties. In the case of airframes, this could be stability characteristics (or instability/ maneuverability) and drag. Research efforts [Gage et al., 1995, Gallman and Kroo, 1996, Nguyen and Huang, 1994] have relied on neural networks and genetic programming to optimize the design process for airfoils.

More recently, fluid flow simulations have been used to study the formation of blockages in arteries. It has been noticed that when a vein graft is incorporated to bypass a blockage, new blockages start to form almost immediately. The onset and rate of growth of these blockages depends on a variety of factors – the adhesiveness of graft walls, the pressure of particulate fluid flowing through the graft, and the concentration and nature of particulates [Lei et al., 1997, Liu, 1996]. Simulations are being used to design materials with low adhesiveness that can be used for vein grafts. It has been observed that the formation of blockages can be impeded by increasing the fluid pressure within the graft. In experiments,

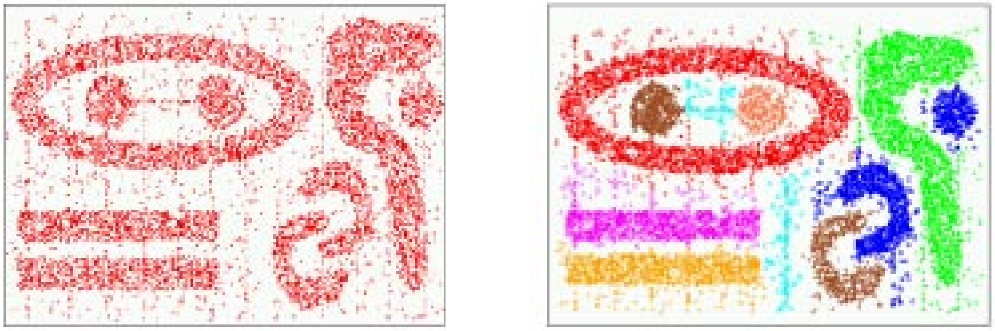


Figure 5: (left) Typical unlabeled input data to a clustering algorithm. (right) Clusters mined by the CHAMELEON algorithm based on similarity and proximity metrics [Karypis et al., 1999]. Figure courtesy G. Karypis (Univ. Minnesota).

this is achieved by pinching the artery. Machine learning techniques are being explored for determining both the physical composition as well as the structure of the artificial graft.

3 Inductive Mining Techniques

We start our discussion of data mining techniques by examining some key approaches that have been successfully applied to various scientific domains. The most commonly held view of data mining is one of induction — proceeding from the specific to the general. Techniques in this family differ in the nature of the induced representations (decision trees/rules/correlations/deviations/trends/associations etc.), the form of the data they operate on (continuous /time-series/discrete/labeled/nominal etc.), or the domains motivating their application (financial/economic/scientific/engineering/computational etc.). The patterns, in turn, can be characterized based on aspects such as accuracy, precision, expressiveness, interpretability, 'surprisingness/interestingness,' or actionability (by the scientific enterprise). For example, a pattern that can translate into sound scientific decisions is better than one that is accurate, interesting, but provides no tangible benefit. A good example is reflected in the pattern "People who are good at knitting tend not to have the Y chromosome" [Ridley, 2000]!

3.1 Underlying Principles

Clustering

An area where tremendous progress has been made in inductive learning is *clustering* — a fundamental procedure in pattern recognition that looks for regularities in training exemplars. By revealing associations between individuals of a population, it can be used to provide a compact representation of the input problem domain (see Fig. 5). The basic goal of clustering is to obtain a $c-$partition of the input data that exhibits categorically homogeneous subsets, where n is the number of training exemplars and $2 \leq c \leq n$ [Bezdek, 1981]. Different clustering methods have been proposed that represent clusters in different ways - for example, using a representative exemplar of a cluster, a probability distribution over a space of attribute values, necessary and sufficient conditions for cluster membership etc. To represent a cluster by a collection of training exemplars and to 'assign' new samples to existing clusters, some form of a utility measure is used. This is normally based on some mathematical property such as distance, angle, curvature, symmetry, and intensity, exhibited by the members of the cluster. Various techniques have been proposed to reveal such associations; the most prevalent model this either as a form of density estimation (unsupervised) or classification (supervised). Complexity control can then achieved by regularization methods that operate within the structural risk minimization framework [Vapnik, 1995]. It has been recognized [Ruspini, 1969] that *no* universal clustering criterion can exist and that selection of any such criterion is subjective and depends on the domain of application under question. Clustering serves key applications in sky survey cataloging [Fayyad et al., 1996], bioinformatics [Schulze-Kremer, 1999], spatial data mining [Ng and Han, 1994], geographical mining [Knorr and Ng, 1996], dynamical systems analysis [Zhao, 1994], and various other domains. See [Ganti et al., 1999b] for an excellent survey on clustering algorithms as proposed in the database community.

Scientific Function-Finding

From a heuristics point of view, the ‘empirical discovery’ research of Langley, Simon, and Bradshaw in the 1980s [Langley et al., 1990] constitutes arguably one of the earliest systematic studies of data mining in scientific domains. The BACON system presented in [Langley et al., 1990] attempts scientific function-finding and uses a set of heuristics to explore the space of possible functional forms between columns in a numerical table. For example, if two columns of a table correspond to quantitative measurements of the force and the distance between atomic particles, BACON would use the monotonically decreasing relationship between the two columns to identify their product as a potential (new) column (of values) for further exploration; eventually leading to the inverse square relationship between the two variables. BACON has been shown to rediscover relationships such as Kepler’s laws, Dalton’s equations in chemistry, and other patterns in thermodynamics, optics, and electricity. In addition, it has been shown to form internal representations of ‘intrinsic concepts’ that are later used in expressing the laws. We will return to this issue in our discussion on constructive induction (see later). While the original goal of BACON was to study the use of heuristic search as a mechanism for data-driven discovery, its main drawback was the inability to handle noise and uncertainty in physical data measurements. Detection of such higher order relationships now constitutes a major activity in computational science (see the earlier section on data mining applications in materials engineering). Notice that function-finding is broader than curve fitting since the functional form and/or structure of the relationship is not known beforehand.

Universal Function Approximators

More general forms of functional relationships can be modeled by neural networks [Jordan and Bishop, 1997], which are shown to approximate any function to any required level of accuracy (perhaps with exponential increase in complexity) [Cybenko, 1989]. Neural networks use one or more layers of intermediate functional elements to model the dependence of an output signal(s) on given input parameters. It is shown in [Cybenko, 1989] that two layers of in-between elements are sufficient to model any continuous function. The task of learning in this scenario is thus to determine the interconnection strengths and threshold biases (weights) that will result in acceptable approximation. While the general problem of determining weights has been shown to be NP-complete [Blum and Rivest, 1992], neural networks have emerged as a valuable tool for engineers and scientists in pattern recognition, signal processing, and function approximation. Such applications utilize some form of gradient-descent or other local optimization techniques to 'train' neural networks. Since neural networks function effectively as 'black-boxes', their use in enabling knowledge discovery is limited. Their results are notorious for being inscrutable and it is typically very difficult to reverse-engineer a set of logical rules that capture their decision-making. Limited successes have been achieved in mining 'M-of-N rules' that model patterns of the form 'If three of these five features are present, the patient has coronary heart disease' [Fu, 1999]. Neural networks also suffer from other drawbacks, such as the capacity to incorporate prior knowledge in only a limited form [Towell and Shavlik, 1994] and excessive dependence on the original network topology [Opitz and Shavlik, 1997]. For an excellent introduction, we refer the reader to [Jordan and Bishop, 1997].

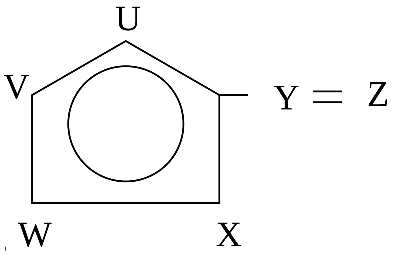
Logical Representations

While neural networks are attribute-value based techniques, more expressive schemes can be obtained by harnessing the representational power of logic (specifically, first-order logic). In this formalism, facts and measurements are represented as logical facts and data mining corresponds to forming an intensional rule that uses relational representations to model the dependence of a 'goal' predicate on certain input predicates (relations). This 'inductive logic programming' (ILP) approach [Bratko and Muggleton, 1995] can be used to find patterns such as:

```
patient(X,'diabetes') :- symptom(X,'S1'), symptom(X,'S2')
```

which indicates that a patient (`X`) suffers from diabetes if he/she demonstrates symptoms `S1` and `S2`. Notice the easy comprehensibility of the rule which could be later used in diagnostics and What-if analyses. In addition, such rules can also be recursive, a feature which makes ILP amenable to automated program synthesis [Bratko and Muggleton, 1995] and discovery of complex relationships such as protein secondary structure [Muggleton, 1999]. In addition, ILP allows the incorporation of apriori background knowledge, a necessary pre-requisite for mining in complex structured domains. ILP systems typically take a database of positive examples, negative examples, and background knowledge and attempt to construct a predicate logic formula (such as `patient(X,Y)`) so that all (most) positive examples can be logically derived from the background knowledge and no (few) negative examples can be logically derived. ILP has strong parallels in deductive database technology, as demonstrated in [Dzeroski, 1996]. Fig. 6 describes the results of correlating the mutagenicity of chemical compounds with their structure [Srinivasan and King, 1999b]. As shown, the expressiveness of ILP [Bratko and Muggleton, 1995] makes it a highly desirable tool in structured domains where comprehension and interpretation of patterns is important.

In the most general setting of first-order logic, relational induction as described above is undecidable. A first restriction to function-free horn clauses results in decidability (this is the form of programs used in the programming language PROLOG). Decidability here is with respect to induction; for deduction, first-order logic is semi-decidable. However, ILP is often prohibitively expensive and the standard practice is to restrict the hypothesis space to a proper subset of first-oder logic (the popularity of association rules [Agrawal et al., 1993] can



A chemical is highly mutagenic if
it contains a double bond conjugated to a
5-membered aromatic ring via a carbon atom.

Figure 6: Two representations of a construct mined by ILP in a chemical setting. The input consists of positive and negative examples of mutagenic compounds, detailing their bonding structure. The background knowledge consists of commonly known rules in chemistry and bond theory. The output is represented in two forms: (left) graphical depiction and (right) English statement of the pattern. Figure courtesy A. Srinivasan (Oxford University).

be attributed to a further restriction of the hypothesis space that permits certain optimizations). Some commercial systems (like Golem [Muggleton and Feng, 1990]) further require that background knowledge be ground, meaning that only base facts can be provided as opposed to intensional information (rules). This renders the overall complexity polynomial in the space of the database but pseudo-polynomial (and sometimes exponential) in the space of the modifiable parameters (such as the rule length, clause depth, compression factor, and rule cover). In addition, the ability to process, represent, and mine predicates involving numeric attributes is limited. Nevertheless, ILP has witnessed successful applications in bio-informatics, finite-element mesh design, chemical structure prediction, water quality prediction, and biological structure discovery. See [Muggleton, 1999] for an excellent survey of ILP applications.

Constructive Induction

This compelling research frontier refers to the ‘invention’ of new features for subsequent use in data mining. An excellent example of human-performed constructive induction can be had from Francis Crick’s experiences in unraveling the language of DNA. We use the fascinating account of Matt Ridley in [Ridley, 2000] to illustrate the basic idea. Crick was attempting to determine how sequences comprising four bases (A - Adenine; C - Cytosine; G - Guanine; T - Thymine) coded for twenty amino acids. The goal was to determine the right level of in-between representation (codons) that aided in this transcription process. Crick’s reasoning was that two bases for each amino acid is too few (16) while three bases provide an ample starting point for fine tuning the analysis. To prevent Nature from making translation errors, he reasoned that base sequences that could be misread should be eliminated. This removed sequences such as AAA etc., leading to 60 sequences which can be clustered into 20 groups of sequences whose members are equivalent under rotational transformations (ACG, CGA, GAC etc.). While this idea turned out to be close but not perfect — three bases were right, but not the encoding — it provides an excellent example of the benefits of constructive induction in data mining. Ridley reports that this has been called the ‘greatest wrong theory in history.’ From a computational perspective, Craven and Shavlik [Craven and Shavlik, 1993] showed that the problem of finding the right codon representation is notoriously difficult with traditional machine learning techniques. As the mass of data generated under the Human

Genome project gets assimilated, the role of (semi-automated) constructive induction will only become paramount.

Integrated Approaches

While ILP provides a logic-based representation for scientific background knowledge, other approaches to mining in scientific domains utilize various forms of theory-driven reasoning to augment, direct, and support empirical analyses. For example, the MECHEM program [Valdés-Pérez, 1994] uses domain-specific constraints on chemical reaction pathways to direct the search for plausible molecular reaction steps. MECHEM has found applications in areas such as urea synthesis and propane oxidation. Similar systems have been reported in graph theory, link analyses, and componential analyses. A survey of such systems is provided in [Valdés-Pérez, 1999a] which credits their success to systematic and comprehensive search in large spaces and good paradigms of human-computer interaction [Valdés-Pérez, 1999b]. It thus cannot be overemphasized that data mining processes are fundamentally iterative and interactive [Hellerstein et al., 1999]. Systems that enable such interaction and perform integrated exploration and mining [Ramakrishnan and Grama, 1999] can thus improve performance in the long run, while sacrificing some exploration time in the short-term.

3.2 Best Practices

One of the key issues to be addressed prior to data mining is the decision on the right mechanism for representation. For example, while a simple structure might exist, the feature coordinate system might not reflect the simplicity properly. Consider the case when a particular feature system imposes the following pattern involving disease symptoms and medical treatments [Rice, 1995]:

Treatment 1 is best if $x^2 + y^2 \leq 1$
Treatment 2 is best otherwise.

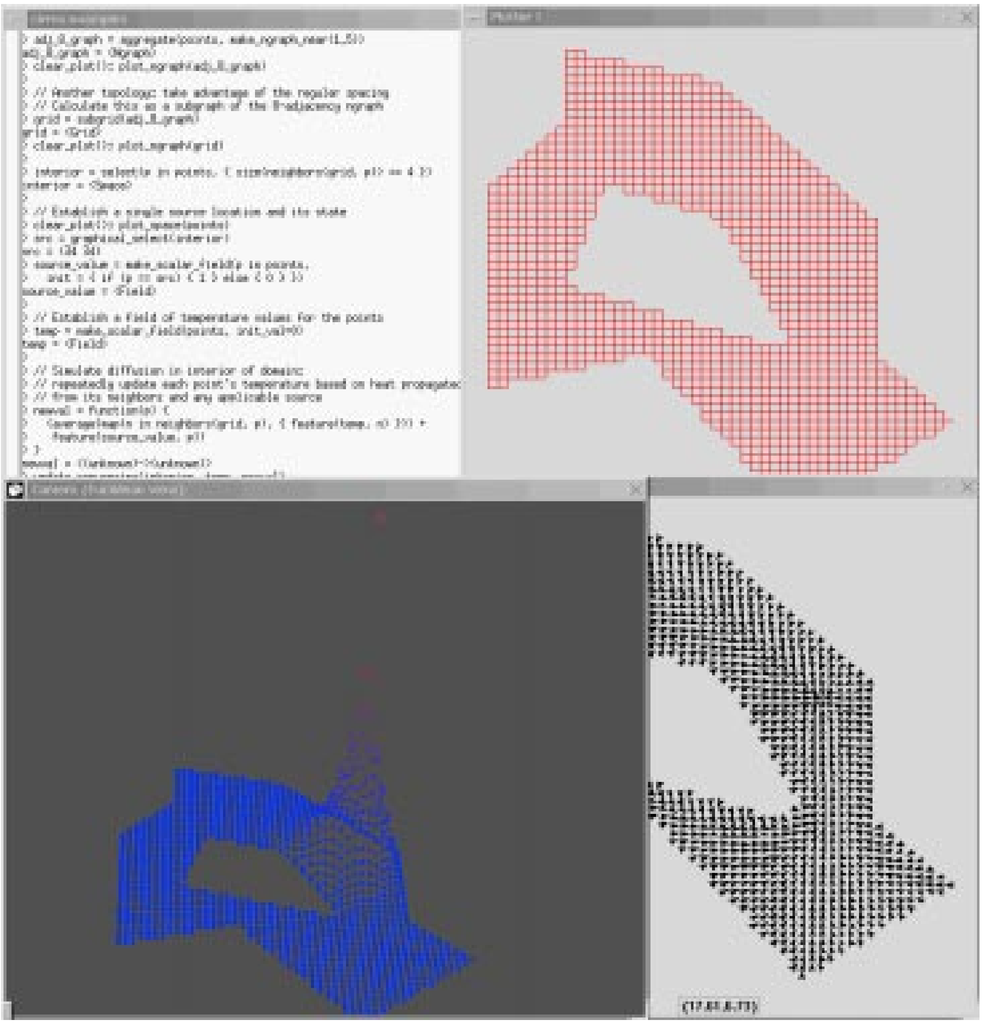
If we now choose new coordinates (x', y') such that

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1.0001 \\ 1 & 1.0000 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

then Treatment 1 is best in a very long, very thin area (in (x', y') coordinates). If the following features are now chosen,

$$\begin{aligned} f1 &= x' \\ f2 &= y' \\ f3 &= x' + y' \sin x' + z1 \\ f4 &= x' - y' + 1/x' + e^{y'} + z2 \end{aligned}$$

where $z1$ and $z2$ are irrelevant, or random, it will then take a lot of data and effort to recover the original simple pattern. Thus, the pattern mined might depend in an unstable manner on the features actually used $(f1, f2, f3, f4)$ and no reasonable amount of brute force computing can provide a robust selection methodology in such a situation. In addition, the algorithms/techniques used should be able to isolate as many irrelevant features as possible from the induced generalization. We identify various issues that are pertinent in mining scientific data.



```
> adj_8_graph = aggregate(points, make_ngraph_near(1.5))
adj_8_graph = <Ngraph>
> clear_plot(); plot_ngraph(adj_8_graph)
>
> // Another topology: take advantage of the regular spacing
> // Calculate this as a subgraph of the 8-adjacency ngraph
> grid = subgrid(adj_8_graph)
grid = <Grid>
> clear_plot(); plot_ngraph(grid)
>
> interior = select(p in points, { size(neighbors(grid, p)) == 4 })
interior = <Space>
>
> // Establish a single source location and its state
> clear_plot(); plot_space(points)
> src = graphical_select(interior)
src = (34 34)
> source_value = make_scalar_field(p in points,
>    init = { if (p == src) { 1 } else { 0 } })
source_value = <Field>
>
> // Establish a field of temperature values for the points
> temp = make_scalar_field(points, init_val=0)
temp = <Field>
>
> // Simulate diffusion in interior of domain:
> // repeatedly update each point's temperature based on heat propagatec
> // from its neighbors and any applicable source
> newval = function(p) {
>    (average(map(n in neighbors(grid, p), { feature(temp, n) })) +
>     feature(source_value, p))
> }
newval = (Unknown)->(Unknown)
```

Figure 7: An instance of the SAL interpreter: window showing typical user interaction (top left), the finite difference discretization (top right), resulting heat flows (bottom right), and the thermal hill from a single source (bottom right). Figure courtesy C. Bailey-Kellogg (Dartmouth College).

Determining High-Level Structures

The past decades have witnessed the maturing of clustering algorithms, primarily with the advent of statistical pattern recognition and database systems. It is beyond the scope of this chapter to summarize these developments. As identified in [Cheeseman and Stutz, 1996], it is important to caution, however, that the indiscriminate application of clustering algorithms is not recommended and every application domain merits serious study and pre-processing before attempting clustering and/or interpreting the results of clustering. Such preprocessing techniques involve principal component analysis, feature handcrafting, data filtering, and visualization procedures such as Sammon's mapping, which produces class-preserving projections. Most clustering algorithms assume a prior on the structure of the induced clusters (e.g. some parametric approaches), thresholds on observed features, or non-parametric ideas such as valley-seeking [Fukunaga, 1990]. An interesting approach to clustering is provided by the Chameleon system [Karypis et al., 1999] that induces a graph (based on a neighborhood metric), and then views clustering as a form of graph partitioning. This approach allows the incorporation of both interconnectivity and closeness constraints on the induced clusters (background knowledge in a limited form). As shown in Fig. 5, it is flexible enough to handle clusters of widely varying sizes, shapes, densities, and orientations. An alternative to clustering numerical data based on such measures of similarity constitutes clustering based on co-occurrence and categorical information [Gibson et al., 1998, Ganti et al., 1999a], which helps prevent information loss due to discretization.

A more holistic approach to determining high-level structures is demonstrated in the SAL system [Zhao et al., 1999] that provides the imagistic paradigm of spatial aggregation [Yip and Zhao, 1996], ontologies for scientific data abstractions (fields, graphs etc.), and a compositional modeling framework for structure discovery in data-rich domains. Various important mining operations can be realized by applying primitives in a programmatic manner to neighborhood graphs, such as aggregation and classification. Fig. 7 describes a snapshot of the SAL system that uses the graph-theoretic primitives to find thermal hills in the flow obtained by aggregating classes of temperature values. The SAL framework has been used in many applications, including weather data analysis [Huang and Zhao, 1998], optimization and qualitative analysis of other physical systems [Yip and Zhao, 1996]. The role played by such frameworks in integrated control and system design is elaborated upon later in the chapter.

Controlling the Complexity of Induction

Various mechanisms have been proposed to address the complexities of techniques such as inductive logic programming to form high-level representations. Three main approaches can be identified. At a basic level, domain specific restrictions are being increasingly incorporated into the mining process. For example, both syntactic and semantic restrictions help curtail the search. A syntactic restriction refers to constraints posed on the nature of predicates that can appear in the antecedent and consequent parts of a rule and the nature of variable interactions allowed in the predicates. Semantic restrictions model consistency constraints and provide various means of *sanity checks*. Such constraints can also enable optimizations that *push* costly mining operations deeper into the computational pipeline [Han et al., 1999]. Second, generality orderings are used to guide the induction of rules. Such orderings are used to prune the search space for generating plausible hypotheses and to aid in abduction (which is the process of constructing a rule that needs to be justified further). And finally, the software architectures of data mining systems are typically augmented with natural database query interfaces [Imielinski and Mannila, 1996], so this aspect can be utilized to provide meta-patterns for rule generation ('Find me a pattern that connects something about patients' athletic activities to the dates of occurrence of some symptoms') [Shen et al., 1996].

Handling Uncertainty and Noise in Measurements

As mentioned earlier, typical datasets are characterized by mislabeled or unavailable data, and lack of repeatability or accuracy in measurements (see the examples on microarray data and landuse segmentation described earlier). A promising approach that is found to be effective in such domains is described in [Zhao and Nishida, 1995], where various forms of data relationships underlying the domain are explicitly modeled as qualitative correlations, which are then used to determine inaccurate data. An application to infrared spectrum identification is also described in [Zhao and Nishida, 1995]. A matrix-theoretic approach to this problem is provided by the Singular Value Decomposition (SVD), introduced in Section 4. SVD can help perform subset selection on either the variables or the data points, thus aiding noise reduction and/or removal of redundancy.

Inventing New Features

Research into constructive induction has provided various solutions to this problem, though the larger fundamental question still remains. In an attribute-value setting, statistical software such as SAS/STAT contain various forms of regression tools that incrementally postulate and add terms to form functional relations. In relational settings, description logics (sometimes called terminological logics) have been proposed as an extension to first-order logic [Frazier and Pitt, 1996] where all but one of the variables are quantified. This allows the expression of predicates such as 'at least two,' 'at most three,' which can be viewed as operating upon the individual original user-supplied predicates. Such systems have been used for research into correlating patient symptoms with diseases, and have been shown to excel ILP in their representational basis [Kietz and Morik, 1994]. In addition, [Kietz and Morik, 1994] shows that such representations form one of the largest subsets of first-order logic that is tractable under inductive inference. It has also been shown that ILP (with its logical representation language) is well suited for constructive induction [Srinivasan and King, 1999a] by introducing features consisting of connected sequences of predicates.

4 Data Mining as Approximation and Lossy Compression

The approach of "stepping back to get a bigger picture" is often used (sometimes inadvertently) for analyzing large datasets. We discuss this family of techniques as mining by approximation or lossy compression.

4.1 Underlying Principles

In many applications of scientific data mining, the problem of identifying artifacts of interest in data is similar, if not identical, to the task of lossy data compression. The objective of lossy compression is one of reducing effective storage while retaining as much of the information content as possible. This correspondence between data compression and mining has roots in information theory (mutual information), numerical methods (rank reduction, eigenvalue analysis), algorithmics (geometric simplifications), and applications (latent semantic indexing, etc.).

Compressing Data Defined Over Geometry and Topologies

An important class of problems in scientific computing relies on mesh-based and mesh-free bases for solving integral and differential equations. This class spans particle dynamics, finite element/difference techniques, and boundary element methods. Many of these problems have a time dependency and generate extremely large datasets over typical application runs. The rich set of tools in this domain has prompted the formulation of problems in other domains such as information retrieval and market basket analysis as problems defined over point sets using a vector-space model. The issue of compressing these datasets has received considerable attention, although the correspondence between compression and mining has been tenuous. Compression and analysis of large scientific datasets takes on two major forms: detection of features that span several spatio-temporal scales and detection of artifacts at a specified scale. We discuss these in the context of a variety of applications.

Detecting and Coding Multiresolution Phenomena

Large-scale simulations in molecular dynamics, astrophysics, and materials processing result in time-dependent data, with each timestep comprising of a set of points (typically in 3D space) and attributes associated with each of these points. Attributes range from scalar fields such as potential and pressure to vector fields such as velocity and electrostatic/gravitational fields. These applications present considerable challenges for simulation and modeling due to the highly multi-resolution nature of the associated physical phenomena and the dense nature of node interactions – specifically that in many of these applications, each particle is impacted by every other particle. This issue of all-to-all interaction complexity is addressed by fast methods such as the fast multipole method (FMM) [Board and Schulten, 2000] and Barnes-Hut [Barnes and Hut, 1986]. These methods introduce systematic approximations that reduce computational complexity while causing bounded errors.

Consider the problem of analyzing data from a particle dynamics simulation in materials processing. In these simulations, the smallest timestep must often match the frequency of atomic vibrations *i.e.*, in the range of femtoseconds. However, global conformational changes take several orders of magnitude longer than this time scale. Moreover, these global displacements are often indistinguishable from localized atomic displacements; necessitating the use of innovative algorithms and analysis techniques. This is illustrated in Figure 8, in which a heuristic clustering technique is applied to identify aggregates of particles displaying coherent displacements over large timescales. This large aggregate displacement is indicated by white arrows. The particles within the clusters themselves display localized displacements.

There are several approaches to detecting such aggregate particle behavior. These approaches range from clustering (see Section 3) to rank-reduction techniques. In the case of identifying aggregate displacements, clustering techniques use velocity fields (or differences of positions) over several timesteps to identify particles belonging to the same aggregate.

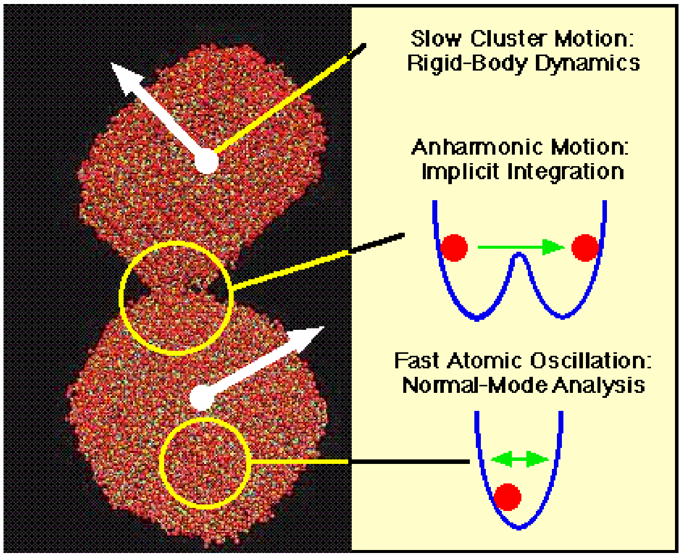


Figure 8: Illustration of difference between global conformational changes and localized atomic displacements. While the localized displacements across all particles display no coherent behavior, global conformational displacements exhibit coherence within the two clusters. The white arrows indicate movement of clusters (coherent behavior) over time-scales much larger than those of atomic vibrations. Such analysis is useful for detecting phenomena such as crack propagation and formation of pores. Figure courtesy A. Nakano (Louisiana State University; `http://www.cclms.lsu.edu/cclms/group/Faculty/nakano.html`).

Note that clustering here is not based on physical proximity, rather on the attribute associated with the behavior being examined. Nakano [Nakano, 1997] uses this approach to examine aggregate behavior in particle systems. This technique uses a membership function $p(i, c)$, which describes the degree of association between atom i and cluster c. The principle of maximum entropy is used to determine $p(i, c)$. In addition to using this as an analysis technique, it is also possible to reduce computational cost of the simulation using clustering. This approach relies on a hierarchy of subdynamics for computing particle trajectories. Atoms are first grouped together into clusters. Dynamics at the cluster level are modeled by rigid-body motion using a large timestep. The fast oscillatory motion of atoms around the local minima in potential is then computed and superimposed. It has been demonstrated that for simulations of sintering of two silicon nitride nanoclusters, the fuzzy-body/implicit-integration/normal-mode (FIN) scheme described above achieves over an order-of-magnitude improvement in performance [Nakano, 1997, Nakano, 1999].

Coding Spatial and Temporal Coherence

In addition to problems associated with multiresolution phenomena, it is often necessary to examine data at a single resolution level (both spatial and temporal) to detect interesting artifacts. In typical domains, neighboring particles (or discretization elements) exhibit spatial as well as temporal coherence – i.e., spatially proximate particles are expected to exhibit similar behavior and a particle is expected to exhibit coherent behavior over consecutive timesteps. The onset of such physical artifacts as cracks and fissures is signaled by a loss in coherence. Such artifacts are of interest and must be preserved by compression schemes.

The task of encoding spatial coherence (or the lack thereof) can be achieved by recursively subdividing the domain and differentially coding with respect to the expected value for the subdomain. This process fits in well with the hierarchical approximations used in many fast simulation techniques (multipole methods, multigrid solvers). Approximations introduced by fast methods also introduce a distortion radii around computed parameters. These parameters can be quantized to any point within the distortion radii without additional loss in accuracy. This process is illustrated in Yang et al. [Yang et al., 1999].

In addition to analyzing high dimensional geometry (continuous attribute sets), in many applications, it is necessary to compress functions implicitly defined over meshes (as opposed to point-sets). This involves the additional task of simplifying the topology in addition to the geometry. A number of schemes have been developed for lossless as well as lossy compression of meshes. Lossless compression techniques for specifying connectivity rely on node orderings for unrolling surface meshes into rings or concentric layers [Bajaj et al., 1999, Rossignac, 1999, Taubin and Rossignac, 1996]. Lossy compression of topology relies on schemes for merging discretization elements based on the error introduced by the simplification. Simple metrics for minimizing this error rely on gradients (merge elements where gradient is minimum) or explicit integration of error in function between simplified and original discretizations. A careful choice of weights for attributes and error metric results in a very effective analysis tool based on mesh simplifications.

4.2 Best Practices

Compressing Continuous Attribute Sets

The key to effective quantization of high dimensional geometry is to identify regions of high and low density and distortion radii. A uniform quantization scheme must rely on a discretization resolution equal to the smallest distortion radii for any attribute. This can result in a significant overhead with respect to compression ratios. Consequently, a multi-dimensional adaptive quantization scheme is needed for arbitrary distributions of attribute values.

One such scheme [Yang et al., 1999] constructs a hierarchical decomposition of the attribute space from a given set of attribute values and distortion radii. This process is illustrated for a two-dimensional problem in Figure 9. The domain is recursively subdivided into quads until each quad contains one entity and the center of the quad is within the distortion radius of the entity contained within. Once such a hierarchical structure has been constructed, entities are assigned to leaf nodes that lie within distortion radii. The problem of representing entity attributes now reduces to the problem of representing populated leaf-level nodes in the tree. This is done by encoding the path from the root to the leaf node. By associating a pre-defined ordering of children in the tree, we can associate d bits per level in the tree for a d-dimensional attribute set. We illustrate this process for a 2D problem in Figure 9. In the example, entity `a` is at level three in the tree and according to the predefined node ordering for children of a node, it is represented as `00 00 11`. This provides the basic quantization mechanism.

This compression and analysis framework can be improved using a number of optimizations. Entities (nodes) that are spatially proximate (in physical space and not attribute space) are likely to share large prefixes in the path from root to leaf. This implies that if the entities are sorted in a proximity preserving order (such as Morton or Hilbert curves), then we can represent entity attributes relative to the previous attributes. The use of spatial coherence for improving compression ratios is illustrated in Figure 9, Time-step 0 in the context of a particle system. The quantized representations for particles `a`, `b`, and `c` are given by `00 00 11`, `00 10`, and `00 11 01` respectively. Assuming that these particles are sorted in the order `a`, `b`, and then `c`, it is easy to see that particles `b` and `c` share the prefix `00` with particle `a`. Consequently, the prefix does not need to be stored for these and the representations for `b` and `c` are simply `10` and `11 01`. While this may not seem to be a significant improvement in this example, in typical trees, the depth can be high. For example, with a normalized domain of unit size in each dimension, a distortion radius of 10^{-3} would require up to 10 levels in the oct tree. In such cases with higher particle densities, significant improvements result from the use of spatial coherence.

Violations of Spatial Coherence

The above framework can be very effective for both compression and analysis. Consider the problem of identifying subdomains that violate the spatial coherence condition. This may occur, for instance, during the formation of cracks or fissures in materials. Here, neighboring particles will exhibit significantly different velocities at the onset of a crack. In this case, a differentially coded, spatially ordered representation of particle velocities would require a large number of bits. The number of bits required to store the position is a good estimation

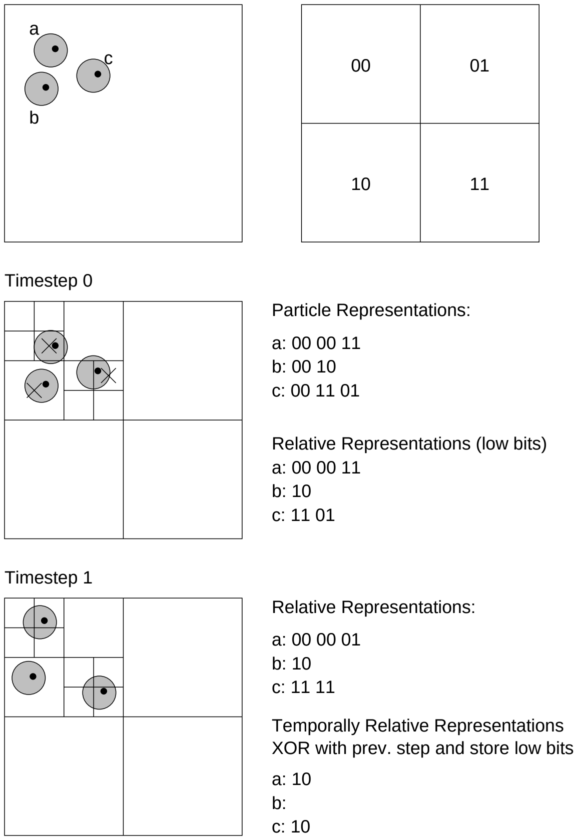

a
c
b
00
01
10
11
Timestep 0
Particle Representations:
a: 00 00 11
b: 00 10
c: 00 11 01
Relative Representations (low bits)
a: 00 00 11
b: 10
c: 11 01
Timestep 1
Relative Representations:
a: 00 00 01
b: 10
c: 11 11
Temporally Relative Representations
XOR with prev. step and store low bits
a: 10
b:
c: 10

Figure 9: Illustration of compression of particle position data using a distortion sphere, and spatial and temporal coherence.

of the spatial coherence. Similar methods can be used to analyze other quantities relating to temporal coherence as well. For example, consider a compression scheme in which difference in particle positions over timesteps are coded in the hierarchical framework. Here, the number of bits is a reflection on the energy of the particles. A similar approach can be used to identify high potential regions in the domain.

Rank Reduction

In each of the above cases, the basic compression technique relies on a distortion radius in a high-dimensional space and quantizes the space adaptively to points within the distortion radius. There are other techniques that are more computationally expensive but are amenable to higher level analysis functions. One such technique relies on rank-reduction of the attributes associated with each point.

Consider, once again, a given point set with an attribute vector at each of the points. The set of all attribute vectors can be viewed as a matrix. A simple rank-one approximation to the matrix is such that each non-zero entry in the column vector can be thought of as having a weighted representation of the corresponding row vector. Using this approach, each point can be mapped to one or more of the singular vectors computed from a singular vector decomposition (SVD) of the attribute vector set matrix. Compression is achieved by simply

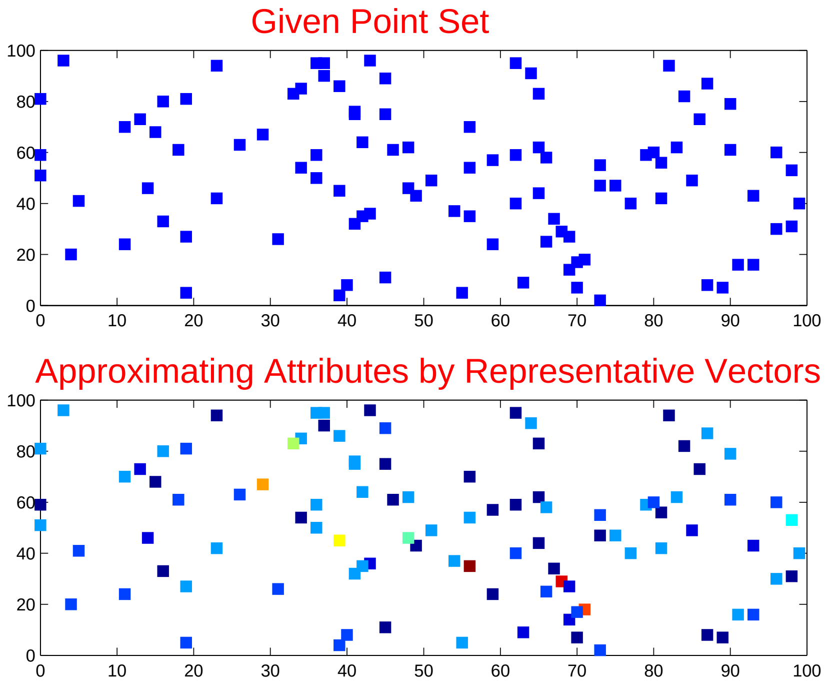


Figure 10: Using rank reduction techniques for analyzing the behavior associated with point sets. Each of the points with the same color is mapped to the same singular vector; i.e., their attributes are within some bounded distance from the representative singular vector. Please see [Ruth et al., 2000, Yang et al., 2000] for more details.

storing the singular vectors and indices into this set of singular vectors at points instead of complete attribute vectors.

This technique is powerful from the point of view of data analysis as well. Multiple points mapped to the same set of singular vectors can be viewed as exhibiting approximately identical behavior (in terms of their attribute vectors). This process is illustrated in Figure 10. The set of singular vectors can be viewed as the dominant cluster behavior and can be used to further characterize system behavior. Techniques similar to this have been explored for identifying principal components for visualizing data with very high dimensionality and for compressing discrete transaction data using semi-discrete transforms (SDD) [Kolda and O'Leary, 1998, Ruth et al., 2000, Yang et al., 2000].

A critical aspect of using lossy compression techniques for data analysis is the selection of appropriate functions over which simplification metrics can be defined. Consider the analysis of a fluid flow simulation with a view to identifying vortices. On a simply connected domain, the velocity flow field consists of two parts — a pure gradient component (with zero curl) and a pure curl component (with zero div). Given an arbitrary vector field V associated with a fluid, vorticity is defined as the curl of V. The process of lossy compression must preserve error with respect to the curl of V while de-emphasizing other parameters. Similar considerations arise in almost all applications. A unified framework relying on a weighted set of attribute vectors (with all appropriate features) can handle such applications well.

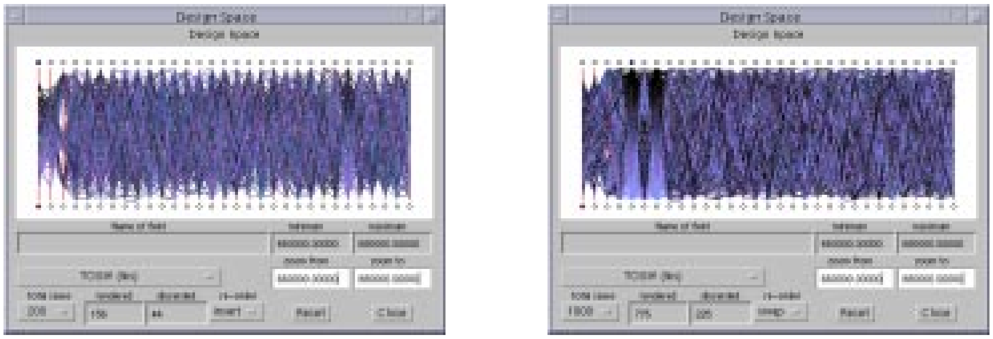


Figure 11: (left) Visualizing 156 aircraft design points in 29 dimensions [Goel et al., 2000]. (right) A rearrangement of variables with corresponding color driver change reveals an interesting association. Figure courtesy C.A. Shaffer (Virginia Tech).

5 Putting It All Together

5.1 Underlying Principles

The operational strength of techniques such as those presented earlier relies on an integration of methodologies for storage, retrieval, and post-processing in data mining. The importance of support for such data-intensive operations is increasingly underscored in scientific circles [Chandy et al., 1998, Moore et al., 1998a, Moore et al., 1998b, Rice and Boisvert, 1996]. In this section, we outline various aspects of data mining in larger scientific contexts with illustrative examples.

Recommender Systems

One of the emerging areas of research in computational science and engineering is the design of powerful programming abstractions for application support systems [Gannon et al., 1998, Grosse, 1996, Saltz et al., 1998]. For example, the LSA system [Gannon et al., 1998] can compose a linear system analyzer by using a plug-and-play paradigm to chain together individual components. This compositional modeling framework requires the knowledge-based selection of solution components for individual application domains — in this case, linear algebra. Such *recommender systems* can help in the natural process of a scientist/engineer making selections among various choices of algorithms/software. They are typically designed off-line by organizing a battery of benchmark problems & algorithm executions [Houstis et al., 2000], and mining it to obtain high-level rules that can form the basis of a recommendation. Data mining thus plays a major role in providing decision support in large-scale simulations.

Interactive Visualization

Concomitant with such tools, visualization of recommendation spaces and providing mechanisms for incremental exploration of data becomes paramount. Fig. 11 describes the interface to VizCraft — a PSE for conceptual design of aircraft [Goel et al., 2000]. The goal of this PSE is to minimize the take-off gross weight (TOGW) for a 250-passenger high speed civil transport (HSCT) [Knill et al., 1999]. There are 29 design variables with 68 constraints in a highly non-convex design space. The scenario in Fig. 11 describes a collection of data points over the 29 dimensions superimposed on each other. As can be seen, a rearrangement of the 'driving variable' reveals an interesting association between two aspects of aircraft design. Coupled with virtual reality devices, interactive visualization also plays a major role in applications such as molecular docking, determining protein secondary structure, working in hazardous environments, and telemedicine.

An important aspect of scientific data mining relates to inlined mining and simulation tasks. It can be argued that if one can identify specific processes (and/or subdomains) that are interesting, then computational resources could be steered towards these processes, while supporting other simulation tasks only in so far as to maintain the fidelity of the interesting phenomena. This concept of *computational steering* [Parker et al., 1997] plays an important role in reducing the computation associated with large-scale simulations. For example, applications such as eukaryotic cell cycle modeling involve tens to hundreds of parameters which have to be dynamically modified, tracked, and tuned to achieve desired metabolic processes. Integration of computation, interaction, and postprocessing is the target of many commercial software ventures.

Data and Experiment Management

The realization of the above two goals relies on efficient data modeling that supports the data generation, data analysis, automatic knowledge acquisition, and inference processes in computational science. One of the main requirements of data modeling involves providing storage for problem populations in a structured way, and enabling management of the execution environment by keeping track of the constraints implied by the physical characteristics of the application. In addition, the quantity of information generated for computational steering and manipulated by recommender systems requires a powerful and adaptable database management system (DBMS) with an open architecture. However, traditional relational and OO models are inadequate because fully extensible functionality is required for an environment that keeps changing not only in the size of the data but also in the schema. For example, bioinformatics applications require specialized data structures and customized tables for each new experimental technique introduced [Buneman et al., 1995]. Such frequent changes to source schema are not well addressed by current systems.

Information Integration

With the rapid emergence of data formats and applications such as bioinformatics supporting a veritable cottage industry of databases, information integration becomes paramount to support holistic scenarios. For example, the query 'Find all information pertaining to human chromosome 22' was considered an impossible query under the Human Genome Project until just a few years back [Buneman et al., 1995]. Information integration is typically addressed by remapping queries to originating sources, introducing a transparency layer of middleware in between data sources, or using other mediator-based schemes. This is thus one of the main issues underlying applications such as health care management [Grimson et al., 2000] and digital libraries [Adam et al., 2000, Moore et al., 1998b].

```
-- table no 1
create table FEATURE (
 name       text,    -- record name (primary key)
 nfeatures  integer, -- no. of attributes identifying this feature
 features   text[],  -- numeric/symbolic/textual identification
 forfile    text     -- file-based feature information
);
```

Figure 12: Example schema for the feature record.

```
Field          | Value
name           | pde54 dom02 fd-itpack-rscg SP2-17
system         | pellpack
comp_db        | linearalgebra
composite_id   | pde54 domain 02 fd-itpack-rscg
perfind_set    | pellpack-std-par-grd
pid            | 1432
sequence_no    | 17
eqparms        | pde #54 parameter set 5
solverseq      | 950x950 proc 4 reduced system cg
rundata        | IBM SP2 with 18 compute nodes
....
nerror         | 3
errornames     | {"max abs error","L1 error","L2 error"}
errorvals      | {"0.0022063","0.00011032","0.00022281"}
```

Figure 13: An (incomplete) instance of performance data from the PDE benchmark.

5.2 Best Practices

Data Modeling and Representation

Significant advances have been made in database support for problem solving environments (PSEs) and computational steering. We present concepts from the PYTHIA framework [Houstis et al., 2000] that aids in the rapid prototyping of recommender systems for computational science. In order to facilitate the storage and execution of experiments, the input specification of a problem (e.g. ODEs, PDEs) is decomposed into a set of database tables in PYTHIA. There is a one-to-one mapping between the components of the problem specification and the basic entities in PYTHIA's schema. Features of problem components are also modeled by other tables and tables representing relations are used for designating the constraints in associating features with basic entities. Fig. 12 provides a schema for the definition of a PDE feature which can be instantiated along with associated connections to other records. Experiments are also managed by yet another table, in that each experiment record holds all the necessary information for executing a collection of problems in a specific system. PYTHIA communicates with the host PSE via devices such as I/O files and

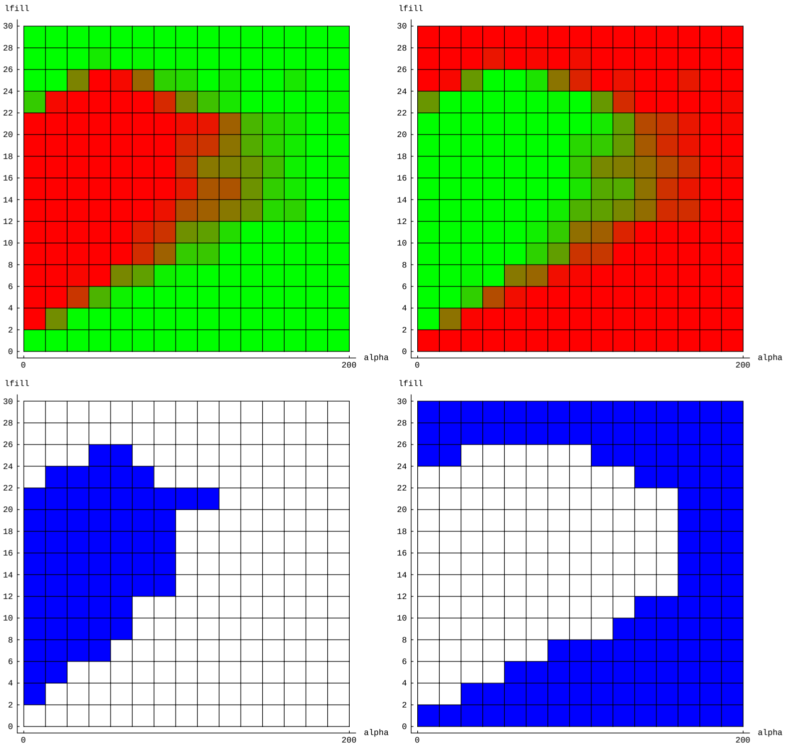


Figure 14: Mining and visualizing recommendation spaces for `gmres` (top left) and direct solve (top right) on nearly singular PDE problems; The intensity in the colormap reflects the number of problem instances in a bucket for which the given algorithm was superior; 90% confidence regions mined automatically for `gmres` (bottom left) and direct solve (bottom right), for `droptol` $= 0$, by the method described in [Ramakrishnan and Ribbens, 2000] are also shown. For these regions, the given methods were superior for at least 90% of the problem instances.

software buses. In order to avoid redundancy in the implementation, an experiment record uses foreign keys for representing the logical connections with the basic records. Tables that store performance data are used for saving selected parts from the output files produced by running these experiments. Atomic entities will be domain specific since they represent the problem definition objects of a targeted domain, but the performance and knowledge-related data schema extend easily to other problem domains. Such a representation for performance information is depicted in Fig. 13. The performance analysis of algorithms proceeds with respect to user specified criteria, such as the total time taken for solving the linear system that arises from the discretization of a PDE. Generalization of such performance rankings produces recommendation spaces and regions (via, say inductive logic programming) that can help in selecting algorithms for newly presented problems. Fig. 14 describes a comparison between an iterative solver (`gmres`) and a direct solver by mining $45,000$ PDE solves on nearly singular problems. The shape of the induced recommendation spaces provides insight into the relative efficacies of the methods. For example, Fig. 14 shows that when the `droptol` parameter for the linear solver is zero, the `lfill` parameter must fall within a relatively narrow interval in order for the iterative method to be the preferred choice. For more details, we refer the reader to [Ramakrishnan and Ribbens, 2000].

The SAL system presented in Fig. 7 achieves a similar objective by combining the interpretation of structures in physical fields with physical knowledge such as locality and linear superposability, thus allowing control placement and parameters to be designed in an *explainable* manner [Bailey-Kellogg and Zhao, 1999]. In other words, data mining constitutes a fundamental methodology in control design.

Integrating Numeric, Symbolic, and Geometric Information

Qualitative analysis of dynamical systems originated from the MaC Project at MIT [Abelson et al., 1989]; the approach taken is to support intelligent simulation by representations and mechanisms that autonomously design and monitor complex physical systems through appropriate mixtures of numerical and symbolic computing and knowledge-based methods. Programs developed in this manner have been shown to automatically prepare numerical experiments from high-level descriptions, and exploit techniques like imagistic reasoning (see discussion on the SAL system in Section 3) and computer vision to identify promising areas for future experiments. Mining qualitative models using the QSIM representation is undertaken in [Hau and Coiera, 1997]; this work resembles incorporating tighter constraints on induction techniques like ILP. Such integration of multi-modal reasoning will only become more important with increased emphasis on computational science and the replacement of many wet-lab procedures by simulation.

6 Future Research Issues

6.1 Mining when Data is Scarce

While massive databases catalog and provide access to petabytes of archived field data or measurements (e.g. sky surveys [Fayyad et al., 1996], the human genome project [Ridley, 2000]), much scientific data is gathered from sophisticated mathematical models, codes, and simulations. For domains such as gas turbine dynamics simulation and aircraft

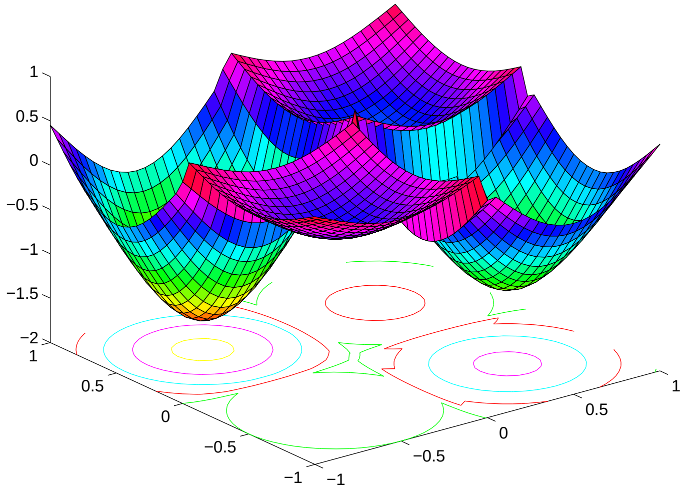


Figure 15: A 2D `pocket` function. Notice the global minimum in the $[-1, 0] \times [0, 1]$ quadrant. Data mining can help reveal the locations of the pockets; one structure-exploiting approach, based on learning evaluation functions for further optimization, is to correlate the magnitude of local minima with the starting point for descent algorithms [Boyan and Moore, 1998].

design, such complex simulations require days, weeks, or even years on petaflops class computing systems. From a data mining point of view, this raises interesting issues not present in many other commercial (and even scientific) domains. First, these applications are characterized not by an abundance of data, but rather by a scarcity of data (owing to the cost and time involved in conducting simulations).

de Boor's function

Visualize the n-dimensional hypercube defined by $x_i \in [-1, 1], i = 1 \cdots n$, with the n-sphere of radius 1 centered at the origin $(\Sigma {x_i}^2 \leq 1)$ embedded inside it. Notice that the ratio of the volume of the cube (2^n) to that of the sphere $(\pi^{n/2}/(n/2)!)$ grows unboundedly with n. In other words, the volume of a high-dimensional cube is concentrated in its corners (a counterintuitive notion at first). Carl de Boor exploited this property to design a difficult-to-optimize function which assumes a *pocket* in each corner of the cube (Fig. 15), that is just outside the sphere [Rice, 1992]. It is easily seen that the function has 2^n local minima and the goal of mining in this scenario is to be able to (i) identify the locations of the pockets and (ii) obtain some indication of where the 'biggest dip' is located. In real-world scientific domains, n is large (say, 30 which means it will take more than half million points to just represent the corners of the n-cube!) and global optimization algorithms require that the `pocket` function be evaluated at a large number of points. Arguably, the task is simplified considerably if one has a symbolic form of the `pocket` function to analyze, but such information amenable to *apriori* analyses is typically not available.

Aircraft Design

This problem is exacerbated in domains such as aircraft design (see Fig. 11). Fig. 16 shows a cross-section of the design space for the representative problem described earlier in Section

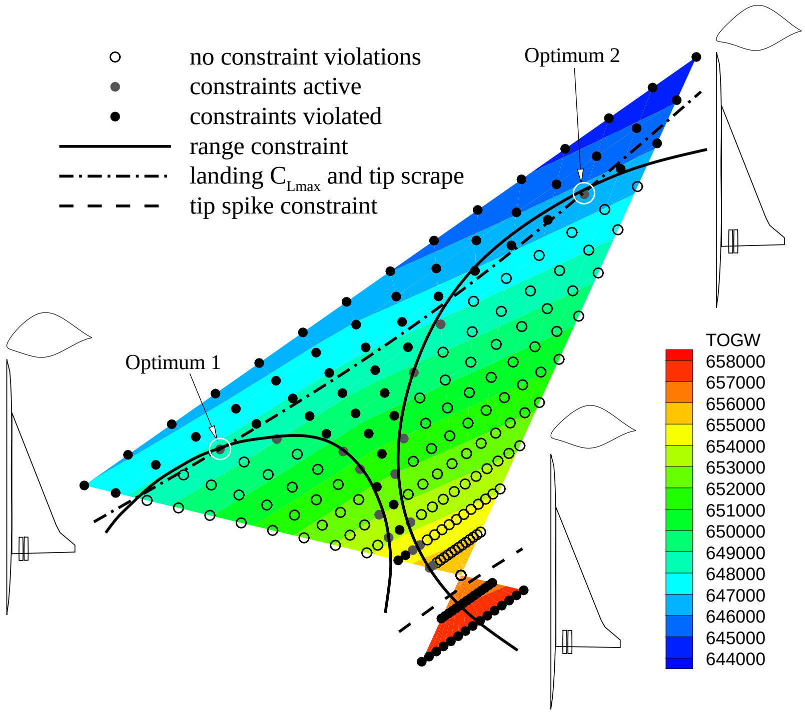


Figure 16: A slice of an aircraft design space through three design points [Knill et al., 1999]. Notice the nonconvexity of the feasible design space induced by the various constraints. Figure courtesy Layne T. Watson (Virginia Tech).

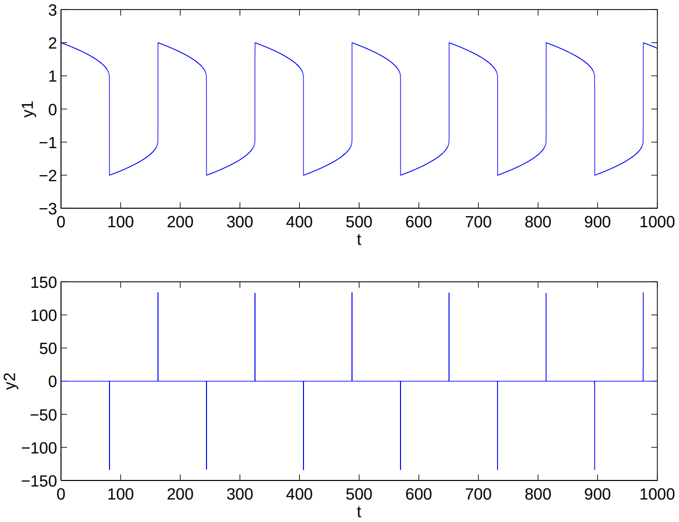


Figure 17: A contour plot of the Van der Pol oscillator of dimension 2 used to model radio circuits, heart beats, and pulsating stars. Notice that the two components of the solution evolve on multiple time scales. Solving this ODE requires switching between stiff and non-stiff methods many times during the domain of integration. Data mining can help mine switching policies to prevent *thrashing* between the two families of algorithms.

5.1. Frequently, the designer will change some aspect of a nominal design point, and run a simulation to see how the change affects the objective function and various constraints dealing with aircraft geometry and performance/aerodynamics. Such a simple approach will be infeasible in the foreseeable future, because months, often years, are required to explore a high dimensional design space, even at low fidelity. As new designs evolve and mature to fit new uses, the need to quickly explore many design and parameter choices becomes increasingly important. Ideally, the design engineer would like a high-level mining system to identify the pockets that contain good designs and which merit further consideration; traditional tools from optimization and approximation theory can then be applied to fine-tune such preliminary analyses.

6.2 Design of Experiments

More importantly, in contrast to business domains, the computational scientist has complete control over the data acquisition process (regions of the design space where data can be collected). A cheap surrogate model can then be constructed that can serve as an alternative starting point for data mining. This methodology, known quite simply as 'Design and Analysis of Computer Experiments (DACE)', is prevalent in the statistical design literature [Sacks et al., 1989] but hasn't received much attention in the data mining community. The goal is to extract useful patterns from the surrogate models, rather than the original, costly codes (or field data, which cannot be controlled). Considerable attention has been devoted to the use of sampling for mining very structured patterns, such as association rules [Kivinen and Mannila, 1994] (this work, however, does not address the issue of 'where to sample.'). Similar aspects are currently being addressed in a project relating to designing experiments for biological molecule crystallization [Hennessy et al., 1994a, Hennessy et al., 1994b].

6.3 Mining 'On-The-Fly'

Inducing control policies that capture patterns on a dynamic scale can help in runtime recommendation of algorithms, algorithm switching, and large scale application composition. The goal here is to integrate the data collection and data mining stages in computational science by incremental refinement techniques such as reinforcement learning [Kaelbling et al., 1996]. For example, consider the Van der Pol relaxed system of dimension 2 [Zwillinger, 1992]:

$$x'' - \mu(1 - x^2)x' + x = 0$$

graphed in Fig. 17 that has a limit cycle whenever $\mu > 0$. For this ODE, the value of x increases for small values and decreases for large values. This problem is complicated because it alternates between being stiff and non-stiff several times in the region of interest, causing difficulties for traditional ODE software. By modeling it as a non-deterministic, stationary system, and learning the utility of taking certain actions (e.g. 'switch algorithm') in various states, control policies can be induced that help automate the 'type-insensitivity' of mathematical software.

6.4 Mining in Distributed and Parallel Environments

Distributed and parallel platforms form an integral part of most data collection and analysis frameworks. Data mining has several important roles to play in such frameworks. In addition to mining with a view to understanding the scientific and engineering processes, mining can play a critical role in rendering many problems tractable. It is impossible to communicate large repositories of geographically distributed data to a client for analysis. Distributed mining can analyze local data and progressively communicate information as required by the client. This view of distributed data mining is consistent with ideas of progressive compression and transmission.

Experiments such as SETI present an interesting computing paradigm for large scale data analysis. Extensions of this paradigm relying on mobile code for analysis present significant untapped potential for scientific data mining. This motivates a truly asynchronous framework – one in which analysis tasks are farmed off and results incorporated into a collective mined information set, if and when they become available. A similar approach to incorporating data as it becomes available to refine a hypothesis follows from the paradigm of dynamic data mining.

7 Concluding Remarks

The field of scientific data mining is still in its infancy. As scientists address various problems associated with large scale simulation and data collection, the associated data handling and analysis problems will become more acute. We envision a balance between, and tight integration of, inlined analysis and computing tasks. A consolidation of various approaches from statistical analysis, information theory, and numerical methods will result in a theoretically sound set of tools for a range of common analysis tasks. While data analysis has already seen some success in understanding scientific and engineering processes, we envision a much greater impact in the near future in domains such as bioinformatics, astrophysics, and materials processing.

One of the major impending developments in computing will be the ubiquitous use of embedded systems (see, for instance, [Estrin et al., 2000]). It is estimated that within the next decade, over 98% of all computing devices will be embedded in physical environments. Such sensing and actuation environments will require a new class of techniques capable of dynamic data analysis in faulty, distributed frameworks. Widespread use of MEMS devices will pose formidable data analysis and handling problems. Examples of such applications include active structures and surfaces. Active structures are capable of sensing environmental changes to adapt themselves. Experiments with active structures involve adaptive chassis for automobiles and earthquake tolerant buildings. Active surfaces are used on aircraft wings to minimize drag by controlling turbulence in the surface. Such devices are based on very fine grain processing and communication elements with hard deadlines for analysis and actuation. The notion of *anytime* analysis, where the analysis task can be stopped at any point of time (real time constraints) and results up to best current estimates are made available, will be critical.

Finally, we expect the dichotomy between commercial and scientific data analysis to blur. Instead, a classification along dimensions of discrete and continuous data based on an underlying generating model will be more suitable. This will result in a rich set of tools for analyzing data from a varied set of sources.

References

[Abbott et al., 1997] Abbott, C., Berry, M., Comiskey, J., Gross, L., and Luh, H.-K. (1997). Parallel Individual-Based Modeling of Everglades Deer Ecology. *IEEE Computational Science and Engineering*, Vol. 4(4).

[Abelson et al., 1989] Abelson, H., Eisenberg, M., Halfant, M., Katzenelson, J., Sacks, E., Sussman, G., Wisdom, J., and Yip, K. (May 1989). Intelligence in Scientific Computing. *Communications of the ACM*, Vol. 32:pp. 546–562.

[Adam et al., 2000] Adam, N., Atluri, V., and Adiwijaya, I. (2000). SI in Digital Libraries. *Communications of the ACM*, Vol. 43(6):pp. 64–72.

[Agrawal et al., 1993] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining Associations between Sets of Items in Large Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press.

[Anderson et al., 1999] Anderson et al., D. (1999). SETI@home: The Search for Extraterrestrial Intelligence. Technical report, Space Sciences Laboratory, University of California at Berkeley. URL: `http://setiathome.ssl.berkeley.edu/`.

[Bailey-Kellogg and Zhao, 1999] Bailey-Kellogg, C. and Zhao, F. (1999). Influence-Based Model Decomposition. *Proc. of the National Conference on Artificial Intelligence (AAAI'99).*

[Bajaj et al., 1999] Bajaj, C., Pascucci, V., and Zhuang, G. (1999). Single Resolution Compression of Arbitrary Triangular Meshes with Properties. *Computational Geometry: Theory and Applications*, Vol. 14:pp. 167–186.

[Barbara et al., 1997] Barbara, D., DuMouchel, W., Faloutsos, C., Haas, P., Hellerstein, J., Ioannidis, Y., Jagadish, H., Johnson, T., Ng, R., Poosala, V., Ross, K., and Sevcik, K. (December 1997). The New Jersey Data Reduction Report. *Bulletin of the IEEE Technical Committee on Data Engineering*, Vol. 20(4):pp. 3–45.

[Barnes and Hut, 1986] Barnes, J. and Hut, P. (1986). A Hierarchical O(n log n) Force Calculation Algorithm. *Nature*, Vol. 324.

[Bennett et al., 1996] Bennett, C., Banday, A., Gorski, K., Hinshaw, G., Jackson, P., Keegstra, P., Kogut, A., Smoot, G., Wilkinson, D., and Wright, E. (1996). 4-Year COBE DMR Cosmic Microwave Background Observations: Maps and Basic Results. *Astrophysical Journal*, Vol. 464, L1.

[Berlin and Gabriel, 1997] Berlin, A. and Gabriel, K. (1997). Distributed MEMS: New Challenges for Computation. *IEEE Computational Science and Engineering*, Vol. 4(1):pp. 12–16.

[Berry et al., 1994] Berry, M., Comiskey, J., and Minser, K. (1994). Parallel Analysis of Clusters in Landscape Ecology. *IEEE Computational Science and Engineering*, Vol. 1(2):pp. 24–38.

[Berry et al., 1999] Berry, M., Drmac, Z., and Jessup, E. (1999). Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*, Vol. 41(2):pp. 335–362.

[Berry et al., 1995] Berry, M., Dumais, S., and O'Brien, G. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, Vol. 37(4):pp. 573–595.

[Berry and Fierro, 1996] Berry, M. and Fierro, R. (1996). Low-rank orthogonal decompositions for information retrieval applications. *Numerical Linear Algebra with Applications*, Vol. 3(4):pp. 301–328.

[Berry et al., 1996] Berry, M., Flamm, R., Hazen, B., and MacIntyre, R. (1996). LUCAS: A System for Modeling Land-Use Change. *IEEE Computational Science and Engineering*, Vol. 3(1):pp. 24–35.

[Bersanelli et al., 1996] Bersanelli et al., B. (Feb 1996). The PLANCK Phase A Study Report. Technical Report Report D/SCI(96)3, ESA. URL: `http://astro.estec.esa.nl/SA-general/Projects/Planck/report/report.html`.

[Bezdek, 1981] Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.

[Blum and Rivest, 1992] Blum, A. and Rivest, R. (1992). Training a 3-node Neural Network is NP-Complete. *Neural Networks*, 5(1):pp. 117–127.

[Board and Schulten, 2000] Board, J. and Schulten, K. (2000). The Fast Multipole Algorithm. *Computing in Science and Engineering*, Vol. 2(1).

[Böhringer et al., 1997] Böhringer, K., Donald, B., MacDonald, N., Kovacs, G., and Suh, J. (1997). Computational Methods for Design and Control of MEMS Micromanipulator Arrays. *IEEE Computational Science and Engineering*, Vol. 4(1):pp. 17–29.

[Boyan and Moore, 1998] Boyan, J. and Moore, A. (1998). Learning Evaluation Functions for Global Optimization and Boolean Satisfiability. In *Fifteenth National Conference on Artificial Intelligence (AAAI'98).*

[Bratko and Muggleton, 1995] Bratko, I. and Muggleton, S. (November 1995). Applications of Inductive Logic Programming. *Communications of the ACM*, Vol. 38(11):pp. 65–70.

[Brodley and Friedl, 1999] Brodley, C. and Friedl, M. (1999). Identifying Mislabeled Training Data. *Journal of Artificial Intelligence Research*, Vol. 11:pp. 131–167.

[Buneman et al., 1995] Buneman, P., Davidson, S., Hart, K., Overton, C., and Wong, L. (1995). A Data Transformation System for Biological Data Sources. In *Proceedings of the VLDB Conference*.

[Chandy et al., 1998] Chandy, K., Bramley, R., Char, B., and Reynders, J. (1998). Report of the NSF Workshop on Problem Solving Environments and Scientific IDEs for Knowledge, Information and Computing (SIDEKIC'98). Technical report, Los Alamos National Laboratory.

[Cheeseman and Stutz, 1996] Cheeseman, P. and Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Practice. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press.

[Craven and Shavlik, 1993] Craven, M. and Shavlik, J. (1993). Learning to Represent Codons: A Challenge Problem for Constructive Induction. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1319–1324. Chambery, France.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Controls, Signals, and Systems*, Vol. 2:pp. 303–314.

[de Bernardis et al., 2000] de Bernardis et al., P. (2000). A Flat Universe from High-Resolution Maps of the Cosmic Microwave Background Radiation. *Nature*, Vol. 404. URL: `http://www.physics.ucsb.edu/~boomerang/papers.html`.

[Drashansky et al., 1999] Drashansky, T., Houstis, E., Ramakrishnan, N., and Rice, J. R. (1999). Networked Agents for Scientific Computing. *Communications of the ACM*, Vol. 42(3):pp. 48–54.

[Dzeroski, 1996] Dzeroski, S. (1996). Inductive Logic Programming and Knowledge Discovery in Databases. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 117–152. AAAI/MIT Press.

[Estrin et al., 2000] Estrin, D., Govindan, R., and Heidemann, J. (2000). Embedding the Internet: Introduction. *Communications of the ACM*, Vol. 43(5):pp. 38–42.

[Fayyad et al., 1996] Fayyad, U., Haussler, D., and Stolorz, P. (1996). Mining Scientific Data. *Communications of the ACM*, Vol. 39(11):pp. 51–57.

[Fayyad et al., 1993] Fayyad, U., Weir, N., and Djorgovski, S. (1993). Automated Cataloging and Analysis of Ski Survey Image Databases: The SKICAT System. In *Proc. of the Second Int. Conf. on Information and Knowledge Management*, pages 527–536, Washington DC.

[Forbus, 1997] Forbus, K. (1997). Qualitative Reasoning. In Tucker, A., editor, *The Computer Science and Engineering Handbook*, pages 715–730. CRC Press.

[Frazier and Pitt, 1996] Frazier, M. and Pitt, L. (1996). Classic Learning. *Machine Learning*, Vol. 25(2–3):pp. 151–193.

[Fu, 1999] Fu, L. (1999). Knowledge Discovery Based on Neural Networks. *Communications of the ACM*, Vol. 42(11):pp. 47–50.

[Fukunaga, 1990] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition.* Academic Press.

[Gage et al., 1995] Gage, P., Kroo, I., and Sobieski, I. (Nov. 1995). A Variable-Complexity Genetic Algorithm for Topological Design. *AIAA Journal*, Vol. 33(11):pp. 2212–2217.

[Gallman and Kroo, 1996] Gallman, J. and Kroo, I. (Jan. 1996). Structural Optimization for Joined Wing Synthesis. *Journal of Aircraft*, Vol. 33(1):pp. 214–223.

[Gannon et al., 1998] Gannon, D., Bramley, B., Stuckey, T., Villacis, J., Balasubramanian, J., Akman, E., Breg, F., Diwan, S., and Govindaraju, M. (1998). Component Architectures for Distributed Scientific Problem Solving. *IEEE Computational Science and Engineering*, Vol. 5(2):pp. 50–63.

[Ganti et al., 1999a] Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999a). CACTUS: Clustering Categorical Data Using Summaries. *Proc. Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 73–83.

[Ganti et al., 1999b] Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999b). Mining Very Large Databases. *IEEE Computer*, Vol. 32(8):pp. 38–45.

[Gibson et al., 1998] Gibson, D., Kleinberg, J., and Raghavan, P. (1998). Clustering Categorical Data: An Approach Based on Dynamical Systems. *Proc. 24th Int. Conf. on Very Large Databases.*

[Goel et al., 2000] Goel, A., Baker, C., Shaffer, C., Grossman, B., Haftka, R., Mason, W., and Watson, L. (2000). VizCraft: A Problem Solving Environment for Configuration Design of a High Speed Civil Transport. *Computing in Science and Engineering.* to appear.

[Goel et al., 1999] Goel, A., Phanouriou, C., Kamke, F., Ribbens, C., Shaffer, C., and Watson, L. (1999). WBCSim: A Prototype Problem Solving Environment for Wood-Based Composites Simulations. *Engineering with Computers*, Vol. 15(2):pp. 198–210.

[Grimson et al., 2000] Grimson, J., Grimson, W., and Hasselbring, W. (2000). The SI Challenge in Health Care. *Communications of the ACM*, Vol. 43(6):pp. 48–55.

[Grosse, 1996] Grosse, E. (1996). Network Programming and CSE. *IEEE Computational Science & Engineering*, Vol. 3(2).

[Han et al., 1999] Han, J., Lakshmanan, L., and Ng, R. T. (1999). Constraint-Based, Multidimensional Data Mining. *IEEE Computer*, Vol. 32(8):pp. 46–50.

[Hau and Coiera, 1997] Hau, D. and Coiera, E. (1997). Learning Qualitative Models of Dynamic Systems. *Machine Learning*, Vol. 26(2–3):pp. 177–211.

[Hellerstein et al., 1999] Hellerstein, J., Avnur, R., Chou, A., Hidber, C., Olston, C., Raman, V., Roth, T., and Haas, P. J. (1999). Interactive Data Analysis: The Control Project. *IEEE Computer*, Vol. 32(8):pp. 51–59.

[Hennessy et al., 1994a] Hennessy, D., Gopalakrishnan, V., Buchanan, B., Rosenberg, J., and Subramanian, D. (1994a). Induction of Rules for Biological Macromolecule Crystallization. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, pages 179–187.

[Hennessy et al., 1994b] Hennessy, D., Gopalakrishnan, V., Buchanan, B., and Subramanian, D. (1994b). The Crystallographer's Assistant. In *Proceedings of AAAI'94*.

[Houstis et al., 2000] Houstis, E., Verykios, V., Catlin, A., Ramakrishnan, N., and Rice, J. (2000). PYTHIA II: A K/DB System for Recommending/Testing Scientific Sofware. *ACM Transactions on Mathematical Software*, Vol. 26(2). to appear.

[Hu, 1996] Hu, W. (October 1996). An Introduction to the Cosmic Microwave Background. Technical report, Institute for Advanced Study, School of Natural Sciences, Olden Lane, Princeton, NJ 08540. URL: `http://www.sns.ias.edu/~whu/beginners/introduction.html`.

[Huang and Zhao, 1998] Huang, X. and Zhao, F. (1998). Finding Structures in Weather Maps. Technical Report OSU-CISRC-3/98-TR11, Department of Computer and Information Science, Ohio State University.

[Imielinski and Mannila, 1996] Imielinski, T. and Mannila, H. (1996). A Database Perspective on Knowledge Discovery. *Communications of the ACM*, Vol. 39(11):pp. 58–64.

[Jiang et al., 1999] Jiang, J., Berry, M., Donato, J., and Ostrouchov, G. (November 1999). Mining Consumer Product Data Via Latent Semantic Indexing. *Intelligent Data Analysis*, Vol. 3(5):pp. 377–398.

[Jordan and Bishop, 1997] Jordan, M. and Bishop, C. (1997). Neural Networks. In Tucker, A., editor, *The Computer Science and Engineering Handbook*, pages 536–556. CRC Press.

[Kaelbling et al., 1996] Kaelbling, L., Littman, M., and Moore, A. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, Vol. 4:pp. 237–285.

[Karypis et al., 1999] Karypis, G., Han, E.-H., and Kumar, V. (1999). Chameleon: Hierarchical Clustering Using Dynamic Modeling. *IEEE Computer*, Vol. 32(8):pp. 68–75.

[Kietz and Morik, 1994] Kietz, J.-U. and Morik, K. (1994). A Polynomial Approach to the Constructive Induction of Structured Knowledge. *Machine Learning*, Vol. 14(1):pp. 193–217.

[Kivinen and Mannila, 1994] Kivinen, J. and Mannila, H. (1994). The Power of Sampling in Knowledge Discovery. In *Proceedings of PODS' 1994*, pages 77–85.

[Knill et al., 1999] Knill, D., Giunta, A., Baker, C., Grossman, B., Mason, W., Haftka, R., and Watson, L. (1999). Response Surface Models Combining Linear and Euler Aerodynamics for Supersonic Transport Design. *Journal of Aircraft*, Vol. 36(1):pp. 75–86.

[Knorr and Ng, 1996] Knorr, E. and Ng, R. (1996). Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8:pp. 884–897.

[Kolda and O'Leary, 1998] Kolda, T. and O'Leary, D. (1998). A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval. *ACM Transactions on Information Systems*, Vol. 16(4):pp. 322–346.

[Koperski et al., 1998] Koperski, K., Han, J., and Adhikary, J. (1998). Mining Knowledge in Geographical Data. *Communications of the ACM.* accepted in the Special Issue on Spatial Database Systems; will be redirected to the Special Issue on Spatial Database Systems, *IEEE Computer.*

[Korn et al., 1997] Korn, F., Jagadish, H., and Faloutsos, C. (May 1997). Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In *ACM SIGMOD*, pages 289–300, Tucson, AZ. URL: `ftp://olympos.cs.umd.edu/pub/TechReports/sigmod97.ps`.

[Langley et al., 1990] Langley, P., Simon, H., and Bradshaw, G. (1990). Heuristics for Empirical Discovery. In Shavlik, J. and Dietterich, T., editors, *Readings in Machine Learning*, pages 356–372. Morgan Kaufmann Publishers, Inc.

[Lee et al., 1998] Lee et al., A. (1998). MAXIMA: An Experiment to Measure Temperature Anisotropy in the Cosmic Microwave Background. In *Proceedings from 3K Cosmology from Space*, Rome, Italy.

[Lei et al., 1997] Lei, M., Kleinstreuer, L., and Archie, J. (1997). Hemodynamic Simulations and Computer-Aided Designs of Graft-Artery Junctions. *J. Biomech. Eng.*, Vol. 119:pp. 343–348.

[Leisawitz, 1999] Leisawitz, D. (June 1999). COBE Analysis Software, Version 4.1. Technical report, Astrophysics Data Facility, NASA Goddard Space Flight Center, Greenbelt, MD 20771, USA. URL: `http://space.gsfc.nasa.gov/astro/cobe/cgis.html`.

[Letsche and Berry, 1997] Letsche, T. and Berry, M. (1997). Large-Scale Information Retrieval with Latent Semantic Indexing. *Information Sciences - Applications*, Vol. 100:pp. 105–137.

[Li and Biswas, 1995] Li, C. and Biswas, G. (1995). Knowledge-Based Scientific Discovery in Geological Databases. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pages 204–209. ACM Press.

[Liu, 1996] Liu, S. (1996). Vein Graft Engineering. *Advances in Bioengineering*, Vol. 33:pp. 473–474.

[Mitchell, 1982] Mitchell, T. (1982). Generalization as Search. *Artificial Intelligence*, Vol. 18(2):pp. 203–226.

[Moore, 1998] Moore, A. (1998). Very Fast EM-based Mixture Model Clustering using Multiresolution kd-trees. Technical report, Carnegie Mellon University. URL: `http://ranger.phys.cmu.edu/users/nichol/KDI/refs.html`.

[Moore and Lee, 1998] Moore, A. and Lee, M. (1998). Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets. *Journal of Artificial Intelligence Research*, Vol. 8:pp. 67–91.

[Moore et al., 1998a] Moore, R., Baru, C., Marciano, R., Rajasekar, A., and Wan, M. (1998a). Data-Intensive Computing. In Kesselman, C. and Foster, I., editors, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

[Moore et al., 1998b] Moore, R., Prince, T., and Ellisman, M. (1998b). Data-Intensive Computing and Digital Libraries. *Communications of the ACM*, Vol. 41(11):pp. 56–62.

[Muggleton, 1999] Muggleton, S. (1999). Scientific Knowledge Discovery using Inductive Logic Pogramming. *Communications of the ACM*, Vol. 42(11):pp. 42–46.

[Muggleton and Feng, 1990] Muggleton, S. and Feng, C. (1990). Efficient Induction of Logic Programs. In Arikawa, S., Goto, S., Ohsuga, S., and Yokomori, T., editors, *Proceedings of the First International Conference on Algorithmic Learning Theory*, pages 368–381. Japanese Society for Artificial Intelligence, Tokyo.

[Murthy, 1998] Murthy, S. (1998). Qualitative Reasoning at Multiple Resolutions. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 296–300. American Association for Artificial Intelligence.

[Nakano, 1997] Nakano, A. (1997). Fuzzy Clustering Approach to Hierarchical Molecular Dynamics Simulation of Multiscale Materials Phenomena. *Comput. Phys. Commun.*, Vol. 105:pp. 139–150.

[Nakano, 1999] Nakano, A. (1999). A Rigid-Body Based Multiple Time-Scale Molecular Dynamics Simulation of Nanophase Materials. *Int. J. High Performance Comput. Appl.*, Vol. 13(2):pp. 154–162.

[Nakano et al., 1995] Nakano, A., Kalia, R., and Vashishta, P. (1995). Dynamics and Morphology of Brittle Cracks: A Molecular-Dynamics Study of Silicon Nitride. *Phys. Rev. Lett.*, Vol. 75:pp. 3138–3141.

[Nakano et al., 1999] Nakano, A., Kalia, R., and Vashishta, P. (1999). Scalable Molecular-Dynamics, Visualization, and Data-Management Algorithms for Material Simulations. *Computing in Science and Engineering*, Vol. 1(5).

[Nayak, 1992] Nayak, P. (1992). Order of Magnitude Reasoning using Logarithms. In *Proceedings of KR-92*.

[Ng and Han, 1994] Ng, R. and Han, J. (1994). Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the Twentieth International Conference on Very Large Databases*, pages 144–155.

[Nguyen and Huang, 1994] Nguyen, T. and Huang, T. (1994). Evolvable 3D Modeling for Model-Based Object Recognition Systems. In Kinnear, Jr., K. E., editor, *Advances in Genetic Programming*, chapter 22, pages 459–475. MIT Press.

[Noor, 1997] Noor, A. (1997). Computational Structural Mechanics. In Tucker, A., editor, *The Computer Science and Engineering Handbook*, pages 847–866. CRC Press.

[Opitz and Shavlik, 1997] Opitz, D. and Shavlik, J. (1997). Connectionist Theory Refinement: Genetically Searching the Space of Network Topologies. *Journal of Artificial Intelligence Research*, Vol. 6:pp. 177–209.

[Parker et al., 1997] Parker, S., Johnson, C., and Beazley, D. (1997). Computational Steering Software Systems and Strategies. *IEEE Computational Science and Engineering*, pages 50–59.

[Peikert and Roth, 1999] Peikert, R. and Roth, M. (Oct. 28, 1999). The "Parallel Vectors" Operator - A Vector Field Visualization Primitive. In *IEEE Visualization '99 Conference*, San Francisco, CA.

[Preston et al., 2000] Preston, E., SaMartins, J., Rundle, J., Anghel, M., and Klein, W. (2000). Models of Earthquake Faults with Long-Range Stress Transfer. *Computing in Science & Eng. (Special Issue on Computational Earth System Science)*, Vol. 2(3).

[Ramakrishnan and Grama, 1999] Ramakrishnan, N. and Grama, A. (1999). Data Mining: From Serendipity to Science (Guest Editors' Introduction to the Special Issue on Data Mining). *IEEE Computer*, Vol. 32(8):pp. 34–37.

[Ramakrishnan and Ribbens, 2000] Ramakrishnan, N. and Ribbens, C. (2000). Mining and Visualizing Recommendation Spaces for Elliptic PDEs with Continuous Attributes. *ACM Transactions on Mathematical Software*, Vol. 26(2). to appear.

[Rice, 1992] Rice, J. (1992). Learning, Teaching, Optimization and Approximation. In Houstis, E., Rice, J., and Vichnevetsky, R., editors, *Expert Systems for Scientific Computing*, pages 89–123. North-Holland, Amsterdam.

[Rice, 1995] Rice, J. (1995). Potential PYTHIA Development/Research Thrusts. Internal Memo, Pellpack Research Group, Department of Computer Sciences, Purdue University.

[Rice and Boisvert, 1996] Rice, J. and Boisvert, R. (1996). From Scientific Software Libraries to Problem-Solving Environments. *IEEE Computational Science & Engineering*, Vol. 3(3):pp. 44–53.

[Ridley, 2000] Ridley, M. (2000). *Genome: The Autobiography of a Species in 23 Chapters.* Harpercollins.

[Rossignac, 1999] Rossignac, J. (January-March 1999). Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 5(1).

[Rubin et al., 2000] Rubin, E., Dietz, R., Lingam, S., Chanat, J., Speir, C., Dymond, R., Lohani, V., Kibler, D., Bosch, D., Shaffer, C., Ramakrishnan, N., and Watson, L. (2000). From Landscapes to Waterscapes: A PSE for Landuse Change Analysis. In *Proceedings of the 16th IMACS World Congress.* to appear.

[Rundle, 2000] Rundle, J. (2000). Computational Earth System Science. *Computing in Science and Engineering*, Vol. 2(3).

[Ruspini, 1969] Ruspini, E. (1969). A New Approach to Clustering. *Inf. Control*, Vol. 15:pp. 22–32.

[Ruth et al., 2000] Ruth, P., Grama, A., Ramakrishnan, N., and Kumar, V. (2000). Compression of and Pattern Extraction from Large Sets of Attribute Vectors. Technical report, Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907.

[Sacks et al., 1989] Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, Vol. 4(4):pp. 409–435.

[Sadarjoen and Post, 1999] Sadarjoen, I. and Post, F. (May 26-28, 1999). Geometric Methods for Vortex Extraction. In *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, Vienna, Austria.

[Saltz et al., 1998] Saltz, J., Sussman, A., Graham, S., Demmel, J., Baden, S., and Dongarra, J. (1998). Programming Tools and Environments. *Communications of the ACM*, Vol. 41(11):pp. 64–73.

[Sarin and Sameh, 1998] Sarin, V. and Sameh, A. (Jan. 1998). An Efficient Iterative Method for the Generalized Stokes Problem. *SIAM Journal on Scientific Computing*, Vol. 19(1):pp. 206–226.

[Schulze-Kremer, 1999] Schulze-Kremer, S. (1999). Discovery in the Human Genome Project. *Communications of the ACM*, Vol. 42(11):pp. 62–64.

[Semtner, 2000] Semtner, A. (2000). Ocean and Climate Modeling. *Communications of the ACM*, Vol. 43(4):pp. 80–89.

[Shen et al., 1996] Shen, W.-M., Ong, K., Mitbander, B., and Zaniolo, C. (1996). Metaqueries for Data Mining. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors, *Advances in Knowledge Discovery and Data Mining*, pages 375–398. AAAI/MIT Press.

[Soria and Cantwell, 1992] Soria, J. and Cantwell, B. (1992). Identification and Classification of Topological Structures in Free Shear Flows. In *Eddy Structure Identification in Free Turbulent Shear Flows.* Kluwer Academic Publishers.

[Srinivasan and King, 1999a] Srinivasan, A. and King, R. (1999a). Feature Construction with Inductive Logic Programming: A Study of Quantitative Predictions of Biological Activity Aided by Structural Attributes. *Data Mining and Knowledge Discovery*, Vol. 3(1):pp. 37–57.

[Srinivasan and King, 1999b] Srinivasan, A. and King, R. (1999b). Using Inductive Logic Programming to Construct Structure-Activity Relationships. In Gini, G. and Katrizsky, A., editors, *Predictive Toxicology of Chemicals: Experiences and Impact of AI Tools (Papers from the 1999 AAAI Spring Symposium)*, pages 64–73. AAAI Press, Menlo Park, CA.

[Stolorz et al., 2000] Stolorz, P., Blom, R., Crippen, R., and Dean, C. (2000). QUAKEFINDER: Photographing Earthquakes from Space. Technical report, Machine Learning Systems Group, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. URL: `http://www-aig.jpl.nasa.gov/public/mls/quakefinder/`.

[Stolorz et al., 1995] Stolorz, P., Mesrobian, E., Muntz, R., Santos, J., Shek, E., Yi, J., Mechoso, C., and Farrara, J. (Aug. 1995). Fast Spatio-Temporal Data Mining from Large Geophysical Datasets. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 300–305, Montreal, Quebec, Canada.

[Szalay, 1999] Szalay, A. (1999). The Sloan Digital Sky Survey. *Computing in Science and Engineering*, Vol. 1(2).

[Tanaka and Kida, 1993] Tanaka, M. and Kida, S. (Sept. 1993). Characterization of Vortex Tubes and Sheets. *Phys. Fluids*, Vol. A 5(9):pp. 2079–2082.

[Taubin and Rossignac, 1996] Taubin, G. and Rossignac, J. (1996). Geometric Compression through Topological Surgery. *ACM Transactions on Graphics*, Vol. 17(2):pp. 84–115.

[Tohline and Bryan, 1999] Tohline, J. and Bryan, G. (1999). Cosmology and Computation. *Computing in Science and Engineering*, Vol. 1(2).

[Towell and Shavlik, 1994] Towell, G. and Shavlik, J. (1994). Knowledge-Based Artificial Neural Networks. *Artificial Intelligence*, Vol. 70:pp. 119–165.

[Valdés-Pérez, 1994] Valdés-Pérez, R. (1994). Conjecturing Hidden Entities via Simplicity and Conservation Laws: Machine Discovery in Chemistry. *Artificial Intelligence*, Vol. 65(2):pp. 247–280.

[Valdés-Pérez, 1999a] Valdés-Pérez, R. (1999a). Discovery Tools for Scientific Applications. *Communications of the ACM*, Vol. 42(11):pp. 37–41.

[Valdés-Pérez, 1999b] Valdés-Pérez, R. (1999b). Principles of Human Computer Collaboration for Knowledge Discovery in Science. *Artificial Intelligence*, Vol. 107(2):pp. 335–346.

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer Verlag, New York.

[Vashishta et al., 1996a] Vashishta, P., Kalia, R., Nakano, A., and Jin, W. (January 1996a). Silica under Very Large Positive and Negative Pressures: Molecular Dynamics Simulation on Parallel Computers. *International Journal of Thermophysics*, Vol. 17(1):pp. 169–178.

[Vashishta et al., 1996b] Vashishta, P., Nakano, A., Kalia, R., and Ebbsjo, I. (February 1996b). Crack Propagation and Fracture in Ceramic Films Million Atom Molecular Dynamics Simulation on Parallel Computers. *Materials Science and Engineering*, Vol. B37:pp. 56–71.

[Venkatasubramanian et al., 1994] Venkatasubramanian, V., Chan, K., and Caruthers, J. (1994). Computer-Aided Molecular Design Using Genetic Algorithms. *Computers and Chemical Engineering*, Vol. 18(9):pp. 833–844.

[Venkatasubramanian et al., 1995] Venkatasubramanian, V., Chan, K., and Caruthers, J. (1995). Evolutionary Large Scale Molecular Design Using Genetic Algorithms. *J. Chem. Info. and Comp. Sci.*, Vol. 35:pp. 188–195.

[Weir et al., 1995] Weir, N., Fayyad, U., Djorgovski, S., and Roden, J. (December 1995). The SKICAT System for Processing and Analyzing Digital Imaging Sky Surveys. *Publ. Astron. Soc. Pac.*, Vol. 107:pp. 1243–1254.

[Yang et al., 1999] Yang, D.-Y., Grama, A., and Sarin, V. (1999). Error-Bounded Compression of Particle Data for Hierarchical Approximation Techniques. In *Proceedings of the Supercomputing Conference*.

[Yang et al., 2000] Yang, D.-Y., Johar, A., Szpankowski, W., and Grama, A. (March 2000). Summary Structures for Frequency Queries on Large Transaction Sets. In *Data Compression Conference*, pages 238–247, Snowbird, UT.

[Yates and Chapman, 1990] Yates, L. and Chapman, G. (1990). Streamlines, Vorticity Lines, and Vortices. Technical Report AIAA-91-9731, American Inst. of Aeronautics and Astronautics.

[Yip and Zhao, 1996] Yip, K. and Zhao, F. (1996). Spatial Aggregation: Theory and Applications. *Journal of Artificial Intelligence Research*, Vol. 5:pp. 1–26.

[Zhao, 1994] Zhao, F. (1994). Extracting and Representing Qualitative Behaviors of Complex Systems in Phase Spaces. *Artificial Intelligence*, Vol. 69(1–2):pp. 51–92.

[Zhao et al., 1999] Zhao, F., Bailey-Kellogg, C., Huang, X., and Ordonez, I. (1999). Intelligent Simulation for Mining Large Scientific Datasets. *New Generation Computing*. to appear.

[Zhao and Nishida, 1995] Zhao, Q. and Nishida, T. (1995). Using Qualitative Hypotheses to Identify Inaccurate Data. *Journal of Artificial Intelligence Research*, Vol. 3:pp. 119–145.

[Zwillinger, 1992] Zwillinger, D. (1992). *Handbook of Ordinary Differential Equations*. Academic Press.