Data Mining: The Next Generation [1]

1 Introduction

Data Mining has enjoyed great popularity in recent years, with advances in both research and commercialization. The first generation of data mining research and development has yielded several commercially available systems, both stand-alone and integrated with database systems; produced scalable versions of algorithms for many classical data mining problems; and introduced novel pattern discovery problems.

In recent years, research has tended to be fragmented into several distinct pockets without a comprehensive framework. Researchers have continued to work largely within the parameters of their parent disciplines, building upon existing and distinct research methodologies. Even when they address a common problem (for example, how to cluster a dataset) they apply different techniques, different perspectives on what the important issues are, and different evaluation criteria. While different approaches can be complementary, and such a diversity is ultimately a strength of the field, better communication across disciplines is required if Data Mining is to forge a distinct identity with a core set of principles, perspectives, and challenges that differentiate it from each of the parent disciplines. Further, while the amount and complexity of data continues to grow rapidly, and the task of distilling useful insight continues to be central, serious concerns have emerged about social implications of data mining. Addressing these concerns will require advances in our theoretical understanding of the principles that underlie Data Mining algorithms, as well as an integrated approach to security and privacy in all phases of data management and analysis.

Researchers from a variety of backgrounds assembled at Dagstuhl to re-assess the current directions of the field, to identify critical problems that require attention, and to discuss ways to increase the flow of ideas across the different disciplines that Data Mining has brought together. The workshop did not seek to draw up an agenda for the field of data mining. Rather, it offers the participants' perspective on two technical directions—compositionality and privacy—and describes some important application challenges that drove the discussion. Both of these directions illustrate the opportunities for cross-disciplinary research, and there was broad agreement that they represent important and timely

[1] This report is based on a workshop organized by Raghu Ramakrishnan (University of Wisconsin-Madison, USA), Johann-Christoph Freytag (Humboldt-Universität zu Berlin, Germany), and Rakesh Agrawal (IBM Almaden Research Center, USA) at Dagstuhl (see http://www.dagstuhl.de/04292/), and is jointly authored by all participants, including, in alphabetical order:
Toni Bollinger (IBM Development Laboratory Böblingen, Germany);
Christopher W. Clifton (Purdue University, USA);
Saso Dzeroski (Josef Stefan Institute, Slovenia);
Jochen Hipp (DaimlerChrylser AG, Germany);
Daniel Keim (University of Munich, LMU, Germany);
Stefan Kramer (Technische Universität München, Germany);
Hans-Peter Kriegel (University of Munich, LMU, Germany);
Ulf Leser (Humboldt-Universität zu Berlin, Germany);
Bing Liu (University of Illinois at Chicago, USA);
Heikki Mannila (University of Helsinki, Finland);
Rosa Meo (Universita di Torino, Italy);
Shinichi Morishita (University of Tokyo, Japan) ;
Raymond Ng (University of British Columbia, Canada);
Jian Pei (Simon Fraser University, Canada);
Prabhakar Raghavan (Yahoo Inc., USA);
Myra Spiliopoulou (Otto-von-Guericke-Universität, Germany);
Jaideep Srivastava (University of Minnesota, USA);
Vicenc Torra (IIIA-CSIC, Spain).

Dagstuhl Seminar Proceedings 04292
Perspectives Workshop: Data Mining: The Next Generation
http://drops.dagstuhl.de/opus/volltexte/2005/270

areas for further work; of course, the choice of these directions as topics for discussion also reflects the personal interests and biases of the workshop participants.

1.1 Organization of this Report

We give an overview of the topics discussed at the workshop in Section 2, and then discuss applications, compositionality, and privacy in subsequent sections, in that order. Next, we consider application challenges specifically tied to compositionality and privacy, bringing together the earlier discussion threads, and present conclusions.

2 Background

In this section, we provide an overview of the three main discussion tracks.

2.1 Applications

Data Mining is applicable to practically any application where the rate of data collection exceeds the ability of manual analysis, there is an interest in understanding the underlying nature of the application, including unexpected insights, and there is potentially a benefit to be obtained in doing so. We identified a number of applications that can benefit from data mining. These include: Life Sciences (LS), Customer Relationship Management (CRM), Web Applications, Manufacturing, Competitive Intelligence, Retail/Finance/Banking, Computer/Network/Security, Monitoring/Surveillance applications, Teaching Support, Climate modeling, Astronomy, and Behavioral Ecology. Indeed, most scientific disciplines are becoming data-intensive and turning to data mining as a tool.

Are there common themes arising in diverse application domains for which we need to develop unifying foundations and analysis techniques?

- Often, the unique characteristics of data in different application domains (e.g., text, microarray data, genomic sequences, time-series, spatial data) and the distinct nature of the analysis require us to develop new analysis tools. Has the field matured enough that we can identify some recurring themes and techniques that cut across different domains?
- Appropriate application of a given suite of tools to solve a particular problem is often a challenging task in its own right, involving a complex process of obtaining, cleaning and understanding data; selecting and/or generating the appropriate features; choosing suitable analysis techniques; and so on. Are there important commonalities across data mining tasks from different application domains? Is there enough experience to develop a robust methodology that suggests the tools to select and the steps to follow, given a set of data and a set of high-level objectives?

We focus our discussion on two application domains, Life Sciences (LS) and Customer Relationship Management (CRM).

2.2 Compositionality

Database query languages derive their power from compositionality. Can we develop compositional approaches and optimization of multi-step mining "queries" to efficiently explore a large space of candidate models using high-level input from an analyst? This would significantly reduce the time taken in practice to explore a large and complex dataset iteratively.

We discussed examples of real applications that made use more than one data mining operation, and considered how the different steps were, or could have been, composed. Abstracting from the examples, we discussed possible primitive operations that could provide a basis for an algebra of composition, leading to opportunities for optimization in the KDD process as well as the computation itself.

We also felt that compositionality held promise for improving data mining environments, e.g., to store data mining results and their provenance in a secure, searchable, sharable, scalable manner. However, we did not explore the issues in detail. Example questions include:

- Given a set of ongoing mining objectives, how should the data in a warehouse be organized, indexed, and archived?
- Do we need to extend SQL to support mining operations? What is the appropriate granularity? Operations such as clustering or light-weight operations that can be used to implement clustering and other higher-level operations? Examples of the two approaches.
- Do we need to extend SQL to store and reason about mining algorithms and derivations?

2.3 Privacy

As in the discussion of compositionality, we first assembled a list of application scenarios that illustrate and motivate privacy-preserving data mining. We then abstracted from the examples to identify and discuss several foundational issues:

- How can we enable effective mining while controlling access to data according to specific privacy and security policies?
- What are the limits to what we can learn, given a set of governing policies?
- What issues arise in mining across enterprises? Issues in mining in a service-provider environment?
- Recent developments and reports on privacy and data mining.

3 Application Study I: Life Sciences (LS)

We use *Life Sciences* to include molecular biology, bioinformatics, biochemistry, and medical research. Data Mining in LS (**DM/LS**) typically analyses data that is generated by some experimentation, e.g., microarray experiments, or collected as part of clinical routine or in the course of medical studies. A common goal of DM is the detection of some kind of correlation, e.g., between genetic features and phenotypes, or between medical treatments and reactions of patients. Interest in data mining in LS is immense, in research as well as in the pharmaceutical and biotech industry. This is probably due to the potentially high revenues attracting companies, and the complexity of problems attracting researchers.

In our discussion, we distinguish medical data from molecular biology data for several reasons:

- Typically, medical data is directly connected to a person, while molecular biology data is not.
- In molecular biology research many data sets, e.g., the genomes of several species, are publicly available; therefore, data mining applications in LS often use data produced elsewhere. This is rarely the case in medical applications.
- Molecular biology research is often publicly funded and mostly devoted to basic research. In contrast, medical research is often financed by pharmaceutical companies, and often directly addresses questions regarding drug development or drug improvement.

3.1 Example DM/LS Applications

In this section, we briefly describe several specific applications of Data Mining in Life Sciences, in order to provide a concrete setting for the more general discussion that follows.

3.1.1 Finding Transcription Factor Binding Sites

Transcription of one messenger RNA (mRNA) is initiated by several transcription factors that bind the upstream region, the range of approximately fifty nucleotides from the transcription start site of the mRNA in the genomic sequence. The combination of these factors controls the transcription. Each factor binds a particular short string of length several nucleotides, called a binding site, in the upstream region and hence the identification of several transcription binding sites is helpful in finding the corresponding factors, motivating the prediction of transcription factor binding sites in the literature of the bioinformatics research.

To this end, comparison between the upstream regions of mRNAs appears to be meaningful; however, we have to be careful in selecting mRNAs to compare, because some mRNAs are transcribed only at a special point of time, or in a specific part of cell. Thus, grouping mRNAs according to the patterns of their expressions monitored by microarrays, for instance, is expected to provide a cluster of mRNAs that share a combination of transcription factor binding sites in common. This analysis calls for clustering algorithms.

Subsequently, upstream regions of mRNAs in the same group are examined to discover binding sites, short strings of several nucleotides. For this purpose, multiple alignment algorithms are utilized to uncover highly conserved common strings. This approach was successful in predicting known binding sites of budding yeast with a high probability [BJV+98]. Although it seems promising for investigating genomes of unicellular species, care has to be taken in extending this technique to vertebrate genomes. One major concern is that serious noise in gene expression data observed by microarrays may cause unreliable clusters. Another is that the transcription start sites are frequently multiple and are not fully understood, making it difficult to acquire upstream regions. These issues are outside the scope of data mining tasks per se, but we have to bear them in mind when applying data mining tools.

3.1.2 Functional Annotation of Genes

One major concern of the current so-called post-genome area is the determination of gene function, also named functional genomics. There are several ways to do this, including knock-out or RNAi to suppress or down-regulate the expression of genes, or deduction from homologues genes (i.e., genes with the same evolutionary ancestor) with known function. However, these and many other techniques heavily depend on the knowledge about homologous genes in species that are can be subjected to experimentation, in contrast to humans. If no close homologues are known, functional annotation can be approached by a data mining approach.

As an example, we sketch the idea presented in [GSS03]. Consider a microarray experiment measuring expression intensities of genes in a cell, under given circumstances. We could conduct such an experiment at defined time points while changing environmental conditions, e.g., adding poisonous or removing nutritious substances. The cell will react to these changes by activating signaling and metabolic pathways, e.g., to move to a less energy-consuming state. The genes in such a pathway will react accordingly to the environmental change. Therefore, if we have performed such an experiment and have found a group of co-regulated genes of which some have functional annotation and some have not, we can try to deduce the annotation for the latter from the annotation of the former.

Data mining plays a major role in this process. First, raw microarray data needs to be transformed and normalized to allow for comparison. Often, changes in expression levels are broadly discretized to either “up”, “down”, or “constant”. Next, clusters of co-regulated genes need to be determined. Hierarchical clustering or k-means clustering are popular approaches, but many other methods have been used as well. Once these clusters have been determined, the functional annotation of the genes in a cluster needs to be retrieved. Typically, this information is encoded in ontologies such as the Gene Ontology [GO01]. These ontologies are essentially DAGs (**d**irected **a**cyclic **g**raphs); in the case of the Gene Ontology, there is one DAG for the description of biological processes, one for molecular functions, and one for cellular locations. In the ideal case, most genes of a cluster have an annotation and all of them have the same annotation, e.g., “cell wall construction” or “DNA repair”. In this case, we may assume that the genes in the cluster not having some functional information assigned also play a role in the annotated processes. This deduction becomes more difficult if the annotations in the cluster are mixed, in which case one may try to use the hierarchical relationships between annotation terms to derive a less specific annotation.

Characteristics of this example worth noting are the need to combine data from multiple sources (microarrays, annotation, ontologies), the difficulty in defining confidence scores based on heterogeneous data sets and analysis methods, and the use of hierarchically structured information. Error-analysis, for instance using permutation testing and

cross-validation, is of very high importance, since microarray data sets are extremely noisy. Furthermore, clustering microarray data has a recurring dimensionality problem, since there are many genes (>10.000), but usually only a few observations per genes (<100).

3.1.3 Detection of Evolutionary Differences between Various Species

After the elucidation of human genome in 2001, a number of other genome sequencing projects have finished, and many are still ongoing. These efforts yield complete or partial genomic sequences of many species, e.g. chimpanzee, mouse, rat, chicken, dog, and several fishes (Fugu, Zebra, and Medaka). In sequencing projects, generating the final sequence from the experimental data demands an extremely intensive computation, and it is still hard to generate highly accurate sequences with minimal errors. Most of the assembled sequences are typically fragmented and contain many gaps between contiguous strings. In spite of this incompleteness, comparison across a variety of genomes, which is investigated in the growing area of comparative genomics, provides enormous information.

For instance, regions among the regions of mammalian species such as human, mouse, and rat frequently coincide on coding regions, allowing us to predict novel genes through the alignment of genomic sequences of multiple species. Comparison between mammalian genomes and fish genomes is also useful, since highly conserved regions must have an important function that prevented evolution from changing them over a period of hundreds of millions of years. By contrast, human and chimpanzee genomes are too similar to be effective for discovery of genes and their regulatory sites. However, human–chimpanzee comparative genome research is expected to uncover unique human features such as highly developed cognitive functions and the use of language. For instance, olfactory receptor genes have been steadily lost in all great apes, but the rate of gene loss in the human linkage is extremely high. It is speculated that the acquisition of bipedalism and enhancements in human cognitive capacity might have reduced the necessity for an acute sense of smell.

Comparative genomics calls for sensitive and efficient alignment methods that facilitate the detection of low homologous regions between large-scale genomic sequences. Such alignment methods are categorized as “tools for data mining” in the Life Science community (see e.g. the US National Center for Biotechnology Information at National Institute of Health). The simplest form of pattern search is to look for single or multiple hits of a short consecutive (continuous) sequence, called a seed, which appears in both the query and the database. Many widely used tools implement this idea. Tools searching only for highly conserved regions can safely require that seeds have a certain length and exhibit a high similarity in the query and a potentially matching database sequence, thus allowing for faster algorithms without sacrificing sensitivity [Ken02]. In comparative genomics, however, the expected homology ratio is much lower, around 70-80%, demanding that more sensitive algorithms are applied. Analysis of searching for spaced seeds with “don’t care” letters, in particular, has attracted much attention in the bioinformatics community [BKS03]. Although computing the most sensitive spaced seed is intractable in general, several effective spaced seeds have been proposed.

3.1.4 Revealing Protein-Protein Interactions Using Text Mining

Despite the high importance of databases for Life Science research, most knowledge is still encoded in scientific publications, i.e., unstructured natural-language text. Text mining tries to find and extract this knowledge for specific questions. This area has achieved considerable attention in the Life Science domain in the last 5 years, mostly due to the increasing importance of high-throughput experimentations that generate little information about very many objects.

As a specific example we consider the extraction of protein-protein interaction from publications. In the simplest case, such interactions can be extracted in the following way [JLK+01]: The articles under consideration, typically a set of Medline abstracts, are broken up into sentences. Then, each sentence is searched with a predefined dictionary of protein names. Whenever two proteins occur in the same sentence, it is assumed that there was an interaction between those two. A confidence measure can be derived if multiple co-occurrences of a protein pair in different articles are considered.

There are many systems that use more sophisticated methods. For instance, dictionary-based approaches are known to perform badly on gene and protein names since for most higher developed organisms no conventions have been defined yet or are not used consequently enough by the research community. Advanced methods instead tackle the name recognition problem by turning it into a classification problem of words and phrases. For each word, features such as

contained letters and n-grams, case, embedded special characters or digits, etc., are generated and fed into a machine learning algorithm, such as support vector machines [HBP+05]. Another pitfall of pure co-occurrence is that it produces many false positives. To be more specific, sentences can be matched against rules describing typical patterns for describing protein interaction in natural language, usually expressed as regular expressions on the words or on linguistic annotation of working well enough to be used as input for further analysis. Instead, they are used to produce suggestions for human annotators of protein interaction databases [DMB+03].

The problem requires the combination of several data mining techniques (classification, feature selection, dimensionality reduction, and pattern discovery). Evaluation of the performance of the results is of highest importance, and considerable resources are currently being invested into the generation of manually tagged corpora as input to the machine learning algorithms and for evaluation (see, e.g., [KOT+03]).

3.2 Analysis of Data Mining in Life Sciences

We present our analysis of DM/LS in terms of the several questions. We think that these questions provide a useful framework for discussing whether data mining is a suitable technology for any given application area, though we have only applied it to Life Sciences:

.

1. **Problem Context:** What's the overall objective? What are processes where DM is used? What are the technical and human processes where DM is used?
2. **Data Characteristics:** How data intensive is the area? Volume, structure, quality?
3. **Data Acquisition:** Is data collection actively driven by monitoring the application? What is the cost of data acquisition? How do we integrate Data Mining with data collection? Can we boot-strap by applying mining techniques to the acquisition process itself?
4. **Social and Human Aspects:** Are practitioners in the application domain open to DM? What is the appropriate model for applying DM—product or service? What is the effort involved in "knowledge elicitation" in this domain? How much knowledge needs to be elicited before DM can be applied successfully? What is the elicitation process?
5. **Cost-Benefit Analysis:** Is there a methodology to validate the results of applying DM technologies? Can ROI/value/cost arguments be made? What are the big "pain points" in this application domain? Are they likely to be addressed by DM?
6. **Gap Analysis:** What is the maturity level of DM technology relevant to this domain? How can domain knowledge be incorporated?

Problem Context: The recent flood of molecular biology data demands computational tools for discovery; Biology is becoming an information science. Pattern matching and (multiple) sequence alignment are major data processing tools in Life Science. As stated before, the NCBI categorize these tools into "data mining". Note that sequence similarity is essentially a measure for defining proximity between objects and often serves for nearest neighbor type problems. If we focus on microarray data analysis, classification is used for separating ill from healthy, clustering for identifying co-regulation of genes, association rule mining for regulatory element discovery, and text mining for associating genes with gene ontology terms. Furthermore, inductive logic programming has been used for finding matching substructures between molecules, and decision trees for functional annotation in Swiss-Prot, a major protein database.

- Research on complex diseases is one of the major next objectives in Life Science, after monogenetic diseases have been considerably studied. Finding biomarkers or reporter genes for complex diseases is really a challenging problem to which data mining could be applicable.
- Large-scale collection of variations in genomic sequences and haplotypes has been recently promoted worldwide, with large-scale data gathering projects in the UK, Iceland, and Estonia. Inclusion of these variations into data mining algorithms is crucial.
- Large-scale comparative genomics is an emerging research area that is urged by elucidation of a number of vertebrate genomic sequences. Tools for seeking homologous sequences are primary means, but advanced sequence analysis, such as discovery of regulatory elements, calls for clustering and classification. Also, phylogenetic algorithms, deducing the evolutionary tree or network structure of species and individuals from sequence data, are a very important line of research.

- Drug discovery is another promising area. Although a well established research area, it still offers great opportunities and hosts major challenges. Since the three dimensional structural data of about 28,000 proteins has been resolved experimentally and physical interactions among proteins are measured in large-scale, predicting the 3D structure of a protein from its one dimensional sequence information and estimating the possibility of interactions among proteins and compounds are challenging.
- In general, more biological background should be incorporated into data mining algorithms to increase prediction accuracy. For instance, alignment algorithms can be improved by using score matrices based on statistics of short sequence usage in a particular group of species.

Data Characteristics: Data in Life Science is extremely diverse. We have to be careful in the application of DM to these datasets, since many of them suffer from poor data quality, leading to a high level of noise, and incompleteness, resulting in missing or unknown values due to problems in the experiment.

- The total amount of sequence data, including textual annotation, is still less than one terabyte. Because of its simple structure of a sequence, the format of sequence data is standardized.
- Microarray data and protein array data are essentially tabular, but of enormous size, as 30,000 genes can today easily be monitored on a small chip. Since no central availability of large dataset in this area in a consistent format has been achieved yet, GeneExpressionOmnibus and ArrayExpress are targeting this goal. Currently, different datasets are available only from different sites in distinct data formats, making comparative analysis difficult.
- One typical proteomics data example is yeast two-hybrid datasets that describe physical interactions between budding yeast genes in an undirected graph. Another is mass-spectrometry that initially consists of spectra and is finally processed into peptides and proteins. Raw mass-spectrometry data sets are huge, while the number of identified proteins is relatively small. Analysis of raw mass-spectrometry data to determine protein and peptide sequences is a difficult computational task and still needs improvement.
- Three dimensional structural data of proteins constitutes another high-volume dataset in Life Science. To date, the structures of about 28,000 proteins are registered in public databases, though some of them are redundant.
- Medline currently stores about 15,000,000 publications, growing at a rate app. 500,00 abstracts per year, and is a major source of knowledge in Life Science. Though abstracts are to some extent annotated with standardized keywords (so called MESH terms, Medical subject headings), the heterogeneity and variability of natural language makes text mining still a very challenging task. Furthermore, the amount of textual data will grow rapidly as more and more full text articles, especially those of the fast growing number of open access journals, are becoming available.
- Annotating biological phenotypes and behavior is indispensable to process experimental data such as gene expressions or protein structures. Many efforts have been made to facilitate annotation tasks by several organizations such as Gene Ontology Consortium, MGED, the Protein Structure Initiative, and SBML.org.
- Forcing or knocking out the expression of a particular gene allows us to produce mutants systematically. This research area is called phonemics, and a number of groups have recently reported genome-wide screening data for mutants with abnormal morphology. In the past, cell morphology researchers have processed information on cells manually. These time-consuming, entirely subjective tasks demand image-processing software. Several large-scale public image databases are emerging, but no serious standardizing efforts are being made yet.
- DNA sequences of two unrelated people are equal at about 99.9%, but the remaining 0.1% contains the genetic variants that influence how people differ in their risk of disease or their response to drugs. Sites in the genome where the DNA sequences of many individuals differ by a single base are called single nucleotide polymorphisms (SNPs). About 10 million SNPs exist in human populations, but finding all SNPs is costly to achieve. Since some SNPs are strongly correlated in genotypes, the HapMap project aims at identification of less than one million independent SNPs.
- Uncovering the DNA sequence variants that are concerned with common disease risk is expected to be useful in understanding the complex causes of disease in humans. Discovering SNPs that are highly correlated with diseases or responses to drugs demands numerous samples. Towards this end, for instance, the Japanese government is conducting a project for collecting SNPs and clinical data from 300,000 people.

- Very often, Life Science data is of high dimensionality, where the number of dimensions by far exceeds the number of observations. This is typical for microarray and proteomics data (many more genes or proteins than experiments), and is equally evident in the domain of structural bioinformatics (many more substructures than experimentally observed structured) in text mining (many more words and sentences than papers).

Data Acquisition: In general, data acquisition costs in Life Science are high, and therefore data mining and analysis are actively intertwined with data collection to minimize collection of redundant data. Keep in mind, though, that the cost has been changing quickly; for instance, sequencing used to be very expensive (10-14 USD per base), but now it is cheap (less than 10 cent per base). Medical datasets and molecular biology datasets are quite different with respect to privacy issues:

- Because of privacy issues in medical data, data is allowed to be collected for a study after permission by an ethics commission for specific purposes.
- There are more than 500 public bioinformatics databases that are distributed among the WWW. Their integration is mostly hindered by missing standards and comparability.

Social and Human Aspects: There are several points common to DM across Life Sciences:

- It is questionable if science can be modeled in terms of business processes. For instance, collection of patients' clinical data may inherently take many years. Studies in Life Science may not be an engineering task, but closer to a kind of art. There are too many unknown factors in environment and measurement process, because living "things" are the object of study. Whether or not the inherently discovery-oriented approaches in the Life Sciences can be modeled as a knowledge discovery process is an open question; notwithstanding, there is a creative process underlying discovery that must be supported by tools and software.
- Data mining researchers actually provide mining services to biologists. Few classical consultants are involved.
- Products such as microarray databases and analysis software are often sold together with chip hardware. Software tools for designing primers and siRNA sequences are also provided with users as a part of service by biotechnology companies. For protein identification, commercial products, for instance Mascott and Seaquest, are frequently used. There are also companies for text mining.

On the other hand, the cultures of medical and biological informatics differ in some important ways:

- Bioinformaticians are open to new analysis techniques and are developing novel and useful tools, thereby driving the Life Science field. By contrast, medical doctors are usually hesitant, partly because they are not familiar with statistics. Thus, the direct communication between data mining researchers and medical doctors may not be successful. There must be an interface, such as biostatistians or bioinformaticians. Biologists are diverse. Biologists who process large-scale biological data need bioinformatics software, and hence they are mostly open.

Cost-Benefit Analysis: Validating data mining predictions is absolutely crucial, because many failed studies and wrong results are likely to be produced due to poor data quality and high error rate, for instance in gene expression and proteomics experiments. ROC and cross-validation are standard ways of computational validation before moving to biological validation. Biological and medical validations need more studies and more patients, and hence they are very expensive and time consuming tasks, but promising revenues may be rewarded.

Gap Analysis: Many traditional algorithms can be applied in Life Sciences, although there are several problems where new algorithms are required.

- The real issues would be how to deal with errors and poor data quality. Also, it is hard to build models when we face serious knowledge gaps, as in pathways.

In general, it is expected that progress in the study of complex diseases such as tumors, diabetes, or Alzheimer disease, will only be reached if the evolving experimental methods are accompanied by advances in the area of data mining and data analysis, leading to algorithms that jointly analyze data from multiple sources and of multiple types in a robust manner. One major challenge is the development of confidence scores for combinations of different techniques. The currently applied multi-step processes, where each step applies a specific algorithm with a specific error or performance measure, are difficult to combine such that an overall estimation of the cost, quality, and reliability of the results is derived. There is a great danger in unsystematically applying different methods and stopping whenever the first results occurs that seems interestingly enough, without having a measure for the significance of the finding.

A second challenge concerns the automatic, or at least semi-automatic, exploration of the search space of different methods for different steps in a multi-step analysis process. For instance, in the first example given above, a clustering step is followed by a pattern discovery and then by a pattern matching step. For each of these two steps many different algorithms may be used, and the second and third step are often merged into one multiple alignment step, where again a manifold of methods may be used. Often, it is unclear which combination of algorithms is best for a given problem. Tools to explore the search space systematically should help in eventually finding the best combination.

A third challenge is the development of methods that mine numerical data together with textual data. Textual data plays a major role in the Life Science, e.g., in the form of database annotations, scientific publications, or medical record. This text might be unstructured natural language, semi-structured descriptions, or structured codified diagnoses. Including the knowledge expressed in this text into the analysis of experimental results and vice-versa should yield better and more trusted data mining results.

There are great opportunities, making this area worth the effort. Bioinformatics is moving more and more into medical applications, and the development of new drugs or new diagnostic tools carry the premise of a billion-dollar market. For instance, the analysis of 3D patterns together with information about protein interactions and pathways has many applications in drug development. Algorithms for correlating genotypes with phenotypes, especially using SNPs related to drug metabolism, are expected to make dosages of drugs much more patient-specific, thus reducing adverse effects. The development of robust classifiers for gene expression data has already resulted in commercial products offering faster and more precise tools, for instance for tumor diagnosis. Another not yet fully exploited opportunity lies in the integrated usage of genetic data with medical data, for instance to find correlation between genotypes and effect of treatment or between medical markers and expected disease outcome. Medical research has a long history in studying the effect of new drugs on the healing of certain diseases, but the inclusion of genetic data into this process may lead to much more detailed results.

4 Application Study II: Customer Relationship Management (CRM)

Customer Relationship Management (CRM) emerged in the last decade to reflect the central role of the customer for the strategic positioning of a company. CRM takes a holistic view of customers. It encompasses all measures for understanding the customers and for exploiting this knowledge to design and implement marketing activities, align production and coordinate the supply-chain. CRM puts emphasis on the coordination of such measures, also implying the integration of customer-related data, meta-data and knowledge and the centralized planning and evaluation of measures to increase customer lifetime value. CRM gains in importance for companies that serve multiple groups of customers and exploit different interaction channels for them. This is due to the fact that information about the customers, which can be acquired for each group and across any channel, should be integrated with existing knowledge and exploited in a coordinated fashion.

It should be noted, however, that CRM is a broadly used term, and covers a wide variety of functions, not all of which require data mining.[2] These functions include *marketing automation* (e.g., campaign management, cross- and up-sell, customer segmentation, customer retention), *sales force automation* (e.g., contact management, lead generation, sales

[2] Customer Relationship Management (CRM) as a product/market segment was first introduced by companies such as Siebel and Oracle, with many other players, including SAP, PeopleSoft and Microsoft joining subsequently. The initial set of products mostly support easy management of information for customer facing functions, including *contact management*, *sales force automation*, etc. Applying data mining to better understand customers, and its use for relationship management, constitutes a more recent phenomenon.

analytics, generation of quotes, product configuration), and *contact center management* (e.g., call management, integration of multiple contact channels, problem escalation and resolution, metrics and monitoring, logging interactions and auditing), among others. We focus on how backend data mining and analytics can make these functions more effective.

A Note on Personalization: While personalization and CRM are often spoken of in the same breath, it is important to note that the two are distinct, and each can exist without the other. Specifically, one could have customer relationship management without personalization, i.e., the manner in which relationship management is done is not sensitive to the needs of individual (or groups of) customers. Furthermore, non-personalized CRM does not necessarily indicate bad or unsatisfactory service; it just means that 'everyone is treated the same'. For example, Domino's Pizza has (or at least used to have) a `30 minute or free' guarantee on Pizza delivery, regardless of who was placing the order. The service itself was rated very high on customer satisfaction, but it was nonetheless non-personalized.

On the other hand, personalization can be for purposes other than customer relationship management. For example, analysis of an auto insurance applicant's driving and credit history to increase insurance rates or even cancel the insurance is an example of personalization; albeit in a rather negative sense. Current application of traveler profiling and analysis by the Transportation Security Administration (TSA) in the United States is another such example.

4.1 Analysis of Data Mining in CRM

Problem Context: The maximization of lifetime values of the (entire) customer base in the context of a company's strategy is a key objective of CRM. Various processes and personnel in an organization must adopt CRM practices that are aligned with corporate goals. For each institution, corporate strategies such as diversification, coverage of market niches or minimization of operative costs are implemented by "measures", such as mass customization, segment-specific product configurations etc. The role of CRM is in supporting customer-related strategic measures.

Customer understanding is the core of CRM. It is the basis for maximizing customer lifetime value, which in turn encompasses customer segmentation and actions to maximize customer conversion, retention, loyalty and profitability. Proper customer understanding and actionability lead to increased customer lifetime value. Incorrect customer understanding can lead to hazardous actions. Similarly, unfocused actions, such as unbounded attempts to access or retain *all* customers, can lead to decrease of customer lifetime value (law of diminishing return). Hence, emphasis should be put on correct customer understanding and concerted actions derived from it.

Figure 1 shows an idealized CRM cycle.

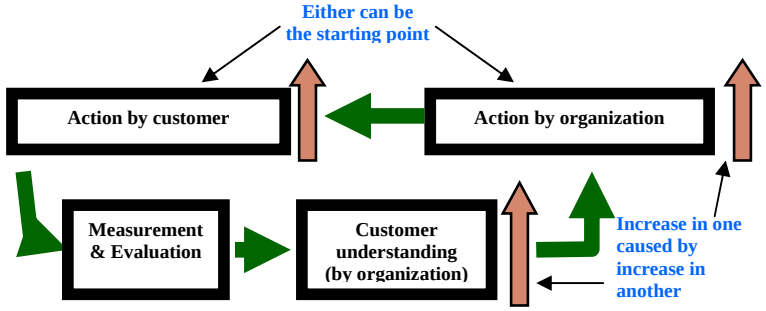

Figure 1. The Basic CRM Cycle.

In this figure, boxes represent actions:

- The customer takes the initiative of contacting the company, e.g. to purchase something, to ask for after sales support, to make a reclamation or a suggestion etc.
- The company takes the initiative of contacting the customer, e.g. by launching a marketing campaign, selling in an electronic store or a brick-and-mortar store etc.

- The company takes the initiative of understanding the customers by analyzing the information available from the other two types of action. The results of this understanding guide the future behavior of the company towards the customer, both when it contacts the customer and when the customer contacts it.

The reality of CRM, especially in large companies, looks quite different from the central coordination and integration suggested by Figure 1:

- Information about customers flows into the company from many channels, but not all of them are intended for the acquisition of customer-related knowledge.
- Information about customers is actively gathered to support well-planed customer-related actions, such as marketing campaigns and the launching of new products. The knowledge acquired as the result of these actions is not always juxtaposed to the original assumptions, often because the action-taking organizational unit is different from the information-gathering unit. In many cases, neither the original information, nor the derived knowledge is made available outside the borders of the organizational unit(s) involved. Sometimes, not even their existence is known.
- The limited availability of customer-related information and knowledge has several causes. Political reasons, e.g. rivalry among organization units, are known to lead often in data and knowledge hoarding. A frequently expressed concern of data owners is that data, especially in aggregated form, cannot be interpreted properly without an advanced understanding of the collection and aggregation process. Finally, confidentiality constraints, privacy considerations and law restrictions often disallow the transfer of data and derived patterns among departments.
- In general, one must assume that data gathered by an organization unit for a given purpose cannot be exported unconditionally to other units or used for other purpose and that in many cases such an export or usage is not permitted at all.
- Hence, it is not feasible to strive for a solution that integrates all customer-related data into a corporate warehouse. The focus should rather be in mining non-integrated, distributed data while preserving privacy and confidentiality constraints.[3]

A more realistic picture of current CRM, incorporating the typical flow of information, is shown in Figure 2:

- Data is collected from multiple organizational units, for different purposes, and stored in multiple locations, leading to redundancies, inconsistencies and conflicting beliefs.
    - No organization unit has access to all data and to all derived knowledge.
    - Some data are not analyzed at all.
    - Not all background knowledge is exploited.
    - Data analysis is performed by many units independently.
    - Some analyses do not mount to actions.

Ideally, the CRM cycle should encompass:

- the exploitation of all data and background knowledge
- the coordination of analyses, and resulting actions, in different parts of the organization

[3] As these observations indicate, CRM solutions address a mix of technical and organizational issues. Some of the problems listed above can be solved by making top level management directly involved in the CRM initiatives. We focus our discussion on the technical aspects, while noting that significant organizational challenges must also be addressed in any real CRM undertaking.

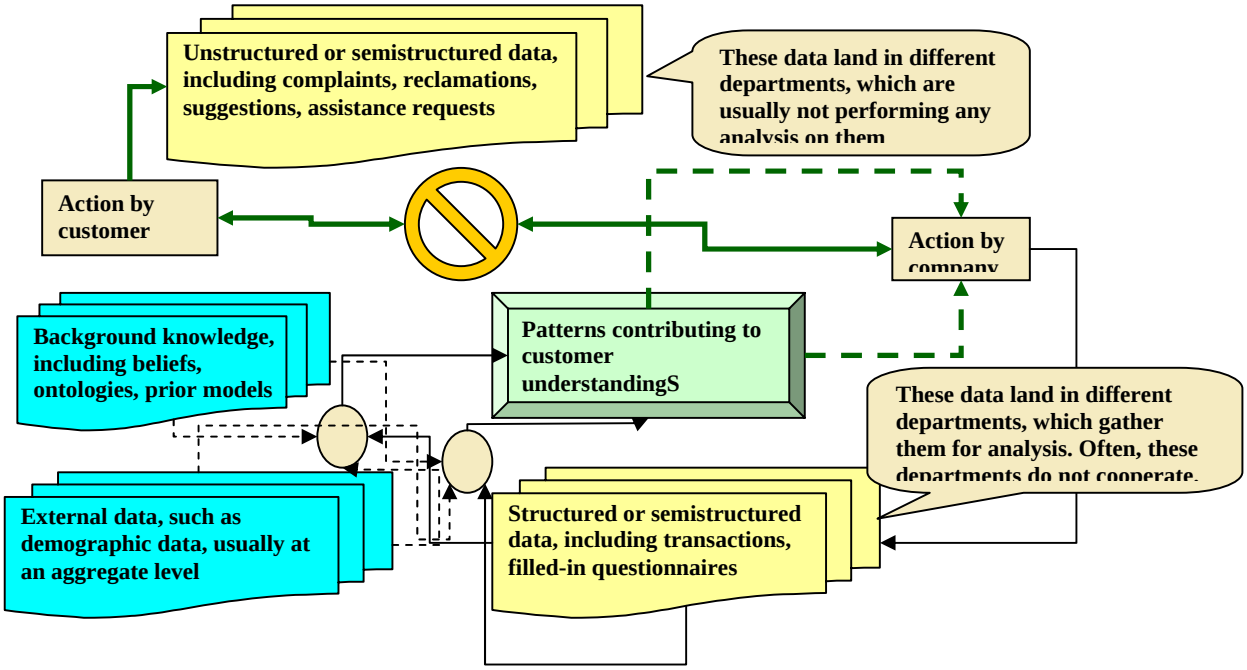

Figure 2. Expanded CRM Cycle – Current State.

Gap Analysis: The grand KDD challenges in CRM arise from the objectives of exploiting all information and coordinating analysis and actions. These objectives require methods to deal with several specific challenges, which we discuss in turn:

- *Cold start:* In CRM, one tries to influence customer behavior on the basis of prior knowledge. Often, there is no (reliable) such prior knowledge.
- *Correct vs. incorrect customer understanding:* CRM is about understanding the customer. It's about time to elaborate on the impact of *misunderstanding* the customer, and to fold this into our analyses and action workflows.
- *Data sovereignty:* There is no such thing as a CRM data warehouse; we are faced with multiple data sources. If and when problems of semantic disparity are solved, we will still face legislative and the political hurdles.[4] We need solutions where the data owner is allowed to specify what data she wants to deliver, at what level of abstraction, and with what meta-information.
- *Data quality:* Some Customer-Company-Interaction channels deliver very good data. Others deliver notoriously poor quality data; web server logs belong to the latter category. The issue of data sovereignty impedes integration of raw data. Despite these odds, data must be enhanced to ensure that the results of the analysis are reliable.
- *Deeper understanding:* Profiling is based on some rudimentary behavioral data and some preferences, carefully extracted from questionnaires or learned from the data using data mining methods. Integration of cultural and psychological data is at its infancy. The experts in those domains come from outside of the KDD community (marketing researchers, practitioners, etc.) and we should establish collaborative relationships with them.
- *Questioning prior knowledge:* Everything associated with prior knowledge is assumed correct. We need mechanisms that capture and align prior knowledge in the face of conflicting information.
- *Actionability:* Pattern discovery should lead to actions. In some cases, this is straightforward, e.g., site customization and personalization, but this step is often omitted. We need mechanisms that incorporate patterns

[4] These are organizational and legal challenges that can be resolved in one way or another. However, this is a separate (non-technical) dimension of CRM.

into the action-taking processes in a seamless way. We also need an understanding of the action-taking processes and their influence on what patterns are most valuable.

5 Data Mining Challenges & Opportunities in CRM and Life Sciences

In this section, we build upon our discussion of CRM and Life Sciences to identify key data mining challenges and opportunities in these application domains.[5]

5.1 Challenges Common to Both Domains

The following challenges are common to both LS and CRM:

- Non-trivial results almost always need a combination of DM techniques. Chaining/composition of DM, and more generally data analysis, operations is important.
    - In order to analyze LS and CRM data, one needs to explore the data from different angles and look at its different aspects. This should require application of different *types* of DM techniques and their application to different "*slices*" of data in an interactive and iterative fashion. Hence, the need to use various DM operators and combine (chain) them into a single "exploration plan".
- There is a strong requirement for data integration before data mining.
    - In both cases, data comes from multiple sources. For example in CRM, data needed may come from different departments of an organization. In LS, biological data may come to different laboratories. Since many interesting patterns span multiple data sources, there is a need to integrate these data before an actual data mining exploration can start.
- Diverse data types are often encountered, which requires the integrated mining of diverse and heterogeneous data.
    - In LS, this is very clear. In CRM, while dealing with this issue is not critical, it is nonetheless important. Customer data comes in the form of structured records of different data types (e.g., demographic data), temporal data (e.g., weblogs), text (e.g., emails, consumer reviews, blogs and chat-room data), (sometimes) audio (e.g., recorded phone conversations of service reps with customers).
- Highly and unavoidably noisy data must be dealt with.
    - Again, this is very clear in LS. In CRM, weblog data has a lot of "noise" (due to crawlers, missed hits because of the caching problem, etc.). Other data pertaining to customer "touch points" has the usual cleaning problems seen in any business-related data.
- Privacy and confidentiality considerations for data and analysis results are a major issue.
    - This is obvious for many types of LS data (e.g., medical records). In CRM, lots of demographic data is highly confidential, as are email and phone logs. Concern about inferencing capabilities makes other forms of data sensitive as well—e.g., someone can recover personally-identifiable information (PII) from web logs.
- Legal considerations influence what data is available for mining and what actions are permissible.
    - In some countries it is not allowed to combine data from different sources or to use it for purposes different from those for which they have been collected. For instance, it may be allowed to use an external rating about credit worthiness of a customer for credit risk evaluation but not for other purposes.
    - Ownership of data can be unclear, depending on the details of how and why it was collected, and whether the collecting organization changes hands.
- Real-world validation of results is essential for acceptance.
    - Just doing "in-silico" DM is not enough in LS (no matter how statistically strong your patterns/discoveries are); the real test is in-vivo and in-vitro biological experiments. In CRM, as in many DM applications, discovered patterns are often treated as hypotheses that need to be tested on new data using rigorous statistical tests for the actual acceptance of the results. This is even more so

[5] Note that there is some overlap with the "gap analysis" in the preceding sections.

for taking or recommending actions, especially in such high-risk applications as in the financial and medical domains. Example: recommending investments to customers (it is actually illegal in the US to let software give investment advice).

5.2 Issues Specific to Life Sciences

Biology is becoming an information science, according to many, and it is anticipated that the role of computer science will grow. As experiments become more and more automatic and easily done, and more data is produced, it is perhaps safe to say that the biology has come to the point that without effective data analysis, it will be hard to progress. To conduct data mining research in Life Sciences, we suggest the following guidelines:

- Don't do research in LS without biologists. LS applications often need significant domain knowledge. What to do in data mining and what mining results make sense always require the judgment of domain experts. Many results also need biological experiments to confirm or to reject. Thus, doing any data mining application without a biologist is highly undesirable, and can result in significant waste of time and effort.
- Be more systematic than biologists want you to be. Do not approach your problem in an ad hoc manner, but systematically. Biologists only want to see the results quickly. However, as a data miner, care must be taken to ensure that you
    - Spend time on defining the search space (your method)- which algorithms, which parameter ranges, which significance scores;
    - Spend time on working towards semi-automatic exploration of search space (i.e., estimate robustness and significance);
    - Don't take the first results that biologists like ("Of course, that's looks interesting …"), but insisting on further exploration, which often results in truly interesting outcome.
- Don't blindly trust prior knowledge. Much of the prior "knowledge" in the biology domain is in the form of hypotheses based on circumstantial evidence. Thus, in the data mining process, you should be prepared to challenge such knowledge. It is often the case that biologists may also be unsure what is right and what is wrong. Thus, one needs to be prepared to explore together with the biologists. In addition, much existing knowledge changes very rapidly (e.g., GeneOntology, SwissProt, Ensemble).
- Do worry about relevance first, then about performance and scalability. For many data miners, performance and scalability are considered the dominant issues, especially in research. However, to do a successful application in the biology domain, relevance and correctness of results are the main concern. Once these have been achieved, we can then worry about the performance issue of algorithms.

5.2.1 Main Technical Challenges

- Combination of multiple types and sources of data, and use of multiple mining techniques. Information about LS comes in all forms, e.g., texts from research publications, sequences, and 3-D structures. Mining such diverse but highly relevant data is a very interesting research direction. In most applications, one needs a combination of techniques including non-data mining techniques. Allowing biologists to explore the solution space through semi-automatic search space exploration can be very valuable.
- Description of variations (e.g., in sequences, haplotypes).
    - Describe and include these variations into mining algorithms.

  The same biological data may have different variations. How to consider these different variations in mining is also an interesting problem because a typical mining algorithm is only able to take a specific type of data.
- Study of complex diseases.
    - Monogenetic diseases are well studied, try to take on the more difficult (and more frequent!) ones;
    - Find biomarkers / reporter genes.

  Many monogenetic diseases are well studied. Data mining should take on the more difficult ones, which not only will challenge existing mining algorithms but also make bigger practical impact to Life Sciences.
- Handling sparse, high-dimensional data. Current theoretical and practical results of data mining or machine learning are based on having a large number of training instances. Because many biological datasets have very

few tuples but a large number of attributes, data mining algorithms tend to overfit the data. Novel solutions are called for. Active and integrated of learning and experimentation could be the solution.

- Including more biological background in algorithms.
    - Example: Alignment, A+ScoringMatrices, A+M+HomologuesSequence, A+M+H+?
    - Example: Expression, E+FunctionalAnnotation, E+F+?

  For some biological problems, existing knowledge is well established. Using such knowledge in the mining algorithm will enable the biologists to find unexpected and novel knowledge from the data. Even if the knowledge is only hypothesis, it still allows the biologist to verify his/her hypothesis or to identify flaws in the hypothesis. In short, both unguided and guided data mining will be very helpful.
- Issues about hierarchies.
  In LS domains, hierarchies or taxonomies are widespread because biological phenomenon often occurs at different levels of detail. Not surprisingly, much of the existing knowledge in LS is represented hierarchically. Biologists are familiar with tree structure representations. The following questions and issues are of particular interest related to hierarchies, and their role in abstracting background knowledge:
    - Hierarchy of species defines proximity between sequences necessary for using information from other species;
    - Hierarchies of functional description (ontologies) of genes give information for measuring quality and consistency of clusters;
    - Multiple sequence alignment methods often use estimated phylogenetic tree of sequences.

5.3 Issues Specific to CRM

- *Developing deeper models of customer behavior:* One of the key issues in CRM is how to understand customers. Current models of customers mainly built based on their purchase patterns and click patterns at web sites. Such models are very shallow and do not have a deep understanding of customers and their individual circumstances. Thus, many predictions and actions about customers are wrong. It is suggested that information from all customer touch-points be considered in building customer models. Marketing and psychology researchers should also be involved in this effort. Two specific issues need to be considered here. First, what level should the customer model be built at, namely at the aggregate level, the segment level, or at the individual level? The deciding factor is how personalized the CRM effort needs to be. Second is the issue of the dimensions to be considered in the customer profile. These include demographic, psychographic, macro-behavior (buying, etc.), and micro-behavior (detailed actions in a store, e.g. individual clicks in an online store) features.

- *Acquiring data for deeper understanding in a non-intrusive, low-cost, high accuracy manner:* In many industrial settings, collecting data for CRM is still a problem. Some methods are intrusive and costly. Datasets collected are very noisy and in different formats and reside in different departments of an organization. Solving these pre-requisite problems is essential for data mining applications.

- *Managing the "cold start/bootstrap" problem:* At the beginning of the customer life cycle little is known, but the list of customers and the amount of information known for each customer increases over time. In most cases, a minimum amount of information is required for achieving acceptable results (for instance, product recommendations computed through collaborative filtering require a purchasing history of the customer). Being able to deal with cases where less than this required minimum is known is a therefore a major challenge
- *Evaluation framework for distinguishing between correct/incorrect customer understanding:* Apart from the difficulty of building customer models, evaluating them is also a major task. There is still no satisfactory metric that can tell whether one model is better than another and whether a model really reflects customer behaviors. Although there are *some* metrics for measuring quality of customer models (e.g., there are several metrics for measuring the quality of recommendations), they are quite rudimentary, and there is a strong need to work on better measures. Specifically, the recommender systems community has explored this area.

- *Good actioning mechanisms:* Once data mining has been conducted with promising results, how to use them in the daily performance task is critical and it requires significant research effort. It is common that after some data results are obtained, the domain users do not know how to use them in their daily work. This research may

require the participation of business and marketing researchers. Another way to accommodate actioning mechanisms is to integrate them into the knowledge discovery process by focusing on the discoveries of actionable patterns in customer data. This would make easier for the marketers or other domain experts to determine which actions should be taken once the customer patterns are discovered.

- *Incorporating prior knowledge:* This has always been a problem in practice. Data mining tends to find many pieces of patterns that are already known or redundant. Incorporating prior domain knowledge can help to solve these problems, and also to discover something novel. However, the difficulties of incorporating domain knowledge result in little progress in the past. There are a number of reasons for this. First of all, knowledge acquisition from domain experts is very hard. This is well documented in AI research, especially in the literature of expert systems building. Domain experts may know a lot but are unable to tell. Also, many times, domain experts are not sure what the relevant domain knowledge is, which can be very wide, although the data mining application itself is very narrow. Only after domain experts have seen some discovered patterns then they remember some domain knowledge. The second reason is the algorithmic issue. Many existing methods have difficulty to incorporate sophisticated domain knowledge in the mining algorithm. Also, once the new patterns are discovered, it is important to develop methods that integrate the newly discovered knowledge with the previous knowledge thus enhancing the overall knowledge base. Although there is some general work on knowledge enhancement, much more needs to be done to advance this area and adapt it to CRM problems. Also, integration of these methods with existing and novel Knowledge Management approaches constitutes a fruitful area of research.

6 Foundational Issues for Data Mining

Database Mining has often been described as scalable Machine Learning or statistical modeling, and this is an accurate description in that it captures the main thrust of DM research over the past decade or so. However, is scalability the only characteristic that differentiates DM? The Foundations group discussed this issue at length, and concluded that we need to better understand issues such as compositionality and support for exploratory analysis, if we are to realize the vision of effectively learning from very large datasets.

In applications of DM, the bottle-neck is rarely the execution of the DM algorithm; scalable, fast algorithms are available for a wide range of DM operations. The bottle-necks are the gathering and cleaning of data, which is widely recognized, and the iterative process of selecting a combination of analysis techniques (including simple data selection and transformations such as scaling, as well as DM operations) and an appropriate subset of data to apply these techniques to, in order to extract actionable intelligence from the data. This iterative process involves many people, continues over a period of time, and must deal with an evolving dataset (and therefore "patterns" extracted using different snapshots of data).

Relational database systems, which support applications with similar characteristics, have a compositional algebra at the core of their query language, and this is the source of their flexibility and power. The emphasis on compositionality has led to simple yet powerful query languages which support multi-step analyses, with different steps carried out by different groups. Tools such as OLAP interfaces, which also leverage compositionality of the underlying query operations (selection, grouping, aggregation), have been developed to support "bird's eye" perspectives of data for exploratory applications. Data Mining research has not addressed these issues adequately in the broader setting of learning and statistical modeling algorithms, and we suggest that this is an important future direction. In this workshop, we discussed compositionality in some depth, and developed *signatures* for several DM operations as a first step to understanding how these operations could be composed in applications.

6.1 Compositionality in DM

Compositionality is the technique of constructing complex analyses by using a collection of standard operations as building blocks, with the outputs of some operations serving as inputs to other operations. Relational algebra, consists of the operations of *selecting* some rows of a table, *projecting* out some columns of a table, *joining* two tables on some common columns to produce a combined table, *subtracting* the rows of one table from another (similar) table, and taking the *union* of rows across two similar tables. SQL, the standard relational query language, extends this set of operators,

but retains the key property that the outputs as well as inputs of each operator are simply tables; thus, complex queries can be built by composing simpler queries, starting with single operators applied to a table.

In the context of DM, what are the "operators" of interest, and their inputs and outputs? What is the impact of composition on the statistical properties of extracted patterns? How can compositionality be used to simplify system architectures for DM, and streamline applications of DM over large, time-varying datasets?

6.2 Benefits of Compositionality

What are the expected benefits of a compositional approach to data mining, with a methodology for using multiple operators and techniques based on a sound theory?

- First, it should be possible to provide general *guidelines* and *best practices* for the knowledge discovery process and data mining.
- Second, the compositional approach could guarantee the *completeness* of a given toolkit. Having a complete set of operators, it may be expected that *more* problems can be solved in a *better* way. It would be possible to expose alternatives that analysts would not see as easily without a solution for compositional data mining. As a side effect, improved interfaces to support the knowledge discovery process could be designed.
- Third, the compositional approach would enable the design *of new algorithms* in a principled form, which would also foster reusability.
- Fourth, the compositional approach could improve our *understanding of what is easier or harder computationally*. New kinds of "meta-analyses" based on cost estimation and algebraic properties (e.g., monotonicity and others that go beyond commutativity etc.) would become feasible. Query optimization for database systems involving multiple data processing and mining steps could be explored. This in turn would allow for the *semi-automated exploration of large spaces of analysis plans*. The semi-automated exploration of search spaces has to be based on the trade-off between quality and costs. Moreover, one might characterize the patterns and models in terms of the search space explored.

6.3 Signatures of KDD Operations

We have chosen to describe data mining operations in terms of their *signatures*, that is, in terms of the domain and range of the functions they are computing. The basic idea is that not all operations can be combined in all possible ways meaningfully. For instance, it does not make sense to discretize itemset data. Signatures provide us some guidance as to what can be combined meaningfully.

In order to structure the large number and variety of data processing and mining operations, we organize their signatures in *hierarchies*. The purpose of hierarchies is to order the operators conceptually and also to get a better understanding of the commonalities and differences among them. In the higher levels of the hierarchy, the signatures are described by general terms, e.g., patterns or models. In the lower levels, the signatures may be *specialized* for certain *types of patterns or models*. It is also possible to specialize a signature by *extending the list of parameters* of the algorithms computing the function. For instance, we might have a general signature for a nearest neighbor classifier NN: Set(Data) -> Set(Data). A specialization could include the number of neighbors NN: Set(Data) x Param -> Set(Data) or even the distance measure NN: Set(Data) x Param x Dist -> Set(Data). Including more and more details of an algorithm into a signature might be useful for supporting the semi-automatic exploration of search spaces of multi-step mining procedures.

Data processing and mining operators could be organized in several possible ways:

- according to the *generic basic operations* (possible mappings from data and patterns to data and patterns),
- according to the *type of data* (e.g., multimedia data) or pattern domain (items, strings, graphs, etc.), or
- according to the *type of operation* itself (clustering, classification).

Since we (1) wanted to make the issues and design decisions explicit, and (2) did not want to commit to one choice prematurely, we included all three sorts of hierarchies in the following sections of the report. From a systematic point of view, it would be the best to structure all operations according to a generic hierarchy defined in terms of basic inputs and outputs. However, due to the complexity of the knowledge discovery process, the resulting structure might not be very

user-friendly and “maintainable”, because operations defined for different data types will be found next to each other in the same subtree of the hierarchy. The general idea is to describe the operations in several hierarchies, but to include the interrelationships among them, if there are any. As long as the interrelationships between the hierarchies are clear, it may be feasible to continue this effort for other data types (e.g., itemsets, strings, trees, graphs) and operations (e.g., classification) along these lines.

While signatures appear to be useful for achieving compositionality, additional information is needed for the optimization of multi-step mining tasks. For instance, it would be necessary to include properties such as the time and space complexity of the algorithms or their completeness. If this information were available, it would be possible to derive computational properties of multi-step sequences.

One essential point of the approach to compositionality taken here is that the output of mining algorithms is treated as regular data. In this way, we are circumventing problems with methods like clustering and classification, where logically speaking (at least in their “hard” variants), the data is segregated. In contrast, in the current proposal *the data becomes the bridge*. One advantage is that in this way probabilistic clustering variants are also supported.

It has been observed that some of the operations below are not atomic in the sense that they could be performed by a sequence of other operations. One might be concerned that we are missing the right level of abstraction or granularity of the operations. Also in this respect, we decided to take a “liberal” approach: for the sake of simplicity and “user-friendliness”, we are allowing for some redundancy and non-atomicity, as long as it is documented in the description of the signatures.

In a more or less unordered fashion, we identified several other topics for further discussion:

- Operations can be categorized as preprocessing, data mining, and post-processing operations. *Preprocessing* operations modify the data, *data mining* operations generate patterns or models, and *post-processing* operations deal with the evaluation and validation of patterns and models. Generally, the goal is *to support the whole knowledge discovery process, not just the data mining step* within the process.
- Another idea was to explore large search spaces for multi-step sequences of data processing and mining operations. In this setting, quality and cost measures for evaluating the final result are required.
- One interesting operation not included here would be to transfer “objects” from the realm of patterns/models to the realm of data and vice versa. The general distinction between patterns/models and data, however, seems useful for providing some guidance in the knowledge discovery process on the basis of signatures.

6.4 Hierarchies of Signatures

We begin by sketching a generic hierarchy according of signatures. The numbering scheme (*one to nine*) is also adopted for the data-type- and operation-dependent hierarchies that are presented in subsequent sections. All operations numbered *ten or higher* are specific to the respective types of data or operations (e.g., evaluation or ranking primitives). Next, we present examples of data-dependent signatures. Finally, we introduce operation-dependent signatures.

We use the word “pattern” for both “local patterns” and “models” at the top level of the hierarchy. Another terminological decision was to use the word “predictor” for classification and regression models. Other options would have been “classifier”, which would have been misleading for regression models, and “model”, which, intuitively, would have included clustering.

In Table 1, a generic hierarchy of signatures is shown. The first type of operation involves standard database operations such as selection, projection or joins. The second type includes standard data mining operations. As an illustration, Subitem 2.1 covers pattern discovery, 2.2 covers clustering, and 2.3 covers predictive mining. The third type of operation includes, for instance, the selection of a subset of patterns or models. The fourth type includes the creation of data based on, e.g., a generative model. We cannot include a full list of operations and their explanations here, but the general idea should be clear. The fifth type involves mappings from data and patterns to patterns. For instance, incremental and online learning schemes fall into this category. The sixth type of operation is of particular interest, because it includes the modification of the data on the basis of patterns or models. For instance, a new attribute can be defined in a table of a relational database, depending on whether a pattern occurs or not. Vice versa, a *tuple* may be removed from a table

depending on the occurrence of a pattern. Another example is the definition of a new attribute based on cluster assignments. Finally, operations seven to nine are redundant in the sense that they could be composed from simpler operations. For instance, Set(Data) -> Set(Data) x Set(Pattern) could be composed from Set(Data) -> Set(Data) and Set(Data) -> Set(Pattern). However, it might be more intuitive to the data analyst to perform these data processing or mining operations in one step.

TABLE 1: Generic Hierarchy of Signatures

DecId	Signature
1.	Set(Data) -> Set(Data)
2.	Set(Data) -> Set(Pattern)
2.1	Set(Data) -> Set(LocalPatterns)
2.2	Set(Data) -> Clust
2.3	Set(Data) -> Predictor
3.	Set(Pattern) -> Set(Pattern)
4.	Set(Pattern) -> Set(Data)
5.	Set(Pattern) x Set(Data) -> Set(Pattern)
6.	Set(Pattern) x Set(Data) -> Set(Data)
6.1	"DataExtension" (= extension of data by a new descriptor)
6.1.1	Set(Data) x Set(LocalPattern) -> Set(Data)
6.1.2	Set(Data) x Clust -> Set(Data)
6.1.3	Set(Data) x Predictor -> Set(Data)
6.2	"Data Reduction" (= elimination of *data items*)
6.2.1	Set(Data) x Set(LocalPattern) -> Set(Data)
6.2.2	Set(Data) x Clust -> Set(Data)
6.2.3	Set(Data) x Predictor -> Set(Data)
7.	Set(Data) -> Set(Pattern) x Set(Data)
8.	Set(Pattern) –> Set(Pattern) x Set(Data)
9.	Set(Pattern) x Set(Data) -> Set(Pattern) x Set(Data)

> = 10.- Ranking, Evaluation, etc.

6.5 Examples of Data-Dependent Signatures

In Table 2, we list data-dependent signatures for relational tables. Classical database operations can be found as R.1.1 and R.1.2. R.1.3 seems to be quite practical to make attributes and tuples interchangeable. Also included are segmentation and aggregation operators for data manipulation. R.2 includes typical data mining operations up to the level of classification and regression. R.6 includes data transformation operators using known patterns and models, as already briefly discussed for the generic hierarchy. R.10 to R.13 include operations that are not instantiations of the generic nine possible mappings from data and patterns to data and patterns. R.14 subsumes operations analyzing given predictors. For instance, we can extract important features or important instances from a predictor. This is one of the operations used in the case studies below.

TABLE 2: Signatures for Relational Tables

R.1 Data Manipulation
R.1.1 Selection and Projection
Select: Set(Tuple) x Condition -> Set(Tuple)
Projection:Set(Tuple) x Condition -> Set(Tuple)
R.1.2 Join
Join: Set(Tuple) x Set(Tuple) -> Set(Tuple)
R.1.3 Transpose
Set(Tuple) -> Set(Tuple)
Not atomic, but very useful; makes examples and features interchangeable
R.1.4. Segmentation Methods
Segm: Set(Tuple) -> Set(Set(Tuple))
R.1.5 Aggregation Methods

Aggr: Set(Tuple) -> Set(Tuple)
R.1.6 Feature Selection
Set(Tuple) x FeatureEval -> Set(Tuple)

((FeatureEval: Feature -> Real))

R.2 Data Mining: Set(Tuple) x Param -> Set(Patterns)
R.2.1 PatternDisovery
Set(Data) x Param -> Set(LocalPatterns)
R.2.2 (Hard) Clustering
Set(Tuple) x Param -> Set(Set(Tuple))
R.2.3 PredictiveMining
Set(Tuple) x Param -> Predictor
R.2.3.1 Classification
Set(Tuple) x Param -> Classifier
R.2.3.2 Regression
Set(Tuple) x Param -> Regress
R.6 DataTransformations
R.6.1 TableExtension (Generate additional features)
Set(Tuple) x Set(Pattern) -> Set(Tuple)

R.6.1.1 Set(Tuple) x Set(LocalPattern) -> Set(Tuple)
R.6.1.2 Set(Tuple) x Cluster -> Set(Tuple)

R.6.1.3 Set(Tuple) x Predictor -> Set(Tuple)

R.10 Similarity Metrics:
Sim: Tuple x Tuple -> Real
R.11 Evaluation Primitives (How good is the generated pattern/model?)
- Set(Tuple) x Pattern x EvalMeasure -> Real

R.12 Ranking Primitives (Ranking the generated patterns/models)
- Set(Pattern) -> List(Pattern)

R.13 Comparison Primitives (Comparison of two or more patterns/models)
- Pattern x Pattern x CompMeasure -> Real

The following two operations/operators could also be categorized under “classification”:

R.14 PredictorAnalysis
R.14.1 ImportantInstanceIdentification
Predictor -> Set(Tuple)
R.14.2 ImportantFeatureIdentification
Predictor -> Set(Attribute)

In Table 3, we give another example of a data-dependent hierarchy of signatures. The hierarchy is defined for multi-media, respective feature vector data. Of particular importance are operations for segregating and aggregating the data according to different descriptors. For the purpose of multi-media mining, we use the following data types: multimedia objects MO and feature vectors FV. The general signature for feature extraction is: FeatExt: MO x Int -> FV[Int]. Feature extraction is usually the first step in the knowledge discovery process for multimedia data. Starting with feature extraction, the data are segregated and prepared for the model construction step. After the model construction step, the partial results obtained from different representations are aggregated again to give the final result.

TABLE 3: Signatures for Multimedia Mining
F.1.2 Combination Methods
Comb: FV[Int] x FV[Int]-> FV[Int]
F.1.4. Segmentation Methods

Segm: Set(FV[Int]) -> Set(Set(FV[Int]))
F.1.5 Aggregation Methods
Aggr: Set(FV[Int]) -> Set(FV[Int])
F.1.7 Feature Transformation Methods
Trans: FV[Int] -> FV[Int]
e.g. dimension reduction methods such as PCA, SVD
F.10 Similarity Metrics
Sim: FV[Int] x FV[Int] -> Real
F.12 Rank
Rank: Set(FV[Int]) x FV[Int] x (FV[Int] x FV[Int] -> Real) -> List(FV_id)
F.13 Feature Search
Search: Set(FV[Int]) x FV[Int] -> Bool

6.6 Examples of Operation-Dependent Signatures

In Table 4, we sketch the basic operations involved in clustering. Apart from the actual clustering step (C.2.2), we include operations for, e.g., evaluating (C.11) and optimizing (C.12) clusters. Note also the operator for refining an existing clustering, which is used in the reconstruction of multi-step clustering below. The operator for transforming the data based on a clustering (C.6.1.2) seems to be very useful in applications as well.

TABLE 4: Signatures for Clustering
C.2.2 Cluster Generation:
Set(Data) x Param -> Clust
C.5 Clust x Set(Data) x Param -> Clust
"Cluster Refinement"
C.6 DataTransformations
C.6.1.2 Set(Tuple) x Clust -> Set(Tuple)
C.10 ClusterComparison:
Clust x Clust x SimMeasure -> Real
C.11 ClustEvaluation:
Data x Clust x EvalMeasure -> Real
C.12 ClustOptimization:
Data x (Data x Param -> Clust) x
(Data x Clust x SimMeasure -> Real) -> Clust
Data x (Data x Param -> Clust) x
(Data x Param -> Clust) x
(Data x Clust x EvalMeasure -> Real) -> Clust

6.7 Case Studies

6.7.1 QSAR

The first case study comes from the area of quantitative structure activity relationships (QSAR) [HKK+04], where the goal is to model the biological activity of small molecules. More precisely, we are relating chemical structure to biological activity. Most approaches are based on physico-chemical properties or substructures, either predefined in a library or computed from the data. In the latter case, molecular fragments of interest are generated from a database of chemical compounds. For instance, one might generate all molecular fragments occurring with a minimum frequency in the data. Alternatively, one might generate all fragments frequent in one class of compounds and infrequent in another class. Subsequently, the molecular fragments are used as features to describe the compounds for a machine learning algorithm. For this purpose, each compound is represented as a feature vector of zeros and ones (indicating the absence or presence of a substructure) or counts (indicating the number of occurrences). In the machine learning step, various algorithms have been shown to be useful: rule learning algorithms (e.g., PART), kernel machines (e.g., linear and

quadratic support vector machines) or equation discovery algorithms (LAGRAMGE and CIPER). As an example, it is possible to ask for the two best equations containing only one feature, namely a super-fragment of "c=o", in the system CIPER. This query returns the two equations logHLT = 6.9693 -1.19013*"c=o" and -logHLT = 6.91524 -1.24286*"c-c=o". Finally, the resulting models are analyzed to find the features (substructures) most important for the prediction task.

Therefore, the knowledge discovery can be described by the following steps:

- First, we perform pattern discovery in the form of finding interesting molecular substructures.
- Second, we apply the patterns found in the first step to reformulate the data. That is, we transform the 2D representation of molecular graphs into a feature-vector representation indicating the absence or presence of the patterns in the data.
- Third, a machine learning algorithm is applied to find a predictive model, either for classification or regression.
- Fourth, the predictive model from the previous step is analyzed to return the most important instances or features to the user.

In terms of the operators and signatures given above, the process is summarized in Table 5.

TABLE 5: QSAR by Operators
R.2.1 PatternDiscovery
Set(Data) x Param -> Set(LocalPatterns)

R.6.1.1 Transformation

Set(Tuple) x Set(LocalPattern) -> Set(Tuple)

R.2.3 PredictiveMining
Set(Tuple) x Param -> Predictor
R.14.2 ImportantFeatureIdentification
Predictor -> Set(Attribute)

6.7.2 Gene Expression and Pathological Data

The second case study integrates gene expression and pathological (clinical) data [SKM+04]. The goal was to find characterizations of clusters of liver cancer gene expression measurements in terms of pathological/clinical attributes. In a first step, the database is projected onto the gene expression attributes (ignoring the pathological attributes for the moment). Subsequently, a hard clustering algorithm (such as K-means) is applied. Next, the original data are extended by an attribute indicating the cluster membership. In the fourth step, the original data are projected onto the pathological data and the cluster membership attribute. Finally, a predictive model is induced, mapping the clinical features to the respective group of observations as detected on the molecular level. Table 6 summarizes the case study in terms of the above operators and signatures.

TABLE 6: [SKM+04] by Operators
R.1.1 Project on gene expression data
R.2.2 (Hard) Clustering
Set(Tuple) x Param -> Set(Set(Tuple))
R.6.1.2 Set(Tuple) x Cluster -> Set(Tuple)
generate additional attribute
R.1.1 Project on pathological data
R.2.3.1 Classification
Set(Tuple) x Param -> Classifier

6.7.3 Transcription Factor Data

The third case study is the concerned with the detection of transcription factor binding sites [BJV+98]. Here, the examples are genes, which are described in terms of (1) their expression intensity as measured in microarray experiments and (2) the sequence of their upstream region. A combination of clustering and pattern discovery techniques is applied to

detect transcription factor binding sites on the basis of these two types of data. First, the time series of gene expression data are clustered to find groups of co-regulated genes. Then, for each cluster, the sequences upstream of the genes are retrieved. We can also view this step as the projection of the database on the sequence part. The upstream sequences are compared to find statistically valid patterns. Finally, the results are evaluated by randomization testing and the comparison with known transcription factor binding sites retrieved from public databases. The process includes data preparation, data integration, and data mining steps. For the data preparation (normalization) and data mining steps (clustering and pattern discovery), a variety of different algorithms could be used. Finally, randomization testing is a recurring method in the field of Life Science data analysis. In Table 7, the knowledge discovery process followed in the case study is summarized in terms of the above operators and signatures.

TABLE 7: [BJV+98] by Operators
R.1.1 Project on gene expression data
R.2.2 (Hard) Clustering
Set(Tuple) x Param -> Set(Set(Tuple))
R.1.1 Project on sequence data
For each of the clusters, perform on the sequence data:
R.2.1 PatternDisovery
Set(Data) x Param -> Set(LocalPatterns)

6.7.4 Multistep Clustering

Finally, we give an example for the reconstruction of a data mining algorithm using operators and signatures. In *multi-step clustering*, we perform clustering of complex objects incrementally in several, computationally inexpensive steps. For efficiency reasons, we start clustering the objects in an abstract representation first. Then, the clustering is refined in a more detailed representation to correct potential errors made in the previous iteration. This process is continued at increasing levels of detail until the final, correct clustering is obtained. Table 8 summarizes the reconstruction of multi-step clustering by means of our operators and signatures.

TABLE 8: Multistep Clustering
C.2.2 ClusterGeneration:
Data x Param x SimMeasure -> Clust
C.5 Cluster Refinement Operator:
Clust x Param x SimMeasure -> Clust
C.10 Cluster Comparison (termination condition):
Clust x Clust x SimMeasure -> Real

7 Privacy-Preserving Data Mining

This section is structured as followed. We start by describing application scenarios for privacy-preserving data mining, and then discuss ways to measure privacy. We then summarize our preliminary discussions on extending the definition of a privacy breach from Evfimievski et al. [ESAG02, EGS03] to the data publishing scenario. Next, we discuss research problems in the area of privacy policies, and conclude with a discussion of benchmarks in the area of privacy-preserving data mining.

7.1 Application Scenarios

We discussed application scenarios where different aspects of privacy preserving data mining technology would make a difference.

7.1.1 Information Integration

Information integration involves combining data from several different databases to extract information from their union. Let us mention four examples where information integration has great value, but where the privacy of the data that needs to be integrated needs to be preserved.

Yodlee

Yodlee[6] provide a data integration infrastructure that allow companies to set up integrated information portals that draw upon data from different information sources and give an integrated view of all the data. An example is www.myciti.com, a website that gives customers an integrated view of their bank statements, bills, payments, and investments, among other data. Privacy is of paramount importance, both from the customer's perspective and from the perspective of the data providers whose data is integrated: Yodlee has the mandate of integrating the data, but it should not have access to the actual data that is shown to a customer; it leaves the master copy of the data at the original data source. Yodlee would like to architect the system such that it *cannot* learn private data from the interactions of users with the system and from the data that users are accessing from remote databases.

In addition, the sites from which Yodlee integrates data might have specific privacy policies (for example, expressed in P3P). Integration of this data by Yodlee requires it to adhere to the *integration* of the different privacy policies, a subject we will revisit in more detail in Section 5.

Mergers and Acquisitions

Another example is the sharing of intellectual property for possible mergers and acquisitions and joint projects between two companies. For example, consider two biotech companies that (through high-level discussions) have found out that they have possibly complementary technology to develop a new drug. A privacy-preserving data mining application could analyze the intellectual property of the two companies with the goal to assess the benefits of a joint venture in this area. If there are sufficient commonalities, talks between the companies are triggered. If there is not enough indication for joint possibilities, then none of the two companies learns anything.

Sharing Data between Government Agencies

A similar situation as in the previous section occurs between two government agencies, say agencies *A* and *B*. By law, agencies *A* and *B* are not allowed to share data about US citizens. However, assume that both agency *A* and agency *B* have (incriminating) pieces of information about an individual *i*, however neither of them individually has enough evidence that could convince a judge to obtain a court order for a wiretap. However, if agency *A* and agency *B* would pool their data, sufficient evidence is available about *i* to obtain a court order. Thus *A* and *B* would like to integrate their data in a privacy preserving way. The agencies would like to receive notice if there is enough evidence about an entity (a person, place, event, etc.) in the *union* of the two databases, while not revealing any information in the case that there is not enough evidence (for example, for a law-abiding citizen *j*).

Medical Research

Our last example in the area of information integration is in the medical domain. Consider a researcher that would like to perform research on the co-occurrences of symptoms of a rare disease. This disease has occurred in hospitals all over the United States. The researcher would like to maximize the power of simple hypothesis tests, such as the average blood sugar level is higher for patients who show symptom *A* versus patients who show symptom *B*. For this test, which could for example be performed by a simple *t*-test, we need the average and the variance of the blood sugar level for patients with symptoms *A* and *B*. The power of the test depends on the number of patients in the study; in order to make statistically sound statements, it is imperative to draw upon as large a pool of patients as possible.

Other more sophisticated questions and associated types of analyses might require access to the history of patients in order to quantify how their symptoms are changing over time. Even other types of analyses by nature need to integrate

[6] www.yodlee.com

data from different medical institutions, for example research in epidemiology. In all of these studies, the number of patients directly influences the statistical significance of the study.

However, there are severe restrictions on how patients' data can be shared today. For example, recent legislation such as HIPPA (http://www.hhs.gov/ocr/hipaa) sets standards to guarantee privacy and confidentiality of medical records of patients. However, due to fear of legal repercussions, HIPPA today is implemented by not sharing data at all, and medical institutions do not even want to give out aggregate data for fear of privacy violations. While we believe that privacy of medical data is of utmost importance, privacy-preserving data mining of data from several health care providers could be a great benefit to society. We need techniques that give provable privacy guarantees while being able to provide the answers to complex analysis queries.

7.1.2 Recommender Systems

Enterprises build extensive profiles of their customers with the goal to recommend other products. In such co-called *recommender systems* users pool their profiles and profile similarities and differences are used to make recommendations. Recommender systems are widely used these days, for example the list of "Customers who purchased this book also purchased ..." at amazon.com and TV show recommendations in Tivo. Today's systems collect all user profile data in a central location where the data is mined for recommendations. Is it possible for users to share interests in a privacy-preserving way without revealing their true profiles to a central server?

We can extend the model of simple recommendations to a more general digital "assistant". As an example, consider the digital assistant of Alice. Alice's digital assistant learns Alice's preferences over time, and the assistant should learn over time what Alice values and what not. Based on this customization, the digital assistant should go out on the web and find items (information, products, etc.) that are valuable for Alice, that are new and that Alice might be interested in, or that have changed since the last time Alice has interacted with them. However, the recommendations of the digital assistant will not only depend on having learned from Alice, but its actions are also based on the memory of other users in the system. How to share personal data for such a collaborative network of digital assistants is a challenging research problem.

7.1.3 Watchlist Applications

Consider the Government CAPPS II Program (Computer Assisted Passenger Profiling System). This program will perform background checks on airline passengers at the time of check-in, and it assigns a resulting risk score based on a data mining model. The risk score will be used to select passengers for further screening at the airport. A data mining model for this application could be simply a list of persons with partial information about them (such as name, latest known address, etc.), and scoring against this data mining model could be as simple as the evaluation of the "distance" of a passenger from one of the people on the hotlist. A more sophisticated data mining model could be based on even more private data of persons such as their social network, their web browsing habits etc.

Watchlist applications have privacy concerns both during the model building phase and during the model application phase. During the model building phase, the creator of the watchlist will want to build the model based on as much private data as possible in order to maximize the accuracy of the model. Building such data mining models in a privacy-preserving way is a significant challenge. However, even during the model application phase, a watchlist needs to apply the model in a privacy-preserving fashion. While scoring individuals, the watchlist creator should not be able to learn any information about the individuals that are scored except in the case that the risk score is sufficiently high. Privacy-preserving fuzzy matching of fields such as addresses and names that are often misspelled is a key operation in this application.

7.1.4 Sensor Networks

Networks of small connected sensors that can sense some aspect of their environment will soon become ubiquitous. Examples include networks of cameras, motion detectors, temperature sensors, light sensors, and (in the future) small sensors that will be able to "sniff" the DNA of people around them. If the physical world becomes digitized, location privacy, which we currently take for granted, will become a luxury. Already today, GPS chips in cell phones allow the

determination of a user's location up to a dozen feet, and new startup companies are providing location-based services and tools for tracking the location of users (www.ulocate.com).

Spatio-temporal traces of locations are a great source of data for data mining, however there are strong privacy concerns. For example, a car rental company might be interested in ensuring that its renters did not drive their rental cars out of state, or not on dirt roads unless the rental was an SUV. Besides enforcement of policies, such data is also a valuable source as input for data mining models. For example, by learning the driving patterns in a given area, drivers could be warned when they enter a road segment where other drivers often had to brake suddenly. However, in general car rental companies should not be able to gain access to the exact location of their renters at any time.

Other applications for privacy-preserving analysis of data whose plain publishing would violate location privacy abound. For example, car manufacturers would like to have the ability to continuously collect data from cars in a privacy-preserving way. Such data could help to find out in advance when parts need to be warranted, and they are a source of valuable information for continuous improvement of the next generation of vehicles for the car manufacturer.

7.1.5 Data Publishing

Another application area is data publishing, with the problem of publishing census data the most researched topic in this area. For census data, there are two forces that are pulling in opposite directions. Researchers would like the census data to be published directly as it is collected in order to make the best inferences with the data possible. For example, the government would like publish detailed demographic statistics based for as small a geographic region as possible in order to enable the best possible allocation of public funds. However, individual citizens whose data is contained in the census would like the insurance of protection of their privacy.

7.1.6 Off-Shoring

Over the last decade, the operations of many service activities such as call center operation, tax preparation, claim processing, and medical record preparation have been outsourced to offshore companies. In 1999, the government passed the Gramm-Leach-Bliley Act (also known as the GLB Act), a law that includes provisions to protect consumers' personal financial data held by financial institutions. Based on the GLB Act, it has been argued that no financial data about US citizens is permitted to leave the physical boundaries of US territory. Can techniques for privacy-preserving data mining be used to enable offshore storage and analysis of US data while adhering to the GLB act?

7.2 Defining and Measuring Privacy

In all of the applications discussed in the previous section, privacy is of utmost importance. In order for privacy to be implementable, we need technical definitions that capture our intuition of what it means for data to be private. In this section, we shortly survey several existing ways of measuring privacy, and we describe research directions in defining and measuring privacy.

7.2.1 Confidence Intervals

Agrawal and Srikant suggested to measure privacy through the length of a confidence interval [AS00]. Consider N clients, each having a private value for a numerical attribute X. The clients would like to share their values x_i of X with a server, which then computes some statistics about the distribution of X. However, the clients would like to preserve the privacy of their values x_i. For an intuition of the protocol by Agrawal and Srikant, consider an individual client who has a private value x_i of attribute X. The client computes a randomized value $z_i = x_i + y_i$, where y_i is drawn from a known distribution F_Y. Given that the server knows F_Y, it can now approximate F_X given the values $z_1,\ldots,z_N$. In particular, the server can compute a confidence interval $I(z_i) = [x - (z_i), x + (z_i)]$ such that $x_i \in I$ with probability at least p. The privacy measure of Agrawal and Srikant is the length of this interval $I(z_i)$.

However, using confidence intervals to measure privacy can be misleading if the server has some prior knowledge about the distribution of X [ESAG02]. As a simple example, assume that the randomizing distribution F_Y is the uniform

distribution over [-0.25,0.25], supposedly creating a confidence interval length of 0.5. However, assume that server has prior knowledge that the domain of attribute X is the interval [0,1]. This prior knowledge allows the server to constrain the value of x_i for some values of z_i. Assume that a client sends randomized value $z_i = 1.1$ to the server. Then the server knows with probability 1 that $x_i \in [0.85,1]$, and thus the length of the confidence interval that constrains x_i is only 0.15 instead of 0.5.

7.2.2 Privacy Breaches

Recall the situation from the previous section. Given our required size of 0.5 for the confidence interval for a value x_i, the server's posterior probability of the value x_i being an element of [0.85,1] was with 100% much higher than the required threshold that we would obtain through a confidence interval of size 0.5 with uniform distribution, which is 0.15/0.5 = 30%. Evfimievski et al. introduce the notion of a *privacy breach*, which captures this difference in prior and posterior knowledge about a private value [ESAG02]. Intuitively, a privacy breach with respect to some property $P(x)$ occurs when, for some possible outcome of randomization (e.g., a possible view of the server), the prior probability $P(x)$ was below a given threshold, however the posterior probability of $P(x)$ is higher than a given threshold. In this case a privacy breach has occurred; privacy is preserved if no privacy breach occurs.

7.2.3 Mutual Information

Agrawal and Aggarwal suggests to measure privacy using Shannon's information theory [S49a, S49]. The average amount of information in the nonrandomized attribute X depends on its distribution and is measured by its differential entropy

$$h(X) = \mathop{\mathbf{E}}_{x\sim X}\left(-\log_2 f_X(x)\right) = -\int_{\Omega_X} f_X(x)\log_2 f_X(x)\,dx\,.$$

The average amount of information that remains in X after the randomized attribute Z is disclosed can be measured by the conditional differential entropy

$$h(X|Z) = \mathop{\mathbf{E}}_{(x,z)\sim(X,Z)}\left(-\log_2 f_{X|Z=z}(x)\right) = -\int_{\Omega_{X,Z}} f_{X,Z}(x,z)\log_2 f_{X|Z=z}(x)\,dx\,dz\,.$$

The average information *loss* for X that occurs by disclosing Z can be measured in terms of the difference between the two entropies:

$$I(X;Z) = h(X) - h(X|Z) = \mathop{\mathbf{E}}_{(x,z)\sim(X,Z)} \log_2 \frac{f_{X|Z=z}(x)}{f_X(x)}\,.$$

This quantity is also known as the *mutual information* between random variables X and Z. Agrawal and Aggarwal propose to measure privacy ($\Pi(X)$) and privacy loss ($\Pi(X|Z)$) as follows:

$$\Pi(X) := 2^{h(X)};\ \Pi(X|Z) := 1 - 2^{-I(X;Z)}.$$

However, this information-theoretic measure of privacy also has its difficulties. Intuitively, the above measure ensures that privacy is preserved in the average, but not in the worst case. A more detailed discussion and quantitative analysis of this issue can be found by Evfimievski et al. [EGS03].

7.2.4 Economic Measures of Privacy

Each of the above mentioned research measure privacy as an absolute standard that is either adhered to or compromised. In practice, however, entities might be willing to compromise privacy if they are suitably compensated. Thus assuming

that we can quantify the utility that we gain from the result of the data mining process, we can break the premise that privacy is absolute and we can try to design mechanisms where entities have economic incentive to disclose some properties of their private data, as long as their individual benefit exceeds the perceived costs of loss of privacy.

There exist already many examples where people compromise their privacy to gain economic benefit. As an example, consider frequent flyer programs that include incentives such as double mileage etc. for members of their programs. Another example is frequent shopper cards in grocery stores. Here consumers trade the economic benefit of extra savings in the grocery store versus the creation of a detailed profile of their household's shopping behavior. In all of these cases consumers did not give up their privacy for free, but they participated in a mechanism that had sufficient incentives to motivate clients to give up some privacy.

Kleinberg et al. measure private information by its monetary value [KPR01]. The cost of each piece of information must be determined fairly, so as to reflect the contribution of this piece in the overall profit. The paper discusses two scenarios. In the first scenario, information about participating entities is collected by a single server. In the mechanism, each client has her own preference which it would like the server to adopt. The server will adopt the majority preference over all clients that send their preferences to the server. The server's payoff increases with the number of clients that support the majority preference, an indication of the market size. Each client's payoff is either zero or one, depending on whether the client's preference was chosen by the server. The second scenario considers a recommendation systems and collaborative filtering. Clients pool their information about interesting items at the server with the goal to learn about new items from the server.

This work could be a starting thought for a general privacy model for privacy-preserving data mining, where participating entities disclose private information to a server and then benefit from the data model built by the server. It is an open question how to generalize and extend the results from Kleinberg et al. to other data mining models. Besides the design of a suitable definition that captures the value of privacy data, research is also necessary in the design of efficient mechanisms in which rational participants can achieve a good balance between the release of private data and its protection. For example, we can imagine that business will adopt the idea of transferring liability to another entity which serves as an "insurance" against privacy breaches. For example, the use of economic models for data privacy might allow us to use economics to substitute strong adversary with weaker "rational" adversaries such that in practice there are still very strong incentives against privacy breaches, even though they are not computationally infeasible.

7.3 Limiting Disclosure in Data Publishing

In this section we consider the census problem, which has received a lot of attention in the statistics literature.

7.3.1 Background

To introduce the problem, let us recall the setup from section. We have *N clients* that hold private data x_1, x_2, …, x_N whose union is dataset $D = \cup_{1 \le i \le N} x_i$.[7] The clients respond truthfully to a request from a *server* which collects {x_1, …, x_N} and constructs *D*. The clients permit the server publishing of as close a dataset *D′* to *D* as possible as long as confidentiality of the private data is maintained and the server limits disclosure of *D*. We have *users* of the published data who would like to maximize the utility gained from *D′*, for example they would like to use *D′* to mine aggregate statistics about *D*. It is in the interest of these users that the server publishes *D′* such that the queries from the users can be answered as accurately as possible. Malicious adversaries (with unlimited computational power) will examine the published data *D′* and will try to extract as much private information as possible. An adversary might have additional knowledge besides *D′*, for example, she might know one or more clients and some properties of their values x_i. In extreme cases, an adversary might know properties of x_i for one or several clients *i*.

Disclosure is inappropriate association of information to a client [Fie03]. Feinberg distinguishes identity disclosure (a client is identifiable from *D′*), attribute disclosure (sensitive information about a client is released through *D′*), and inferential disclosure (*D′* enables an adversary to infer attributes of a client more accurately than without *D′*). Note that inferential data disclosure is a very strict notion, and in most cases the only way to ensure no inferential data disclosure is

[7] We make the simplifying assumption that all records are distinct.

by publishing no data at all. Thus instead of completely avoiding inferential disclosure, in practice we are interested in *disclosure limitation*, where we can quantify the information that we have disclosed about D and thus ensure that the clients have not incurred any harm.

As a concrete example of the above setup, consider the U.S. census. In this case, the clients are U.S. citizens who participate in the census. The server is the U.S. Bureau of the Census that collects the data. The users of the data are the public.

7.3.2 Previous Work

The need to publish data while maintaining data privacy has spurred a large body of work in the statistics community. This work focuses on protecting the privacy of sensitive entries in contingency tables [Fel72, Cox80, Cox82, Cox87, DF03, DF00, SF04]. An entry in a contingency table is regarded as sensitive if it has a small value (i.e., only a few tuples in the data take on the attribute values corresponding to the entry under consideration.) Typically, a fixed small threshold is chosen (usually 1 or 2) and contingency table entries with values less than this threshold are considered sensitive and must be hidden. Intuitively, the idea behind considering low count cells as sensitive is that the corresponding attribute values could potentially be used as an approximate key to uniquely identify or narrow down information about an individual through linkage with other databases.

Avoiding Cells with Low Counts

There have been different approaches on how to avoid low counts. We survey several techniques in the next sections; we are to blame for any omissions.

Cell Suppression. One simple approach is to delete all small counts. However, in many cases we can conclude information about the suppressed cells from the margin totals that we are publishing. Thus often we need to further suppress additional cells through complementary suppressions. In addition, finding the minimum number of cells to suppress is an NP-hard problem [AW89]. Besides these problems in creating a dataset with suppressed cells, cell suppression reduces the utility of D' severely.

Output Perturbation. A general framework for output perturbation was introduced by Duncan and Pearson [DP91]. Assume that the original data **D** is an $N \times p$ matrix, where N is the number of clients, and p is the number of attributes in the data. We can now describe output perturbation with the following simple matrix manipulation:

$$\mathbf{D}' = \mathbf{ADB} + \mathbf{C} \qquad (1)$$

The matrices **A**, **B**, and **C** have different functions: Matrix **A** perturbs the input records, matrix **B** transforms the attributes, and matrix **C** adds (random) perturbation to the data.

Data Swapping. In data swapping, we remove counts from one cell and add them to another cell. Through repeated data swapping, we can generate a table that is consistent with a set of released marginals.

Cell Rounding. In this approach, we round the value to a multiple of a pre-defined base number. Thus the final cell value will be an interval that contains the actual value. Intuitively, the larger the base number, the more privacy, however, the loss of information also increases.

Query Set Restriction. In this technique, queries on D are either answered correctly or denied completely. This approach has been thoroughly examined in the literature on statistical database security, and it has been shown that strong attacks against the technique of query set restriction severely limits its applicability [AW89].

Comments. In all previous methods, the goal was to guard against small cell counts by ensuring that all cell entries in the contingency table are above a threshold. However, the choice of threshold is usually made in an ad-hoc manner and none of the existing work provides any algorithms or theoretical foundations for how the threshold is to be chosen. Thus, even though the techniques are powerful and elegant, they do not offer formal privacy guarantees or provide a mathematical specification of the level of privacy that is guaranteed.

***k*-Anonymity**

Samarati and Sweeney introduced the notion of *k*-anonymity. Consider again the set of *N* clients, each having this time a record consisting of several attributes. We distinguish between *quasi-identifying attributes* and *secret attributes*; without loss of generality assume that there is one quasi-identifying attribute X^Q and one secret attribute X^S. Each client sends its data x_i^Q, x_i^S to the server, which assembles the dataset $D = \cup_i \left(x_i^Q, x_i^S \right)$. The server wants to publish a modified dataset D' which does not leak private information in *D*. The privacy notion of *k*-anonymity is met if for every record with value x^Q for attribute X^Q in D' there exist at least $k-1$ other records also with value x^Q for attribute X^Q in D'.

While *k*-anonymity is an easy metric to understand and it intuitively guards against "linking" of D' against another dataset *E* through the quasi-identifying attributes, it is an open research question to determine how *k*-anonymity guards against an adversary that has background knowledge about *D*.

		x_1	x_2	x_3
y_1	z_1	8	5	3
	z_2	4	7	0
y_2	z_1	2	1	0
	z_2	2	4	0
y_3	z_1	0	1	0
	z_2	3	8	2

Figure 1: Tabulated Dataset *D*

	x_1	x_2	x_3
y_1	12	12	3
y_2	4	5	0
y_3	3	8	2

Figure 2: Marginal D_{AB}

7.3.3 New Research Directions

From the discussion in the previous section we see that the research community has developed very interesting ideas for preserving privacy in data publishing. In addition, we have powerful mathematical tools that help us quantify what disclosure we undertake when publishing contingency tables. However, formal notions of privacy in data publishing do not exist yet, and we believe that one important avenue of future work research is formal yet practical definitions of privacy and disclosure in data publishing.

A notable difference is the recent work on query view security by Miklau and Suciu [MS04] a first attempt to give a formal definition of privacy in data publishing. Miklay and Suciu give a formal definition of data privacy, stating that privacy is enforced if the prior and posterior information about a client are not changed by exposing D'. While this work characterizes the set of queries answerable without breaching privacy, its privacy definition is very strict – a query is only permissible if the adversary's belief for all possible properties of the data remains the same after disclosing D'. Under this definition, most interesting queries, for instance even most aggregate queries about D' are disallowed.

One Possible Approach Based on Privacy Breaches

During the workshop, we explored a relaxed definition of disclosure in data publishing based on the notion of a privacy breach from Evfimievski et al. [EGS03] and ideas from secure data publishing in Miklau et al. [MS04]. In the remainder of this section, we will outline our very preliminary thoughts that we had on how such a relaxation could happen.

Let us start with a simple example. Consider a relation D with three categorical attributes X, Y, and Z that can take three, three, and two values, respectively. Thus any record in the relation D can take $3 \times 3 \times 2 = 18$ distinct values. We define the *contingency* table for relation D as the $3 \times 3 \times 2$ table $T(D)$ giving the number of tuples in the relation taking each of the 18 different values for their attributes. Figure 1 shows an example table $T(D)$ which is the result of the following SQL Query:

```
Select X, Y, Z, COUNT(*)
From D
Group By X, Y, Z
```

Now consider the following simple *COUNT* query on the relation D:

```
Select D.X, D.Y, COUNT(*)
From D
Group By X, Y
```

This query returns as answer the table $T_{XY}(D)$ shown in Figure 2. Every cell in this table corresponds to one X,Y-value combination, where the number in a cell indicates the number of records having value $X = x$ and $Y = y$. The table $T_{XY}(D)$ is often also called the XY-marginal of D. Does publishing of $T_{XY}(D)$ cause a privacy breach? For example, the a-priori probability of a record having value $X = x_1 \wedge Y = y_1$ might have been 0.05, whereas $T_{XY}(D)$ shows that in D this probability is $12/39 \approx 0.3$. Thus by publishing $T_{XY}(D)$ the belief of an attacked about the likelihood of a given property about D has changed significantly.

This observation motivates an application of the notion of privacy breaches. Let us give a general definition that assume that there is a general class of queries Q such that the server would like the adversary should not to gain much knowledge about the answers of Q on D by publishing D'. In other words, by looking at D', the adversary's belief about what the true answers to Q on D should not change significantly.

Definition 4.1 (Privacy Breach). *Assume that we are given a database D and a class of sensitive queries Q. Let $D' = f(D)$ be a dataset that we publish. Then given parameters $0 < \rho_1 < \rho_2 < 1$, we say that D' has caused a privacy breach if for some $q \in Q$ the following holds*:

$$P[q(D)] \leq \rho_1 \Rightarrow P[q(D)|D'] \geq \rho_2 \qquad (2)$$

Note that we can *test* whether publishing a set of marginals causes a privacy breach: We need to calculate (or assume that we know from general knowledge) the a-priori $(P[q(D)])$ and the a-posteriori probability $P[q(D)|D']$.

Open Problems

With this definition of a privacy breach, we can now define the following optimization problem: Given parameters $0 < \rho_1 < \rho_2 < 1$, find the *maximal* set of marginals which does not cause a privacy breach. Note that this is very different from the problem of checking whether publishing a set of marginals causes a privacy breach (although a solution to this problem could be a valuable subroutine). One insight here is that *not* publishing a more specific set of marginals leaks some information and this has to be incorporated into our calculations. The optimization problem in the previous paragraph is only one instance of a plethora of new research problems in this space. First, we can relax the restriction that we are only looking at COUNT queries, and we can consider a other and larger classes of queries. Second, our privacy definition is only one of several possible definitions of privacy, and the exploration of other definitions is an open challenge.

Another important problem arises when we consider longitudinal data. In all our examples, our data D was static. However, in all real-life instances of the problem, the data is subject to constant change, and we have the requirement to *re-publish* updated instances of D over time.

Beyond Numerical Data. While we have concentrated on the problem of limiting disclosure when publishing numerical data, similar problems exist with non-numerical data (for example text or geospatial data). It is an open research problem how to define privacy and utility when publishing such data; one challenge is that the class of possible queries that users might have over such data is much broader than simple COUNT queries against cells in a contingency table.

7.4 Specifying and Enforcing Privacy Policies

There was much agreement but little discussion on research into the specification and enforcement of privacy policies. We need research into languages that allow simple specifications of privacy policies and research into tools that allow us to verify that a policy captures the intent. A first thought was that a tool that would take a privacy policy specification and give concrete instances of actions that are permitted and disallowed would be valuable for end-users.

7.5 Benchmarks

We believe that it is important that the community identifies benchmark datasets on which we can measure the performance of privacy-preserving data mining algorithms. These datasets can be either tied to a specific application scenario (e.g., an application scenario from Section 2) or they can be associated with a given technical challenge.

8 References

[AA01] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th Symposium on Principles of Database Systems*, 2001.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy preserving data mining. In *Proceedings of the 19th ACM SIGMOD Conference on Management of Data*, 2000.

[AW89] Nabil R. Adam and John C. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing. Surv*eys, 21(4):515–556, 1989.

[BJV+98] Brazma A, Jonassen I, Vilo J, Ukkonen E. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research 8(11),* pp. 1202-1215, 1998.

[BKS03] Buhler, J., Keich, U. and Sun, Y. (2003). "Designing Seeds for Similarity Search in Genomic DNA". RECOMB 2003, Berlin, Germany.

[Cox80] L. H. Cox. Suppression, methodology and statistical disclosure control. *Journal of the American Statistical Association*, 75:377–385, 1980.

[Cox82] L. H. Cox. Solving confidentiality protection problems in tabulations using network optimization: A network model for cell suppression in the US economic censuses. In *Proceedings of the International Seminar on Statistical Confidentiality*, 1982.

[Cox87] L. H. Cox. New results in dislosure avoidance for tabulations. In *International Statistical Institute Proceedings of the 46th Session*, 1987.

[DF00] A. Dobra and S. E. Feinberg. Assessing the risk of disclosure of confidential categorical data. *Bayesian Statistics 7, Oxford University Press*, 2000.

[DF03] A. Dobra and S. E. Feinberg. Bounding entries in multi-way contingency tables given a set of marginal totals. In *Foundations of Statistical Inference: Proceedings of the Shoresh Conference, Springer Verlag*, 2003.

[DMB+03] Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., et al. PreBIND and Textomy—Mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics 4(1): 11*, 2003.

[DP91] G. T. Duncan and R.W. Pearson. Enhancing access to microdata while protecting confidentiality. *Statistical Science*, 6:219–239, 1991.

[EGS03] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2003.

[ESA+02] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining*, 2002.

[Fel72] I. P. Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 67:337:7–18, 1972.

[Fie03] Stephen E. Fienberg. *Encyclopedia of Social Measurement*, chapter Confidentiality and disclosure limitation. *Academic Press*, 2003.

[GSS03] Gat-Viks I. , Sharan R., Shamir R. Scoring clustering solutions by their biological relevance. *Bioinformatics 19(18):* 2381-2389, 2003.

[Go01] The GeneOntology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research 11(8):* 1425-33, 2001.

[HBP+05] Hakenberg, J., Bickel, S., Plake, C., Brefeld, U., Zahn, H., Faulstich, L., Leser, U. and Scheffer, T. (2005). "Systematic Feature Evaluation for Gene Name Recognition." *BMC Bioinformatics* 6 (Suppl 1):S9.

[HKK+04] Helma C, Kramer T, Kramer S, DeRaedt L. Data Mining and Machine Learning Techniques for the Identification of Mutagenicity Inducing Substructures and Structure--Activity Relationships of Noncongeneric Compounds, *J. Chem. Inf. Comput. Sci. 44*, 1402-1411, 2004.

[JLK+01] Jenssen, T. K., Laegreid, A., Komorowski, J. and Hovig, E. A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet 28(1):* 21-8, 2001.

[Ken02] Kent, W. J. BLAT--the BLAST-like alignment tool. *Genome Res* 12(4): 656-664, 2002.

[KOT+03] Kim, J. D., Ohta, T., Tateisi, Y., Tsujii, J., GENIA corpus-a semantically annotated corpus for bio-textmining. *Bioinformatics 19, Suppl 1*: I180-I182, 2003.

[KPR01] Jon Kleinberg, Christos H. Papadimitriou, and Prabhakar Raghavan. On the value of private information. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, 2001.

[KKT03] Koike, A., Kobayashi, Y. and Takagi, T. Kinase pathway database: an integrated protein-kinase and NLP-based protein-interaction resource. *Genome Research 13(6A):* 1231-43, 2003.

[MS04] Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, 2004.

[SKM+04] J. Sese, Kurokawa, Y. , Monden, M., Kato, K. and Morishita, S. Constrained clusters of gene expression profiles with pathological features. *Bioinformatics, Oxford University Press*, 2004.

[SF04] A. Slavkovic and S. E. Feinberg. Bounds for cell entries in two-way tables given conditional relative frequencies. In *Privacy in Statistical Databases*, 2004.

[Sha49] Claude Elwood Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28-4:656–715, 1949.

[SW49] Claude Elwood Shannon and Warren Weaver. The Mathematical Theory of Communication. University of Illinois Press, 1949.