3

Data Mining: An Overview

The process of understanding the patterns found in large data sets is known as data mining. Some of the aspects of data mining that have been elucidated in the following section are association rule learning, cluster analysis, regression analysis, automatic summarization and examples of data mining. The chapter on data mining offers an insightful focus, keeping in mind the complex subject matter.

Data Mining

Data mining is an interdisciplinary subfield of computer science. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.

The term is a misnomer, because the goal is the extraction of patterns and knowledge from large amounts of data, not the extraction (*mining*) of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence, machine learning, and business intelligence. The book *Data mining: Practical machine learning tools and techniques with Java* (which covers mostly machine learning material) was originally to be named just *Practical machine learning*, and the term *data mining* was only added for marketing reasons. Often the more general terms (*large scale*) *data analysis* and *analytics* – or, when referring to actual methods, *artificial intelligence* and *machine learning* – are more appropriate.

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection), and dependencies (association rule mining). This usually involves using database techniques such as spatial indices. These patterns can then be seen as a kind of summary of the input data, and may be used in further analysis or, for example, in machine learning and predictive analytics. For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system. Neither the data collection, data preparation, nor result interpretation and reporting is part of the data mining step, but do belong to the overall KDD process as additional steps.

The related terms *data dredging*, *data fishing*, and *data snooping* refer to the use of data mining

methods to sample parts of a larger population data set that are (or may be) too small for reliable statistical inferences to be made about the validity of any patterns discovered. These methods can, however, be used in creating new hypotheses to test against the larger data populations.

Etymology

In the 1960s, statisticians used terms like "Data Fishing" or "Data Dredging" to refer to what they considered the bad practice of analyzing data without an a-priori hypothesis. The term "Data Mining" appeared around 1990 in the database community. For a short time in 1980s, a phrase "database mining", was used, but since it was trademarked by HNC, a San Diego-based company, to pitch their Database Mining Workstation; researchers consequently turned to "data mining". Other terms used include Data Archaeology, Information Harvesting, Information Discovery, Knowledge Extraction, etc. Gregory Piatetsky-Shapiro coined the term "Knowledge Discovery in Databases" for the first workshop on the same topic (KDD-1989) and this term became more popular in AI and Machine Learning Community. However, the term data mining became more popular in the business and press communities. Currently, Data Mining and Knowledge Discovery are used interchangeably. Since about 2007, "Predictive Analytics" and since 2011, "Data Science" terms were also used to describe this field.

In the Academic community, the major forums for research started in 1995 when the First International Conference on Data Mining and Knowledge Discovery (KDD-95) was started in Montreal under AAAI sponsorship. It was co-chaired by Usama Fayyad and Ramasamy Uthurusamy. A year later, in 1996, Usama Fayyad launched the journal by Kluwer called Data Mining and Knowledge Discovery as its founding Editor-in-Chief. Later he started the SIGKDDD Newsletter SIGKDD Explorations. The KDD International conference became the primary highest quality conference in Data Mining with an acceptance rate of research paper submissions below 18%. The Journal Data Mining and Knowledge Discovery is the primary research journal of the field.

Background

The manual extraction of patterns from data has occurred for centuries. Early methods of identifying patterns in data include Bayes' theorem (1700s) and regression analysis (1800s). The proliferation, ubiquity and increasing power of computer technology has dramatically increased data collection, storage, and manipulation ability. As data sets have grown in size and complexity, direct "hands-on" data analysis has increasingly been augmented with indirect, automated data processing, aided by other discoveries in computer science, such as neural networks, cluster analysis, genetic algorithms (1950s), decision trees and decision rules (1960s), and support vector machines (1990s). Data mining is the process of applying these methods with the intention of uncovering hidden patterns in large data sets. It bridges the gap from applied statistics and artificial intelligence (which usually provide the mathematical background) to database management by exploiting the way data is stored and indexed in databases to execute the actual learning and discovery algorithms more efficiently, allowing such methods to be applied to ever larger data sets.

Process

The Knowledge Discovery in Databases (KDD) process is commonly defined with the stages:

(1) Selection

(2) Pre-processing

(3) Transformation

(4) *Data Mining*

(5) Interpretation/Evaluation.

It exists, however, in many variations on this theme, such as the Cross Industry Standard Process for Data Mining (CRISP-DM) which defines six phases:

(1) Business Understanding

(2) Data Understanding

(3) Data Preparation

(4) Modeling

(5) Evaluation

(6) Deployment

or a simplified process such as (1) pre-processing, (2) data mining, and (3) results validation.

Polls conducted in 2002, 2004, 2007 and 2014 show that the CRISP-DM methodology is the leading methodology used by data miners. The only other data mining standard named in these polls was SEMMA. However, 3–4 times as many people reported using CRISP-DM. Several teams of researchers have published reviews of data mining process models, and Azevedo and Santos conducted a comparison of CRISP-DM and SEMMA in 2008.

Pre-processing

Before data mining algorithms can be used, a target data set must be assembled. As data mining can only uncover patterns actually present in the data, the target data set must be large enough to contain these patterns while remaining concise enough to be mined within an acceptable time limit. A common source for data is a data mart or data warehouse. Pre-processing is essential to analyze the multivariate data sets before data mining. The target set is then cleaned. Data cleaning removes the observations containing noise and those with missing data.

Data Mining

Data mining involves six common classes of tasks:

- Anomaly detection (Outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.
- Association rule learning (Dependency modelling) – Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is some-

times referred to as market basket analysis.

- Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
- Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".
- Regression – attempts to find a function which models the data with the least error.
- Summarization – providing a more compact representation of the data set, including visualization and report generation.

Results Validation

An example of data produced by data dredging through a bot operated by statistician Tyler Viglen, apparently showing a close link between the best word winning a spelling bee competition and the number of people in the United States killed by venomous spiders. The similarity in trends is obviously a coincidence.

Data mining can unintentionally be misused, and can then produce results which appear to be significant; but which do not actually predict future behaviour and cannot be reproduced on a new sample of data and bear little use. Often this results from investigating too many hypotheses and not performing proper statistical hypothesis testing. A simple version of this problem in machine learning is known as overfitting, but the same problem can arise at different phases of the process and thus a train/test split - when applicable at all - may not be sufficient to prevent this from happening.

The final step of knowledge discovery from data is to verify that the patterns produced by the data mining algorithms occur in the wider data set. Not all patterns found by the data mining algorithms are necessarily valid. It is common for the data mining algorithms to find patterns in the training set which are not present in the general data set. This is called overfitting. To overcome this, the evaluation uses a test set of data on which the data mining algorithm was not trained. The learned patterns are applied to this test set, and the resulting output is compared to the desired output. For example, a data mining algorithm trying to distinguish "spam" from "legitimate" emails would be trained on a training set of sample e-mails. Once trained, the learned patterns would be applied to the test set of e-mails on which it had *not* been trained. The accuracy of the patterns can then be measured from how many e-mails they correctly classify. A number of statistical methods may be used to evaluate the algorithm, such as ROC curves.

If the learned patterns do not meet the desired standards, subsequently it is necessary to re-evaluate and change the pre-processing and data mining steps. If the learned patterns do meet the desired standards, then the final step is to interpret the learned patterns and turn them into knowledge.

Research

The premier professional body in the field is the Association for Computing Machinery's (ACM) Special Interest Group (SIG) on Knowledge Discovery and Data Mining (SIGKDD). Since 1989 this ACM SIG has hosted an annual international conference and published its proceedings, and since 1999 it has published a biannual academic journal titled "SIGKDD Explorations".

Computer science conferences on data mining include:

- CIKM Conference – ACM Conference on Information and Knowledge Management
- DMIN Conference – International Conference on Data Mining
- DMKD Conference – Research Issues on Data Mining and Knowledge Discovery
- DSAA Conference – IEEE International Conference on Data Science and Advanced Analytics
- ECDM Conference – European Conference on Data Mining
- ECML-PKDD Conference – European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
- EDM Conference – International Conference on Educational Data Mining
- INFOCOM Conference – IEEE INFOCOM
- ICDM Conference – IEEE International Conference on Data Mining
- KDD Conference – ACM SIGKDD Conference on Knowledge Discovery and Data Mining
- MLDM Conference – Machine Learning and Data Mining in Pattern Recognition
- PAKDD Conference – The annual Pacific-Asia Conference on Knowledge Discovery and Data Mining
- PAW Conference – Predictive Analytics World
- SDM Conference – SIAM International Conference on Data Mining (SIAM)
- SSTD Symposium – Symposium on Spatial and Temporal Databases
- WSDM Conference – ACM Conference on Web Search and Data Mining

Data mining topics are also present on many data management/database conferences such as the ICDE Conference, SIGMOD Conference and International Conference on Very Large Data Bases

Standards

There have been some efforts to define standards for the data mining process, for example the 1999 European Cross Industry Standard Process for Data Mining (CRISP-DM 1.0) and the 2004 Java Data Mining standard (JDM 1.0). Development on successors to these processes (CRISP-DM 2.0 and JDM 2.0) was active in 2006, but has stalled since. JDM 2.0 was withdrawn without reaching a final draft.

For exchanging the extracted models – in particular for use in predictive analytics – the key standard is the Predictive Model Markup Language (PMML), which is an XML-based language developed by the Data Mining Group (DMG) and supported as exchange format by many data mining applications. As the name suggests, it only covers prediction models, a particular data mining task of high importance to business applications. However, extensions to cover (for example) subspace clustering have been proposed independently of the DMG.

Notable Uses

Data mining is used wherever there is digital data available today. Notable examples of data mining can be found throughout business, medicine, science, and surveillance.

Privacy Concerns and Ethics

While the term "data mining" itself has no ethical implications, it is often associated with the mining of information in relation to peoples' behavior (ethical and otherwise).

The ways in which data mining can be used can in some cases and contexts raise questions regarding privacy, legality, and ethics. In particular, data mining government or commercial data sets for national security or law enforcement purposes, such as in the Total Information Awareness Program or in ADVISE, has raised privacy concerns.

Data mining requires data preparation which can uncover information or patterns which may compromise confidentiality and privacy obligations. A common way for this to occur is through data aggregation. Data aggregation involves combining data together (possibly from various sources) in a way that facilitates analysis (but that also might make identification of private, individual-level data deducible or otherwise apparent). This is not data mining *per se*, but a result of the preparation of data before – and for the purposes of – the analysis. The threat to an individual's privacy comes into play when the data, once compiled, cause the data miner, or anyone who has access to the newly compiled data set, to be able to identify specific individuals, especially when the data were originally anonymous.

It is recommended that an individual is made aware of the following before data are collected:

- the purpose of the data collection and any (known) data mining projects;
- how the data will be used;
- who will be able to mine the data and use the data and their derivatives;
- the status of security surrounding access to the data;
- how collected data can be updated.

Data may also be modified so as to *become* anonymous, so that individuals may not readily be identified. However, even "de-identified"/"anonymized" data sets can potentially contain enough information to allow identification of individuals, as occurred when journalists were able to find several individuals based on a set of search histories that were inadvertently released by AOL.

The inadvertent revelation of personally identifiable information leading to the provider violates

Fair Information Practices. This indiscretion can cause financial, emotional, or bodily harm to the indicated individual. In one instance of privacy violation, the patrons of Walgreens filed a lawsuit against the company in 2011 for selling prescription information to data mining companies who in turn provided the data to pharmaceutical companies.

Situation in Europe

Europe has rather strong privacy laws, and efforts are underway to further strengthen the rights of the consumers. However, the U.S.-E.U. Safe Harbor Principles currently effectively expose European users to privacy exploitation by U.S. companies. As a consequence of Edward Snowden's Global surveillance disclosure, there has been increased discussion to revoke this agreement, as in particular the data will be fully exposed to the National Security Agency, and attempts to reach an agreement have failed.

Situation in The United States

In the United States, privacy concerns have been addressed by the US Congress via the passage of regulatory controls such as the Health Insurance Portability and Accountability Act (HIPAA). The HIPAA requires individuals to give their "informed consent" regarding information they provide and its intended present and future uses. According to an article in *Biotech Business Week', "'[i]n practice, HIPAA may not offer any greater protection than the longstanding regulations in the research arena,' says the AAHC. More importantly, the rule's goal of protection through informed consent is undermined by the complexity of consent forms that are required of patients and participants, which approach a level of incomprehensibility to average individuals." This underscores the necessity for data anonymity in data aggregation and mining practices.*

U.S. information privacy legislation such as HIPAA and the Family Educational Rights and Privacy Act (FERPA) applies only to the specific areas that each such law addresses. Use of data mining by the majority of businesses in the U.S. is not controlled by any legislation.

Copyright Law

Situation in Europe

Due to a lack of flexibilities in European copyright and database law, the mining of in-copyright works such as web mining without the permission of the copyright owner is not legal. Where a database is pure data in Europe there is likely to be no copyright, but database rights may exist so data mining becomes subject to regulations by the Database Directive. On the recommendation of the Hargreaves review this led to the UK government to amend its copyright law in 2014 to allow content mining as a limitation and exception. Only the second country in the world to do so after Japan, which introduced an exception in 2009 for data mining. However, due to the restriction of the Copyright Directive, the UK exception only allows content mining for non-commercial purposes. UK copyright law also does not allow this provision to be overridden by contractual terms and conditions. The European Commission facilitated stakeholder discussion on text and data mining in 2013, under the title of Licences for Europe. The focus on the solution to this legal issue being licences and not limitations and exceptions led to representatives of universities, researchers, libraries, civil society groups and open access publishers to leave the stakeholder dialogue in May 2013.

Situation in The United States

By contrast to Europe, the flexible nature of US copyright law, and in particular fair use means that content mining in America, as well as other fair use countries such as Israel, Taiwan and South Korea is viewed as being legal. As content mining is transformative, that is it does not supplant the original work, it is viewed as being lawful under fair use. For example, as part of the Google Book settlement the presiding judge on the case ruled that Google's digitisation project of in-copyright books was lawful, in part because of the transformative uses that the digitisation project displayed - one being text and data mining.

Software

Free Open-source Data Mining Software and Applications

The following applications are available under free/open source licenses. Public access to application sourcecode is also available.

- Carrot2: Text and search results clustering framework.
- Chemicalize.org: A chemical structure miner and web search engine.
- ELKI: A university research project with advanced cluster analysis and outlier detection methods written in the Java language.
- GATE: a natural language processing and language engineering tool.
- KNIME: The Konstanz Information Miner, a user friendly and comprehensive data analytics framework.
- Massive Online Analysis (MOA): a real-time big data stream mining with concept drift tool in the Java programming language.
- ML-Flex: A software package that enables users to integrate with third-party machine-learning packages written in any programming language, execute classification analyses in parallel across multiple computing nodes, and produce HTML reports of classification results.
- MLPACK library: a collection of ready-to-use machine learning algorithms written in the C++ language.
- MEPX - cross platform tool for regression and classification problems based on a Genetic Programming variant.
- NLTK (Natural Language Toolkit): A suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python language.
- OpenNN: Open neural networks library.
- Orange: A component-based data mining and machine learning software suite written in the Python language.
- R: A programming language and software environment for statistical computing, data mining, and graphics. It is part of the GNU Project.

- scikit-learn is an open source machine learning library for the Python programming language
- Torch: An open source deep learning library for the Lua programming language and scientific computing framework with wide support for machine learning algorithms.
- UIMA: The UIMA (Unstructured Information Management Architecture) is a component framework for analyzing unstructured content such as text, audio and video – originally developed by IBM.
- Weka: A suite of machine learning software applications written in the Java programming language.

Proprietary Data-mining Software and Applications

The following applications are available under proprietary licenses.

- Angoss KnowledgeSTUDIO: data mining tool provided by Angoss.
- Clarabridge: enterprise class text analytics solution.
- HP Vertica Analytics Platform: data mining software provided by HP.
- IBM SPSS Modeler: data mining software provided by IBM.
- KXEN Modeler: data mining tool provided by KXEN.
- LIONsolver: an integrated software application for data mining, business intelligence, and modeling that implements the Learning and Intelligent OptimizatioN (LION) approach.
- Megaputer Intelligence: data and text mining software is called PolyAnalyst.
- Microsoft Analysis Services: data mining software provided by Microsoft.
- NetOwl: suite of multilingual text and entity analytics products that enable data mining.
- OpenText™ Big Data Analytics: Visual Data Mining & Predictive Analysis by Open Text Corporation
- Oracle Data Mining: data mining software by Oracle.
- PSeven: platform for automation of engineering simulation and analysis, multidisciplinary optimization and data mining provided by DATADVANCE.
- Qlucore Omics Explorer: data mining software provided by Qlucore.
- RapidMiner: An environment for machine learning and data mining experiments.
- SAS Enterprise Miner: data mining software provided by the SAS Institute.
- STATISTICA Data Miner: data mining software provided by StatSoft.
- Tanagra: A visualisation-oriented data mining software, also for teaching.

Marketplace Surveys

Several researchers and organizations have conducted reviews of data mining tools and surveys of data miners. These identify some of the strengths and weaknesses of the software packages. They also provide an overview of the behaviors, preferences and views of data miners. Some of these reports include:

- Hurwitz Victory Index: Report for Advanced Analytics as a market research assessment tool, it highlights both the diverse uses for advanced analytics technology and the vendors who make those applications possible.Recent-research
- 2011 Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery
- Rexer Analytics Data Miner Surveys (2007–2013)
- Forrester Research 2010 Predictive Analytics and Data Mining Solutions report
- Gartner 2008 "Magic Quadrant" report
- Robert A. Nisbet's 2006 Three Part Series of articles "Data Mining Tools: Which One is Best For CRM?"
- Haughton et al.'s 2003 Review of Data Mining Software Packages in *The American Statistician*
- Goebel & Gruenwald 1999 "A Survey of Data Mining a Knowledge Discovery Software Tools" in SIGKDD Explorations

Anomaly Detection

In data mining, anomaly detection (also outlier detection) is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.

In particular, in the context of abuse and network intrusion detection, the interesting objects are often not *rare* objects, but unexpected *bursts* in activity. This pattern does not adhere to the common statistical definition of an outlier as a rare object, and many outlier detection methods (in particular unsupervised methods) will fail on such data, unless it has been aggregated appropriately. Instead, a cluster analysis algorithm may be able to detect the micro clusters formed by these patterns.

Three broad categories of anomaly detection techniques exist. Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set. Supervised anomaly detection techniques require a data set that has been labeled as "normal" and "abnormal" and involves training a classifier (the key difference to many other statistical classification problems is the inherent unbalanced nature of outlier detection).

Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given *normal* training data set, and then testing the likelihood of a test instance to be generated by the learnt model.

Applications

Anomaly detection is applicable in a variety of domains, such as intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, and detecting Eco-system disturbances. It is often used in preprocessing to remove anomalous data from the dataset. In supervised learning, removing the anomalous data from the dataset often results in a statistically significant increase in accuracy.

Popular Techniques

Several anomaly detection techniques have been proposed in literature. Some of the popular techniques are:

- Density-based techniques (k-nearest neighbor, local outlier factor, and many more variations of this concept).
- Subspace- and correlation-based outlier detection for high-dimensional data.
- One class support vector machines.
- Replicator neural networks.
- Cluster analysis-based outlier detection.
- Deviations from association rules and frequent itemsets.
- Fuzzy logic based outlier detection.
- Ensemble techniques, using feature bagging, score normalization and different sources of diversity.

The performance of different methods depends a lot on the data set and parameters, and methods have little systematic advantages over another when compared across many data sets and parameters.

Application to Data Security

Anomaly detection was proposed for intrusion detection systems (IDS) by Dorothy Denning in 1986. Anomaly detection for IDS is normally accomplished with thresholds and statistics, but can also be done with soft computing, and inductive learning. Types of statistics proposed by 1999 included profiles of users, workstations, networks, remote hosts, groups of users, and programs based on frequencies, means, variances, covariances, and standard deviations. The counterpart of anomaly detection in intrusion detection is misuse detection.

Software

ELKI is an open-source Java data mining toolkit that contains several anomaly detection algorithms, as well as index acceleration for them.

Association Rule Learning

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal et al. introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule $\{\text{onions}, \text{potatoes}\} \Rightarrow \{\text{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production, and bioinformatics. In contrast with sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Definition

Example database with 5 transactions and 5 items					
transaction ID	milk	bread	butter	beer	diapers
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

Following the original definition by Agrawal et al. the problem of association rule mining is defined as:

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called *items*.

Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the *database*.

Each *transaction* in D has a unique transaction ID and contains a subset of the items in I.

A *rule* is defined as an implication of the form:

$$X \Rightarrow Y$$

Where $X, Y \subseteq I$ and $X \cap Y = \varnothing$..

Every rule is composed by two different sets of items, also known as *itemsets*, X and Y, where X is called *antecedent* or left-hand-side (LHS) and Y *consequent* or right-hand-side (RHS).

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is $I = \{\text{milk}, \text{bread}, \text{butter}, \text{beer}, \text{diapers}\}$ and in the table is shown a small database containing the items, where, in each entry, the value 1 means the presence of the item in the corresponding transaction, and the value 0 represents the absence of an item in that transaction.

An example rule for the supermarket could be $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ meaning that if butter and bread are bought, customers also buy milk.

Note: this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and data-sets often contain thousands or millions of transactions.

Useful Concepts

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on support and confidence.

Let X be an item-set, $X \Rightarrow Y$ an association rule and T a set of transactions of a given database.

Support

Support is an indication of how frequently the item-set appears in the database.

The support value of X with respect to T is defined as the proportion of transactions in the database which contains the item-set X. In formula: $\text{supp}(X) / N$

In the example database, the item-set $\{\text{beer}, \text{diapers}\}$ has a support of since it occurs in 20% of all transactions (1 out of 5 transactions). The argument of $\text{supp}()$ is a set of preconditions, and thus becomes more restrictive as it grows (instead of more inclusive).

Confidence

Confidence is an indication of how often the rule has been found to be true.

The *confidence* value of a rule, $X \Rightarrow Y$, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y.

Confidence is defined as:

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X).$$

For example, the rule $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ has a confidence of $0.2 >$ in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well).

Note that $\text{supp}(X \cup Y)$ means the support of the union of the items in X and Y. This is somewhat confusing since we normally think in terms of probabilities of events and not sets of items. We can rewrite $\text{supp}(X \cup Y)$ as the joint probability $P(E_X \cap E_Y)$, where E_X and E_Y are the events that a transaction contains itemset X or Y, respectively.

Thus confidence can be interpreted as an estimate of the conditional probability $P(E_Y \mid E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Lift

The *lift* of a rule is defined as:

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

or the ratio of the observed support to that expected if X and Y were independent.

For example, the rule $\{\text{milk}, \text{bread}\} \Rightarrow \{\text{butter}\}$ has a lift of $\frac{0.2}{0.4 \times 0.4} = 1.25$.

If the rule had a lift of 1, it would imply that the probability of occurrence of the antecedent and that of the consequent are independent of each other. When two events are independent of each other, no rule can be drawn involving those two events.

If the lift is > 1, that lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

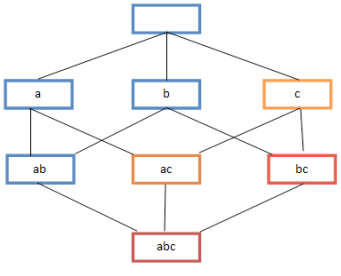
The value of lift is that it considers both the confidence of the rule and the overall data set.

Conviction

The *conviction* of a rule is defined as $\text{conv}(X \Rightarrow Y) = \frac{1 - \text{supp}(Y)}{1 - \text{conf}(X \Rightarrow Y)}$.

For example, the rule $\{\text{milk}, \text{bread}\} \Rightarrow \{\text{butter}\}$ has a conviction of $\frac{1 - 0.4}{1 - 0.5} = 1.2$, and can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions. In this example, the conviction value of 1.2 shows that the rule $\{\text{milk}, \text{bread}\} \Rightarrow \{\text{butter}\}$ would be incorrect 20% more often (1.2 times as often) if the association between X and Y was purely random chance.

Process



Frequent itemset lattice, where the color of the box indicates how many transactions contain the combination of items. Note that lower levels of the lattice can contain at most the minimum number of their parents' items; e.g. {ac} can have only at most $min(a, c)$ items. This is called the *downward-closure property*.

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. A minimum support threshold is applied to find all *frequent item-sets* in a database.
2. A minimum confidence constraint is applied to these frequent item-sets in order to form rules.

While the second step is straightforward, the first step needs more attention.

Finding all frequent item-sets in a database is difficult since it involves searching all possible item-sets (item combinations). The set of possible item-sets is the power set over I and has size $2^n - 1$ (excluding the empty set which is not a valid item-set). Although the size of the power-set grows exponentially in the number of items n in I, efficient search is possible using the *downward-closure property* of support (also called *anti-monotonicity*) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent item-set, all its super-sets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent item-sets.

History

The concept of association rules was popularised particularly due to the 1993 article of Agrawal et al., which has acquired more than 18,000 citations according to Google Scholar, as of August 2015, and is thus one of the most cited papers in the Data Mining field. However, it is possible that what is now called "association rules" is similar to what appears in the 1966 paper on GUHA, a general data mining method developed by Petr Hájek et al.

An early (circa 1989) use of minimum support and confidence to find all association rules is the Feature Based Modeling framework, which found all rules with $\mathrm{supp}(X)$ and $\mathrm{conf}(X \Rightarrow Y)$ greater than user defined constraints.

Alternative Measures of Interestingness

In addition to confidence, other measures of *interestingness* for rules have been proposed. Some popular measures are:

- All-confidence
- Collective strength
- Conviction
- Leverage
- Lift (originally called interest)

Several more measures are presented and compared by Tan et al. and by Hahsler. Looking for techniques that can model what the user has known (and using these models as interestingness measures) is currently an active research trend under the name of "Subjective Interestingness."

Statistically Sound Associations

One limitation of the standard approach to discovering associations is that by searching massive numbers of possible associations to look for collections of items that appear to be associated, there

is a large risk of finding many spurious associations. These are collections of items that co-occur with unexpected frequency in the data, but only do so by chance. For example, suppose we are considering a collection of 10,000 items and looking for rules containing two items in the left-hand-side and 1 item in the right-hand-side. There are approximately 1,000,000,000,000 such rules. If we apply a statistical test for independence with a significance level of 0.05 it means there is only a 5% chance of accepting a rule if there is no association. If we assume there are no associations, we should nonetheless expect to find 50,000,000,000 rules. Statistically sound association discovery controls this risk, in most cases reducing the risk of finding *any* spurious associations to a user-specified significance levels.

Algorithms

Many algorithms for generating association rules were presented over time.

Some well known algorithms are Apriori, Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent itemsets. Another step needs to be done after to generate rules from frequent itemsets found in a database.

Apriori Algorithm

Apriori uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support.

Eclat Algorithm

Eclat (alt. ECLAT, stands for Equivalence Class Transformation) is a depth-first search algorithm using set intersection. It is a naturally elegant algorithm suitable for both sequential as well as parallel execution with locality enhancing properties. It was first introduced by Zaki, Parthasarathy, Li and Ogihara in a series of papers written in 1997.

Mohammed Javeed Zaki, Srinivasan Parthasarathy, M. Ogihara, Wei Li: New Algorithms for Fast Discovery of Association Rules. KDD 1997.

Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li: Parallel Algorithms for Discovery of Association Rules. Data Min. Knowl. Discov. 1(4): 343-373 (1997)

FP-growth Algorithm

FP stands for frequent pattern.

In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root.

Recursive processing of this compressed version of main dataset grows large item sets directly, instead of generating candidate items and testing them against the entire database. Growth starts

from the bottom of the header table (having longest branches), by finding all instances matching given condition. New tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting sum of its children counts. Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree.

Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins.

Others

AprioriDP

AprioriDP utilizes Dynamic Programming in Frequent itemset mining. The working principle is to eliminate the candidate generation like FP-tree, but it stores support count in specialized data structure instead of tree.

Context Based Association Rule Mining Algorithm

CBPNARM is the newly developed algorithm which is developed in 2013 to mine association rules on the basis of context. It uses context variable on the basis of which the support of an itemset is changed on the basis of which the rules are finally populated to the rule set.

Node-set-based Algorithms

FIN, PrePost and PPV are three algorithms based on node sets. They use nodes in a coding FP-tree to represent itemsets, and employ a depth-first search strategy to discovery frequent itemsets using "intersection" of node sets.

GUHA Procedure ASSOC

GUHA is a general method for exploratory data analysis that has theoretical foundations in observational calculi.

The ASSOC procedure is a GUHA method which mines for generalized association rules using fast bitstrings operations. The association rules mined by this method are more general than those output by apriori, for example "items" can be connected both with conjunction and disjunctions and the relation between antecedent and consequent of the rule is not restricted to setting minimum support and confidence as in apriori: an arbitrary combination of supported interest measures can be used.

OPUS Search

OPUS is an efficient algorithm for rule discovery that, in contrast to most alternatives, does not require either monotone or anti-monotone constraints such as minimum support. Initially used to find rules for a fixed consequent it has subsequently been extended to find rules with any item as a consequent. OPUS search is the core technology in the popular Magnum Opus association discovery system.

Lore

A famous story about association rule mining is the "beer and diaper" story. A purported survey of behavior of supermarket shoppers discovered that customers (presumably young men) who buy diapers tend also to buy beer. This anecdote became popular as an example of how unexpected association rules might be found from everyday data. There are varying opinions as to how much of the story is true. Daniel Powers says:

In 1992, Thomas Blischok, manager of a retail consulting group at Teradata, and his staff prepared an analysis of 1.2 million market baskets from about 25 Osco Drug stores. Database queries were developed to identify affinities. The analysis "did discover that between 5:00 and 7:00 p.m. that consumers bought beer and diapers". Osco managers did NOT exploit the beer and diapers relationship by moving the products closer together on the shelves.

Other Types of Association Mining

Multi-Relation Association Rules: Multi-Relation Association Rules (MRAR) is a new class of association rules which in contrast to primitive, simple and even multi-relational association rules (that are usually extracted from multi-relational databases), each rule item consists of one entity but several relations. These relations indicate indirect relationship between the entities. Consider the following MRAR where the first item consists of three relations *live in*, *nearby* and *humid*: "Those who *live in* a place which is *near by* a city with *humid* climate type and also are *younger* than 20 -> their *health condition* is good". Such association rules are extractable from RDBMS data or semantic web data.

Context Based Association Rules is a form of association rule. Context Based Association Rules claims more accuracy in association rule mining by considering a hidden variable named context variable which changes the final set of association rules depending upon the value of context variables. For example the baskets orientation in market basket analysis reflects an odd pattern in the early days of month.This might be because of abnormal context i.e. salary is drawn at the start of the month

Contrast set learning is a form of associative learning. Contrast set learners use rules that differ meaningfully in their distribution across subsets.

Weighted class learning is another form of associative learning in which weight may be assigned to classes to give focus to a particular issue of concern for the consumer of the data mining results.

High-order pattern discovery facilitate the capture of high-order (polythetic) patterns or event associations that are intrinsic to complex real-world data.

K-optimal pattern discovery provides an alternative to the standard approach to association rule learning that requires that each pattern appear frequently in the data.

Approximate Frequent Itemset mining is a relaxed version of Frequent Itemset mining that allows some of the items in some of the rows to be 0.

Generalized Association Rules hierarchical taxonomy (concept hierarchy)

Quantitative Association Rules categorical and quantitative data

Interval Data Association Rules e.g. partition the age into 5-year-increment ranged

Maximal Association Rules

Sequential pattern mining discovers subsequences that are common to more than minsup sequences in a sequence database, where minsup is set by the user. A sequence is an ordered list of transactions.

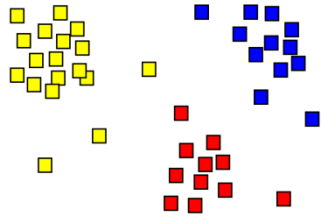
Sequential Rules discovering relationships between items while considering the time ordering. It is generally applied on a sequence database. For example, a sequential rule found in database of sequences of customer transactions can be that customers who bought a computer and CD-Roms, later bought a webcam, with a given confidence and support.

Subspace Clustering, a specific type of Clustering high-dimensional data, is in many variants also based on the downward-closure property for specific clustering models.

Warmr is shipped as part of the ACE data mining suite. It allows association rule learning for first order relational rules.

Expected float entropy minimisation simultaneously uncovers relationships between nodes (referred to as items on this page) of a system and also between the different states that the nodes can be in. The theory links associations inherent to systems such as the brain to associations present in perception.

Cluster Analysis



The result of a cluster analysis shown as the coloring of the squares into three clusters.

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical dis-

tributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

Besides the term *clustering*, there are a number of terms with similar meanings, including *automatic classification*, *numerical taxonomy*, *botryology* and *typological analysis*. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest.

Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939 and famously used by Cattell beginning in 1943 for trait theory classification in personality psychology.

Definition

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms. There is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties. Understanding these "cluster models" is key to understanding the differences between the various algorithms. Typical cluster models include:

- Connectivity models: for example, hierarchical clustering builds models based on distance connectivity.
- Centroid models: for example, the k-means algorithm represents each cluster by a single mean vector.
- Distribution models: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.
- Density models: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- Subspace models: in Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
- Group models: some algorithms do not provide a refined model for their results and just provide the grouping information.
- Graph-based models: a clique, that is, a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example, a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished as:

- hard clustering: each object belongs to a cluster or not
- soft clustering (also: fuzzy clustering): each object belongs to each cluster to a certain degree (for example, a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- strict partitioning clustering: here each object belongs to exactly one cluster
- strict partitioning clustering with outliers: objects can also belong to no cluster, and are considered outliers.
- overlapping clustering (also: alternative clustering, multi-view clustering): while usually a hard clustering, objects may belong to more than one cluster.
- hierarchical clustering: objects that belong to a child cluster also belong to the parent cluster
- subspace clustering: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap.

Algorithms

Clustering algorithms can be categorized based on their cluster model, as listed above. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized.

There is no objectively "correct" clustering algorithm, but as it was noted, "clustering is in the eye of the beholder." The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model has no chance on a data set that contains a radically different kind of model. For example, k-means cannot find non-convex clusters.

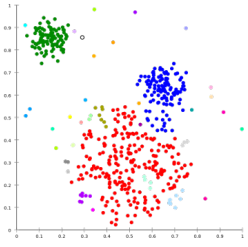
Connectivity-based Clustering (Hierarchical Clustering)

Connectivity based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

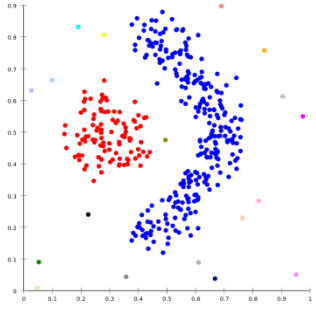
Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

These methods will not produce a unique partitioning of the data set, but a hierarchy from which the user still needs to choose appropriate clusters. They are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering). In the general case, the complexity is $\mathcal{O}(n^3)$ for agglomerative clustering and $\mathcal{O}(2^{n-1})$ for divisive clustering, which makes them too slow for large data sets. For some special cases, optimal efficient methods (of complexity $\mathcal{O}(n^2)$) are known: SLINK for single-linkage and CLINK for complete-linkage clustering. In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering.

Linkage clustering examples



Single-linkage on Gaussian data. At 35 clusters, the biggest cluster starts fragmenting into smaller parts, while before it was still connected to the second largest due to the single-link effect.



Single-linkage on density-based clusters. 20 clusters extracted, most of which contain single elements, since linkage clustering does not have a notion of "noise".

Centroid-based Clustering

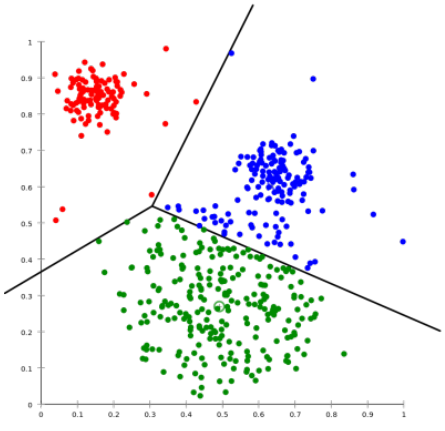
In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, *k*-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximative method is Lloyd's algorithm, often actually referred to as "*k-means algorithm*". It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of k-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (k-medoids), choosing medians (k-medians clustering), choosing the initial centers less randomly (K-means++) or allowing a fuzzy cluster assignment (Fuzzy c-means).

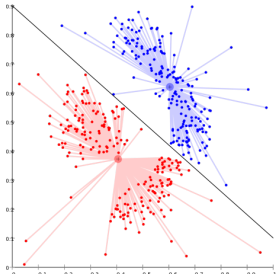
Most k-means-type algorithms require the number of clusters - k - to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders in between of clusters (which is not surprising, as the algorithm optimized cluster centers, not cluster borders).

K-means has a number of interesting theoretical properties. First, it partitions the data space into a structure known as a Voronoi diagram. Second, it is conceptually close to nearest neighbor classification, and as such is popular in machine learning. Third, it can be seen as a variation of model based classification, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model discussed below.

k-Means clustering examples



K-means separates data into Voronoi-cells, which assumes equal-sized clusters (not adequate here)



K-means cannot represent density-based clusters

Distribution-based Clustering

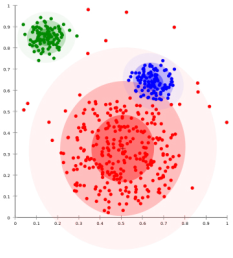
The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A convenient property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

While the theoretical foundation of these methods is excellent, they suffer from one key problem known as overfitting, unless constraints are put on the model complexity. A more complex model will usually be able to explain the data better, which makes choosing the appropriate model complexity inherently difficult.

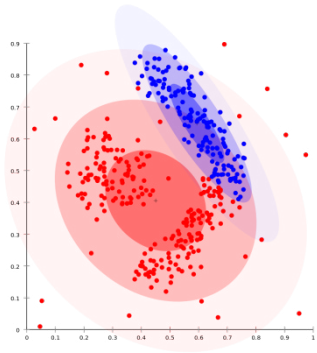
One prominent method is known as Gaussian mixture models (using the expectation-maximization algorithm). Here, the data set is usually modelled with a fixed (to avoid overfitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set. This will converge to a local optimum, so multiple runs may produce different results. In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to; for soft clusterings, this is not necessary.

Distribution-based clustering produces complex models for clusters that can capture correlation and dependence between attributes. However, these algorithms put an extra burden on the user: for many real data sets, there may be no concisely defined mathematical model (e.g. assuming Gaussian distributions is a rather strong assumption on the data).

Expectation-Maximization (EM) clustering examples



On Gaussian-distributed data, EM works well, since it uses Gaussians for modelling clusters



Density-based clusters cannot be modeled using Gaussian distributions

Density-based Clustering

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

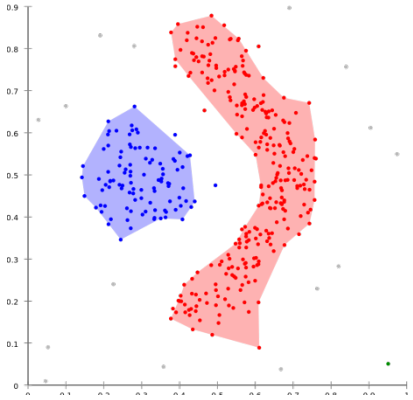
The most popular density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering, it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times. OPTICS is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ε, and produces a hierarchical result related to that of linkage clustering. DeLi-Clu, Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter entirely and offering performance improvements over OPTICS by using an R-tree index.

The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders. Moreover, they cannot detect intrinsic cluster structures which are prevalent in the majority of real life data. A variation of DBSCAN, EnDBSCAN, efficiently detects such kinds of structures. On data sets with, for example, overlapping Gaussian distributions - a common use case in artificial data - the cluster borders produced by these algorithms will often look arbitrary, because the cluster density decreases continuously. On a data set consisting of mixtures of Gaussians, these algorithms are nearly always outperformed by methods such as EM clustering that are able to precisely model this kind of data.

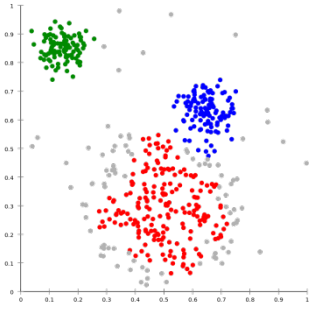
Mean-shift is a clustering approach where each object is moved to the densest area in its vicinity, based on kernel density estimation. Eventually, objects converge to local maxima of density. Similar to k-means clustering, these "density attractors" can serve as representatives for the data set,

but mean-shift can detect arbitrary-shaped clusters similar to DBSCAN. Due to the expensive iterative procedure and density estimation, mean-shift is usually slower than DBSCAN or k-Means.

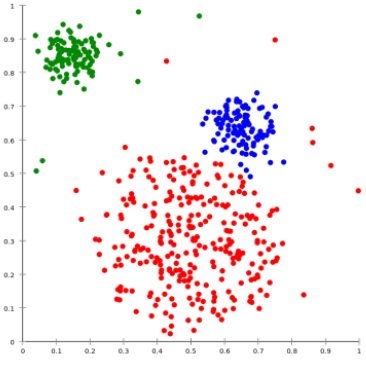
Density-based clustering examples



Density-based clustering with DBSCAN.



DBSCAN assumes clusters of similar density, and may have problems separating nearby clusters



OPTICS is a DBSCAN variant that handles different densities much better

Recent Developments

In recent years considerable effort has been put into improving the performance of existing algorithms. Among them are *CLARANS* (Ng and Han, 1994), and *BIRCH* (Zhang et al., 1996). With the recent need to process larger and larger data sets (also known as big data), the willingness to trade semantic meaning of the generated clusters for performance has been increasing. This led to the development of pre-clustering methods such as canopy clustering, which can process huge data sets efficiently, but the resulting "clusters" are merely a rough pre-partitioning of the data set to then analyze the partitions with existing slower methods such as k-means clustering. Various other approaches to clustering have been tried such as seed based clustering.

For high-dimensional data, many of the existing methods fail due to the curse of dimensionality, which renders particular distance functions problematic in high-dimensional spaces. This led to new clustering algorithms for high-dimensional data that focus on subspace clustering (where only some attributes are used, and cluster models include the relevant attributes for the cluster) and correlation clustering that also looks for arbitrary rotated ("correlated") subspace clusters that can be modeled by giving a correlation of their attributes. Examples for such clustering algorithms are CLIQUE and SUBCLU.

Ideas from density-based clustering methods (in particular the DBSCAN/OPTICS family of algorithms) have been adopted to subspace clustering (HiSC, hierarchical subspace clustering and DiSH) and correlation clustering (HiCO, hierarchical correlation clustering, 4C using "correlation connectivity" and ERiC exploring hierarchical density-based correlation clusters).

Several different clustering systems based on mutual information have been proposed. One is Marina Meilă's *variation of information* metric; another provides hierarchical clustering. Using genetic algorithms, a wide range of different fit-functions can be optimized, including mutual information. Also message passing algorithms, a recent development in Computer Science and Statistical Physics, has led to the creation of new types of clustering algorithms.

Other Methods

Basic sequential algorithmic scheme (BSAS)

Evaluation and Assessment

Evaluation of clustering results sometimes is referred to as cluster validation.

There have been several suggestions for a measure of similarity between two clusterings. Such a measure can be used to compare how well different data clustering algorithms perform on a set of data. These measures are usually tied to the type of criterion being considered in assessing the quality of a clustering method.

Internal Evaluation

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of

using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example, k-Means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.

Therefore, the internal evaluation measures are best suited to get some insight into situations where one algorithm performs better than another, but this shall not imply that one algorithm produces more valid results than another. Validity as measured by such an index depends on the claim that this kind of structure exists in the data set. An algorithm designed for some kind of models has no chance if the data set contains a radically different set of models, or if the evaluation measures a radically different criterion. For example, k-means clustering can only find convex clusters, and many evaluation indexes assume convex clusters. On a data set with non-convex clusters neither the use of k-means, nor of an evaluation criterion that assumes convexity, is sound.

The following methods can be used to assess the quality of clustering algorithms based on internal criterion:

- Davies–Bouldin index

  The Davies–Bouldin index can be calculated by the following formula:

$$DB = \frac{1}{n}\sum_{i=1}^{n} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right)$$

  where n is the number of clusters, c_x is the centroid of cluster x, σ_x is the average distance of all elements in cluster x to centroid c_x, and $d(c_i, c_j)$ is the distance between centroids c_i and c_j. Since algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies–Bouldin index, the clustering algorithm that produces a collection of clusters with the smallest Davies–Bouldin index is considered the best algorithm based on this criterion.

- Dunn index

  The Dunn index aims to identify dense and well-separated clusters. It is defined as the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. For each cluster partition, the Dunn index can be calculated by the following formula:

$$D = \frac{\min_{1 \le i < j \le n} d(i, j)}{\max_{1 \le k \le n} d'(k)},$$

  where $d(i,j)$ represents the distance between clusters i and j, and $d'(k)$ measures the intra-cluster distance of cluster k. The inter-cluster distance $d(i,j)$ between two clusters may be any number of distance measures, such as the distance between the centroids of the clusters. Similarly, the intra-cluster distance $d'(k)$ may be measured in a variety ways, such as the maximal distance between any pair of elements in cluster k. Since internal criterion

seek clusters with high intra-cluster similarity and low inter-cluster similarity, algorithms that produce clusters with high Dunn index are more desirable.

- Silhouette coefficient

  The silhouette coefficient contrasts the average distance to elements in the same cluster with the average distance to elements in other clusters. Objects with a high silhouette value are considered well clustered, objects with a low value may be outliers. This index works well with k-means clustering, and is also used to determine the optimal number of clusters.

External Evaluation

In external evaluation, clustering results are evaluated based on data that was not used for clustering, such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by (expert) humans. Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. However, it has recently been discussed whether this is adequate for real data, or only on synthetic data sets with a factual ground truth, since classes can contain internal structure, the attributes present may not allow separation of clusters or the classes may contain anomalies. Additionally, from a knowledge discovery point of view, the reproduction of known knowledge may not necessarily be the intended result. In the special scenario of constrained clustering, where meta information (such as class labels) is used already in the clustering process, the hold-out of information for evaluation purposes is non-trivial.

A number of measures are adapted from variants used to evaluate classification tasks. In place of counting the number of times a class was correctly assigned to a single data point (known as true positives), such *pair counting* metrics assess whether each pair of data points that is truly in the same cluster is predicted to be in the same cluster.

Some of the measures of quality of a cluster algorithm using external criterion include:

- Rand measure (William M. Rand 1971)

  The Rand index computes how similar the clusters (returned by the clustering algorithm) are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be computed using the following formula:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

  where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. One issue with the Rand index is that false positives and false negatives are equally weighted. This may be an undesirable characteristic for some clustering applications. The F-measure addresses this concern, as does the chance-corrected adjusted Rand index.

- F-measure

  The F-measure can be used to balance the contribution of false negatives by weighting recall through a parameter $\beta \geq 0$. Let precision and recall be defined as follows:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

  where P is the precision rate and R is the recall rate. We can calculate the F-measure by using the following formula:

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

  Notice that when $\beta = 0$, $F_0 = P$. In other words, recall has no impact on the F-measure when $\beta = 0$, and increasing β allocates an increasing amount of weight to recall in the final F-measure.

- Jaccard index

  The Jaccard index is used to quantify the similarity between two datasets. The Jaccard index takes on a value between 0 and 1. An index of 1 means that the two dataset are identical, and an index of 0 indicates that the datasets have no common elements. The Jaccard index is defined by the following formula:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}$$

  This is simply the number of unique elements common to both sets divided by the total number of unique elements in both sets.

- Fowlkes–Mallows index (E. B. Fowlkes & C. L. Mallows 1983)

  The Fowlkes-Mallows index computes the similarity between the clusters returned by the clustering algorithm and the benchmark classifications. The higher the value of the Fowlkes-Mallows index the more similar the clusters and the benchmark classifications are. It can be computed using the following formula:

$$FM = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}}$$

  where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. The FM index is the geometric mean of the precision and recall P and R, while the F-measure is their harmonic mean. Moreover, precision and recall are also known as Wallace's indices B^I and B^{II}.

- The Mutual Information is an information theoretic measure of how much information is shared between a clustering and a ground-truth classification that can detect a non-linear similarity between two clusterings. Adjusted mutual information is the corrected-for-chance variant of this that has a reduced bias for varying cluster numbers.
- Confusion matrix

  A confusion matrix can be used to quickly visualize the results of a classification (or clustering) algorithm. It shows how different a cluster is from the gold standard cluster.

Applications

Biology, computational biology and bioinformatics

Plant and animal ecology

cluster analysis is used to describe and to make spatial and temporal comparisons of communities (assemblages) of organisms in heterogeneous environments; it is also used in plant systematics to generate artificial phylogenies or clusters of organisms (individuals) at the species, genus or higher level that share a number of attributes

Transcriptomics

clustering is used to build groups of genes with related expression patterns (also known as coexpressed genes) as in HCS clustering algorithm . Often such groups contain functionally related proteins, such as enzymes for a specific pathway, or genes that are co-regulated. High throughput experiments using expressed sequence tags (ESTs) or DNA microarrays can be a powerful tool for genome annotation, a general aspect of genomics.

Sequence analysis

clustering is used to group homologous sequences into gene families. This is a very important concept in bioinformatics, and evolutionary biology in general.

High-throughput genotyping platforms

clustering algorithms are used to automatically assign genotypes.

Human genetic clustering

The similarity of genetic data is used in clustering to infer population structures.

Medicine

Medical imaging

On PET scans, cluster analysis can be used to differentiate between different types of tissue in a three-dimensional image for many different purposes.

Business and marketing

Market research

Cluster analysis is widely used in market research when working with multivariate data from surveys and test panels. Market researchers use cluster analysis to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers/potential customers, and for use in market segmentation, Product positioning, New product development and Selecting test markets.

Grouping of shopping items

Clustering can be used to group all the shopping items available on the web into a set of unique products. For example, all the items on eBay can be grouped into unique products. (eBay doesn't have the concept of a SKU)

World wide web

Social network analysis

In the study of social networks, clustering may be used to recognize communities within large groups of people.

Search result grouping

In the process of intelligent grouping of the files and websites, clustering may be used to create a more relevant set of search results compared to normal search engines like Google. There are currently a number of web based clustering tools such as Clusty.

Slippy map optimization

Flickr's map of photos and other map sites use clustering to reduce the number of markers on a map. This makes it both faster and reduces the amount of visual clutter.

Computer science

Software evolution

Clustering is useful in software evolution as it helps to reduce legacy properties in code by reforming functionality that has become dispersed. It is a form of restructuring and hence is a way of direct preventative maintenance.

Image segmentation

Clustering can be used to divide a digital image into distinct regions for border detection or object recognition.

Evolutionary algorithms

Clustering may be used to identify different niches within the population of an evolutionary algorithm so that reproductive opportunity can be distributed more evenly amongst the evolving species or subspecies.

Recommender systems

Recommender systems are designed to recommend new items based on a user's tastes. They sometimes use clustering algorithms to predict a user's preferences based on the preferences of other users in the user's cluster.

Markov chain Monte Carlo methods

Clustering is often utilized to locate and characterize extrema in the target distribution.

Anomaly detection

Anomalies/outliers are typically - be it explicitly or implicitly - defined with respect to clustering structure in data.

Social science

Crime analysis

Cluster analysis can be used to identify areas where there are greater incidences of particular types of crime. By identifying these distinct areas or "hot spots" where a similar crime has happened over a period of time, it is possible to manage law enforcement resources more effectively.

Educational data mining

Cluster analysis is for example used to identify groups of schools or students with similar properties.

Typologies

From poll data, projects such as those undertaken by the Pew Research Center use cluster analysis to discern typologies of opinions, habits, and demographics that may be useful in politics and marketing.

Others

Field robotics

Clustering algorithms are used for robotic situational awareness to track objects and detect outliers in sensor data.

Mathematical chemistry

To find structural similarity, etc., for example, 3000 chemical compounds were clustered in the space of 90 topological indices.

Climatology

To find weather regimes or preferred sea level pressure atmospheric patterns.

Petroleum geology

Cluster analysis is used to reconstruct missing bottom hole core data or missing log curves in order to evaluate reservoir properties.

Physical geography

The clustering of chemical properties in different sample locations.

Statistical Classification

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or *features*. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category.

Terminology across fields is quite varied. In statistics, where classification is often done with logistic regression or a similar procedure, the properties of observations are termed explanatory variables (or independent variables, regressors, etc.), and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the observations are often known as *instances*, the explanatory variables are termed *features* (grouped into a feature vector), and the possible categories to be predicted are *classes*. Other fields may use different terminology: e.g. in community ecology, the term "classification" normally refers to cluster analysis, i.e. a type of unsupervised learning, rather than the supervised learning described in this article.

Relation to Other Problems

Classification and clustering are examples of the more general problem of pattern recognition, which is the assignment of some sort of output value to a given input value. Other examples are

regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence; etc.

A common subclass of classification is probabilistic classification. Algorithms of this nature use statistical inference to find the best class for a given instance. Unlike other algorithms, which simply output a "best" class, probabilistic algorithms output a probability of the instance being a member of each of the possible classes. The best class is normally then selected as the one with the highest probability. However, such an algorithm has numerous advantages over non-probabilistic classifiers:

- It can output a confidence value associated with its choice (in general, a classifier that can do this is known as a *confidence-weighted classifier*).
- Correspondingly, it can *abstain* when its confidence of choosing any particular output is too low.
- Because of the probabilities which are generated, probabilistic classifiers can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of *error propagation*.

Frequentist Procedures

Early work on statistical classification was undertaken by Fisher, in the context of two-group problems, leading to Fisher's linear discriminant function as the rule for assigning a group to a new observation. This early work assumed that data-values within each of the two groups had a multivariate normal distribution. The extension of this same context to more than two-groups has also been considered with a restriction imposed that the classification rule should be linear. Later work for the multivariate normal distribution allowed the classifier to be nonlinear: several classification rules can be derived based on slight different adjustments of the Mahalanobis distance, with a new observation being assigned to the group whose centre has the lowest adjusted distance from the observation.

Bayesian Procedures

Unlike frequentist procedures, Bayesian classification procedures provide a natural way of taking into account any available information about the relative sizes of the sub-populations associated with the different groups within the overall population. Bayesian procedures tend to be computationally expensive and, in the days before Markov chain Monte Carlo computations were developed, approximations for Bayesian clustering rules were devised.

Some Bayesian procedures involve the calculation of group membership probabilities: these can be viewed as providing a more informative outcome of a data analysis than a simple attribution of a single group-label to each new observation.

Binary and Multiclass Classification

Classification can be thought of as two separate problems – binary classification and multiclass

classification. In binary classification, a better understood task, only two classes are involved, whereas multiclass classification involves assigning an object to one of several classes. Since many classification methods have been developed specifically for binary classification, multiclass classification often requires the combined use of multiple binary classifiers.

Feature Vectors

Most algorithms describe an individual instance whose category is to be predicted using a feature vector of individual, measurable properties of the instance. Each property is termed a feature, also known in statistics as an explanatory variable (or independent variable, although features may or may not be statistically independent). Features may variously be binary (e.g. "male" or "female"); categorical (e.g. "A", "B", "AB" or "O", for blood type); ordinal (e.g. "large", "medium" or "small"); integer-valued (e.g. the number of occurrences of a particular word in an email); or real-valued (e.g. a measurement of blood pressure). If the instance is an image, the feature values might correspond to the pixels of an image; if the instance is a piece of text, the feature values might be occurrence frequencies of different words. Some algorithms work only in terms of discrete data and require that real-valued or integer-valued data be *discretized* into groups (e.g. less than 5, between 5 and 10, or greater than 10)

Linear Classifiers

A large number of algorithms for classification can be phrased in terms of a linear function that assigns a score to each possible category k by combining the feature vector of an instance with a vector of weights, using a dot product. The predicted category is the one with the highest score. This type of score function is known as a linear predictor function and has the following general form:

$$score(\mathbf{X}_i, k) = \beta_k \cdot \mathbf{X}_i,$$

where X_i is the feature vector for instance i, β_k is the vector of weights corresponding to category k, and score(X_i, k) is the score associated with assigning instance i to category k. In discrete choice theory, where instances represent people and categories represent choices, the score is considered the utility associated with person i choosing category k.

Algorithms with this basic setup are known as linear classifiers. What distinguishes them is the procedure for determining (training) the optimal weights/coefficients and the way that the score is interpreted.

Examples of such algorithms are

- Logistic regression and Multinomial logistic regression
- Probit regression
- The perceptron algorithm
- Support vector machines
- Linear discriminant analysis.

Algorithms

Examples of classification algorithms include:

- Linear classifiers
    - Fisher's linear discriminant
    - Logistic regression
    - Naive Bayes classifier
    - Perceptron
- Support vector machines
    - Least squares support vector machines
- Quadratic classifiers
- Kernel estimation
    - k-nearest neighbor
- Boosting (meta-algorithm)
- Decision trees
    - Random forests
- Neural networks
- FMM Neural Networks
- Learning vector quantization

Evaluation

Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems (a phenomenon that may be explained by the no-free-lunch theorem). Various empirical tests have been performed to compare classifier performance and to find the characteristics of data that determine classifier performance. Determining a suitable classifier for a given problem is however still more an art than a science.

The measures precision and recall are popular metrics used to evaluate the quality of a classification system. More recently, receiver operating characteristic (ROC) curves have been used to evaluate the tradeoff between true- and false-positive rates of classification algorithms.

As a performance metric, the uncertainty coefficient has the advantage over simple accuracy in that it is not affected by the relative sizes of the different classes. Further, it will not penalize an algorithm for simply *rearranging* the classes.

Application Domains

Classification has many applications. In some of these it is employed as a data mining procedure,

while in others more detailed statistical modeling is undertaken.

- Computer vision
    - Medical imaging and medical image analysis
    - Optical character recognition
    - Video tracking
- Drug discovery and development
    - Toxicogenomics
    - Quantitative structure-activity relationship
- Geostatistics
- Speech recognition
- Handwriting recognition
- Biometric identification
- Biological classification
- Statistical natural language processing
- Document classification
- Internet search engines
- Credit scoring
- Pattern recognition
- Micro-array classification

Regression Analysis

In statistical modeling, regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed. Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables – that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, the estimation target is a function of the independent variables called the regression function. In regression analysis, it is also of interest to characterize the variation of the dependent variable

around the regression function which can be described by a probability distribution. A related but distinct approach is necessary condition analysis (NCA), which estimates the maximum (rather than average) value of the dependent variable for a given value of the independent variable (ceiling line rather than central line) in order to identify what value of the independent variable is necessary but not sufficient for a given value of the dependent variable.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable; for example, correlation does not imply causation.

Many techniques for carrying out regression analysis have been developed. Familiar methods such as linear regression and ordinary least squares regression are parametric, in that the regression function is defined in terms of a finite number of unknown parameters that are estimated from the data. Nonparametric regression refers to techniques that allow the regression function to lie in a specified set of functions, which may be infinite-dimensional.

The performance of regression analysis methods in practice depends on the form of the data generating process, and how it relates to the regression approach being used. Since the true form of the data-generating process is generally not known, regression analysis often depends to some extent on making assumptions about this process. These assumptions are sometimes testable if a sufficient quantity of data is available. Regression models for prediction are often useful even when the assumptions are moderately violated, although they may not perform optimally. However, in many applications, especially with small effects or questions of causality based on observational data, regression methods can give misleading results.

In a narrower sense, regression may refer specifically to the estimation of continuous response variables, as opposed to the discrete response variables used in classification. The case of a continuous output variable may be more specifically referred to as metric regression to distinguish it from related problems.

History

The earliest form of regression was the method of least squares, which was published by Legendre in 1805, and by Gauss in 1809. Legendre and Gauss both applied the method to the problem of determining, from astronomical observations, the orbits of bodies about the Sun (mostly comets, but also later the then newly discovered minor planets). Gauss published a further development of the theory of least squares in 1821, including a version of the Gauss–Markov theorem.

The term “regression” was coined by Francis Galton in the nineteenth century to describe a biological phenomenon. The phenomenon was that the heights of descendants of tall ancestors tend to regress down towards a normal average (a phenomenon also known as regression toward the mean). For Galton, regression had only this biological meaning, but his work was later extended by Udny Yule and Karl Pearson to a more general statistical context. In the work of Yule and Pearson, the joint distribution of the response and explanatory variables is assumed to be Gaussian. This

assumption was weakened by R.A. Fisher in his works of 1922 and 1925. Fisher assumed that the conditional distribution of the response variable is Gaussian, but the joint distribution need not be. In this respect, Fisher's assumption is closer to Gauss's formulation of 1821.

In the 1950s and 1960s, economists used electromechanical desk calculators to calculate regressions. Before 1970, it sometimes took up to 24 hours to receive the result from one regression.

Regression methods continue to be an area of active research. In recent decades, new methods have been developed for robust regression, regression involving correlated responses such as time series and growth curves, regression in which the predictor (independent variable) or response variables are curves, images, graphs, or other complex data objects, regression methods accommodating various types of missing data, nonparametric regression, Bayesian methods for regression, regression in which the predictor variables are measured with error, regression with more predictor variables than observations, and causal inference with regression.

Regression Models

Regression models involve the following variables:

- The unknown parameters, denoted as β, which may represent a scalar or a vector.
- The independent variables, X.
- The dependent variable, *Y*.

In various fields of application, different terminologies are used in place of dependent and independent variables.

A regression model relates *Y* to a function of X and β.

$$Y \approx f(X, \beta)$$

The approximation is usually formalized as $E(Y \mid X) = f(X, \beta)$. To carry out regression analysis, the form of the function *f* must be specified. Sometimes the form of this function is based on knowledge about the relationship between *Y* and X that does not rely on the data. If no such knowledge is available, a flexible or convenient form for *f* is chosen.

Assume now that the vector of unknown parameters β is of length *k*. In order to perform a regression analysis the user must provide information about the dependent variable *Y*:

- If *N* data points of the form (*Y*, X) are observed, where $N < k$, most classical approaches to regression analysis cannot be performed: since the system of equations defining the regression model is underdetermined, there are not enough data to recover β.
- If exactly $N = k$ data points are observed, and the function *f* is linear, the equations $Y = f(X, \beta)$ can be solved exactly rather than approximately. This reduces to solving a set of *N* equations with *N* unknowns (the elements of β), which has a unique solution as long as the X are linearly independent. If *f* is nonlinear, a solution may not exist, or many solutions may exist.
- The most common situation is where $N > k$ data points are observed. In this case, there is

enough information in the data to estimate a unique value for β that best fits the data in some sense, and the regression model when applied to the data can be viewed as an over-determined system in β.

In the last case, the regression analysis provides the tools for:

1. Finding a solution for unknown parameters β that will, for example, minimize the distance between the measured and predicted values of the dependent variable *Y* (also known as method of least squares).
2. Under certain statistical assumptions, the regression analysis uses the surplus of information to provide statistical information about the unknown parameters β and predicted values of the dependent variable *Y*.

Necessary Number of Independent Measurements

Consider a regression model which has three unknown parameters, β_0, β_1, and β_2. Suppose an experimenter performs 10 measurements all at exactly the same value of independent variable vector X (which contains the independent variables X_1, X_2, and X_3). In this case, regression analysis fails to give a unique set of estimated values for the three unknown parameters; the experimenter did not provide enough information. The best one can do is to estimate the average value and the standard deviation of the dependent variable *Y*. Similarly, measuring at two different values of X would give enough data for a regression with two unknowns, but not for three or more unknowns.

If the experimenter had performed measurements at three different values of the independent variable vector X, then regression analysis would provide a unique set of estimates for the three unknown parameters in β.

In the case of general linear regression, the above statement is equivalent to the requirement that the matrix X^TX is invertible.

Statistical Assumptions

When the number of measurements, *N*, is larger than the number of unknown parameters, *k*, and the measurement errors ε_i are normally distributed then *the excess of information* contained in $(N - k)$ measurements is used to make statistical predictions about the unknown parameters. This excess of information is referred to as the degrees of freedom of the regression.

Underlying Assumptions

Classical assumptions for regression analysis include:

- The sample is representative of the population for the inference prediction.
- The error is a random variable with a mean of zero conditional on the explanatory variables.
- The independent variables are measured with no error. (Note: If this is not so, modeling may be done instead using errors-in-variables model techniques).

- The independent variables (predictors) are linearly independent, i.e. it is not possible to express any predictor as a linear combination of the others.
- The errors are uncorrelated, that is, the variance–covariance matrix of the errors is diagonal and each non-zero element is the variance of the error.
- The variance of the error is constant across observations (homoscedasticity). If not, weighted least squares or other methods might instead be used.

These are sufficient conditions for the least-squares estimator to possess desirable properties; in particular, these assumptions imply that the parameter estimates will be unbiased, consistent, and efficient in the class of linear unbiased estimators. It is important to note that actual data rarely satisfies the assumptions. That is, the method is used even though the assumptions are not true. Variation from the assumptions can sometimes be used as a measure of how far the model is from being useful. Many of these assumptions may be relaxed in more advanced treatments. Reports of statistical analyses usually include analyses of tests on the sample data and methodology for the fit and usefulness of the model.

Assumptions include the geometrical support of the variables. Independent and dependent variables often refer to values measured at point locations. There may be spatial trends and spatial autocorrelation in the variables that violate statistical assumptions of regression. Geographic weighted regression is one technique to deal with such data. Also, variables may include values aggregated by areas. With aggregated data the modifiable areal unit problem can cause extreme variation in regression parameters. When analyzing data aggregated by political boundaries, postal codes or census areas results may be very distinct with a different choice of units.

Linear Regression

In linear regression, the model specification is that the dependent variable, y_i is a linear combination of the *parameters* (but need not be linear in the *independent variables*). For example, in simple linear regression for modeling n data points there is one independent variable: x_i, and two parameters, β_0 and β_1:

straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n.$

In multiple linear regression, there are several independent variables or functions of independent variables.

Adding a term in x_i^2 to the preceding regression gives:

parabola: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i, \ i = 1, \ldots, n.$

This is still linear regression; although the expression on the right hand side is quadratic in the independent variable x_i, it is linear in the parameters β_0, β_1 and β_2.

In both cases, ε_i is an error term and the subscript i indexes a particular observation.

Returning our attention to the straight line case: Given a random sample from the population, we estimate the population parameters and obtain the sample linear regression model:

$$\widehat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

The residual, $e_i = y_i - \hat{y}_i$, is the difference between the value of the dependent variable predicted by the model, $\widehat{y}_i$,, and the true value of the dependent variable, y_i. One method of estimation is ordinary least squares. This method obtains parameter estimates that minimize the sum of squared residuals, SSE, also sometimes denoted RSS:

$$SSE = \sum_{i=1}^{n} e_i^2.$$

Minimization of this function results in a set of normal equations, a set of simultaneous linear equations in the parameters, which are solved to yield the parameter estimators, $\widehat{\beta}_0$, $\widehat{\beta}_1$

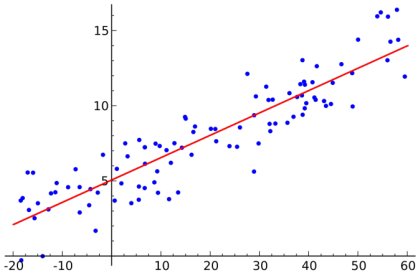


Illustration of linear regression on a data set.

In the case of simple regression, the formulas for the least squares estimates are

$$\widehat{\beta}_1 = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sum (x_i - \overline{x})^2} \text{ and } \widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}$$

where $\overline{x}$ is the mean (average) of the x values and $\overline{y}$ is the mean of the y values.

Under the assumption that the population error term has a constant variance, the estimate of that variance is given by:

$$\hat{\sigma}_\varepsilon^2 = \frac{SSE}{n-2}.$$

This is called the mean square error (MSE) of the regression. The denominator is the sample size reduced by the number of model parameters estimated from the same data, (n-p) for p regressors or (n-p-1) if an intercept is used. In this case, p=1 so the denominator is n-2.

The standard errors of the parameter estimates are given by

$$\hat{\sigma}_{\beta_0} = \hat{\sigma}_\varepsilon \sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum (x_i - \overline{x})^2}}$$

$$\hat{\sigma}_{\beta_1} = \hat{\sigma}_\varepsilon \sqrt{\frac{1}{\sum (x_i - \overline{x})^2}}.$$

Under the further assumption that the population error term is normally distributed, the researcher can use these estimated standard errors to create confidence intervals and conduct hypothesis tests about the population parameters.

General Linear Model

In the more general multiple regression model, there are p independent variables:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i,$$

where x_{ij} is the i^{th} observation on the j^{th} independent variable. If the first independent variable takes the value 1 for all i, $x_{i1} = 1$, then β_1 is called the regression intercept.

The least squares parameter estimates are obtained from p normal equations. The residual can be written as

$$\varepsilon_i = y_i - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_p x_{ip}.$$

The normal equations are

$$\sum_{i=1}^{n} \sum_{k=1}^{p} X_{ij} X_{ik} \hat{\beta}_k = \sum_{i=1}^{n} X_{ij} y_i, \; j = 1, \ldots, p.$$

In matrix notation, the normal equations are written as

$$(X^\top X)\hat{\beta} = X^\top Y,$$

where the ij element of X is x_{ij}, the i element of the column vector Y is y_i, and the j element of $\hat{\beta}$ is $\hat{\beta}_j$. Thus X is $n \times p$, Y is $n \times 1$, and $\hat{\beta}$ is $p \times 1$. The solution is

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y.$$

Diagnostics

Once a regression model has been constructed, it may be important to confirm the goodness of fit of the model and the statistical significance of the estimated parameters. Commonly used checks of goodness of fit include the R-squared, analyses of the pattern of residuals and hypothesis testing. Statistical significance can be checked by an F-test of the overall fit, followed by t-tests of individual parameters.

Interpretations of these diagnostic tests rest heavily on the model assumptions. Although examination of the residuals can be used to invalidate a model, the results of a t-test or F-test are sometimes more difficult to interpret if the model's assumptions are violated. For example, if the error term does not have a normal distribution, in small samples the estimated parameters will not follow normal distributions and complicate inference. With relatively large samples, however, a central limit theorem can be invoked such that hypothesis testing may proceed using asymptotic approximations.

"Limited Dependent" Variables

The phrase "limited dependent" is used in econometric statistics for categorical and constrained variables.

The response variable may be non-continuous ("limited" to lie on some subset of the real line). For binary (zero or one) variables, if analysis proceeds with least-squares linear regression, the model is called the linear probability model. Nonlinear models for binary dependent variables include the probit and logit model. The multivariate probit model is a standard method of estimating a joint relationship between several binary dependent variables and some independent variables. For categorical variables with more than two values there is the multinomial logit. For ordinal variables with more than two values, there are the ordered logit and ordered probit models. Censored regression models may be used when the dependent variable is only sometimes observed, and Heckman correction type models may be used when the sample is not randomly selected from the population of interest. An alternative to such procedures is linear regression based on polychoric correlation (or polyserial correlations) between the categorical variables. Such procedures differ in the assumptions made about the distribution of the variables in the population. If the variable is positive with low values and represents the repetition of the occurrence of an event, then count models like the Poisson regression or the negative binomial model may be used instead.

Interpolation and Extrapolation

Regression models predict a value of the Y variable given known values of the X variables. Prediction *within* the range of values in the dataset used for model-fitting is known informally as interpolation. Prediction *outside* this range of the data is known as extrapolation. Performing extrapolation relies strongly on the regression assumptions. The further the extrapolation goes outside the data, the more room there is for the model to fail due to differences between the assumptions and the sample data or the true values.

It is generally advised that when performing extrapolation, one should accompany the estimated value of the dependent variable with a prediction interval that represents the uncertainty. Such intervals tend to expand rapidly as the values of the independent variable(s) moved outside the range covered by the observed data.

For such reasons and others, some tend to say that it might be unwise to undertake extrapolation.

However, this does not cover the full set of modelling errors that may be being made: in particular, the assumption of a particular form for the relation between Y and X. A properly conducted regression analysis will include an assessment of how well the assumed form is matched by the observed data, but it can only do so within the range of values of the independent variables actually available. This means that any extrapolation is particularly reliant on the assumptions being made about the structural form of the regression relationship. Best-practice advice here is that a linear-in-variables and linear-in-parameters relationship should not be chosen simply for computational convenience, but that all available knowledge should be deployed in constructing a regression model. If this knowledge includes the fact that the dependent variable cannot go outside a certain range of values, this can be made use of in selecting the model – even if the observed dataset has no values particularly near such bounds. The implications of this step of choosing an appropriate functional

form for the regression can be great when extrapolation is considered. At a minimum, it can ensure that any extrapolation arising from a fitted model is "realistic" (or in accord with what is known).

Nonlinear Regression

When the model function is not linear in the parameters, the sum of squares must be minimized by an iterative procedure. This introduces many complications which are summarized in Differences between linear and non-linear least squares

Power and Sample Size Calculations

There are no generally agreed methods for relating the number of observations versus the number of independent variables in the model. One rule of thumb suggested by Good and Hardin is $N = m^n$, , where N is the sample size, n is the number of independent variables and m is the number of observations needed to reach the desired precision if the model had only one independent variable. For example, a researcher is building a linear regression model using a dataset that contains 1000 patients (N). If the researcher decides that five observations are needed to precisely define a straight line (m), then the maximum number of independent variables the model can support is 4, because

$$\frac{\log 1000}{\log 5} = 4.29.$$

Other Methods

Although the parameters of a regression model are usually estimated using the method of least squares, other methods which have been used include:

- Bayesian methods, e.g. Bayesian linear regression
- Percentage regression, for situations where reducing *percentage* errors is deemed more appropriate.
- Least absolute deviations, which is more robust in the presence of outliers, leading to quantile regression
- Nonparametric regression, requires a large number of observations and is computationally intensive
- Distance metric learning, which is learned by the search of a meaningful distance metric in a given input space.

Software

All major statistical software packages perform least squares regression analysis and inference. Simple linear regression and multiple regression using least squares can be done in some spreadsheet applications and on some calculators. While many statistical software packages can perform various types of nonparametric and robust regression, these methods are less standardized; different software packages implement different methods, and a method with a given name may be

implemented differently in different packages. Specialized regression software has been developed for use in fields such as survey analysis and neuroimaging.

Automatic Summarization

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the *information* of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a *representative summary* or *abstract* of the entire document, by finding the most *informative* sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.

Generally, there are two approaches to automatic summarization: *extraction* and *abstraction*. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Research into abstractive methods is an increasingly important and active research area, however due to complexity constraints, research to date has focused primarily on extractive methods. In some application domains, extractive summarization makes more sense. Examples of these include image collection summarization and video summarization.

Extraction-based Summarization

In this summarization task, the automatic system extracts objects from the entire collection, without modifying the objects themselves. Examples of this include keyphrase extraction, where the goal is to select individual words or phrases to “tag” a document, and document summarization, where the goal is to select whole sentences (without modifying them) to create a short paragraph summary. Similarly, in image collection summarization, the system extracts images from the collection without modifying the images themselves.

Abstraction-based Summarization

Extraction techniques merely copy the information deemed most important by the system to the

summary (for example, key clauses, sentences or paragraphs), while abstraction involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require use of natural language generation technology, which itself is a growing field.

While some work has been done in abstractive summarization (creating an abstract synopsis like that of a human), the majority of summarization systems are extractive (selecting a subset of sentences to place in a summary).

Aided Summarization

Machine learning techniques from closely related fields such as information retrieval or text mining have been successfully adapted to help automatic summarization.

Apart from Fully Automated Summarizers (FAS), there are systems that aid users with the task of summarization (MAHS = Machine Aided Human Summarization), for example by highlighting candidate passages to be included in the summary, and there are systems that depend on post-processing by a human (HAMS = Human Aided Machine Summarization).

Applications and Systems for Summarization

There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is *generic summarization*, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is *query relevant summarization*, sometimes called *query-based summarization*, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images. A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured.

At a very high level, summarization algorithms try to find subsets of objects (like set of sentences, or a set of images), which cover information of the entire set. This is also called the *core-set*. These

algorithms model notions like diversity, coverage, information and representativeness of the summary. Query based summarization techniques, additionally model for relevance of the summary with the query. Some techniques and algorithms which naturally model summarization problems are TextRank and PageRank, Submodular set function, Determinantal point process, maximal marginal relevance (MMR) etc.

Keyphrase Extraction

The task is the following. You are given a piece of text, such as a journal article, and you must produce a list of keywords or key[phrase]s that capture the primary topics discussed in the text. In the case of research articles, many authors provide manually assigned keywords, but most text lacks pre-existing keyphrases. For example, news articles rarely have keyphrases attached, but it would be useful to be able to automatically do so for a number of applications discussed below. Consider the example text from a news article:

> "The Army Corps of Engineers, rushing to meet President Bush's promise to protect New Orleans by the start of the 2006 hurricane season, installed defective flood-control pumps last year despite warnings from its own expert that the equipment would fail during a storm, according to documents obtained by The Associated Press".

A keyphrase extractor might select "Army Corps of Engineers", "President Bush", "New Orleans", and "defective flood-control pumps" as keyphrases. These are pulled directly from the text. In contrast, an abstractive keyphrase system would somehow internalize the content and generate keyphrases that do not appear in the text, but more closely resemble what a human might produce, such as "political negligence" or "inadequate protection from floods". Abstraction requires a deep understanding of the text, which makes it difficult for a computer system. Keyphrases have many applications. They can enable document browsing by providing a short summary, improve information retrieval (if documents have keyphrases assigned, a user could search by keyphrase to produce more reliable hits than a full-text search), and be employed in generating index entries for a large text corpus.

Depending on the different literature and the definition of key terms, words or phrases, highly related theme is certainly the Keyword extraction.

Supervised Learning Approaches

Beginning with the work of Turney, many researchers have approached keyphrase extraction as a supervised machine learning problem. Given a document, we construct an example for each unigram, bigram, and trigram found in the text (though other text units are also possible, as discussed below). We then compute various features describing each example (e.g., does the phrase begin with an upper-case letter?). We assume there are known keyphrases available for a set of training documents. Using the known keyphrases, we can assign positive or negative labels to the examples. Then we learn a classifier that can discriminate between positive and negative examples as a function of the features. Some classifiers make a binary classification for a test example, while others assign a probability of being a keyphrase. For instance, in the above text, we might learn a rule that says phrases with initial capital letters are likely to be keyphrases. After training a learner, we can select keyphrases for test documents in the following manner. We apply the same exam-

ple-generation strategy to the test documents, then run each example through the learner. We can determine the keyphrases by looking at binary classification decisions or probabilities returned from our learned model. If probabilities are given, a threshold is used to select the keyphrases. Keyphrase extractors are generally evaluated using precision and recall. Precision measures how many of the proposed keyphrases are actually correct. Recall measures how many of the true keyphrases your system proposed. The two measures can be combined in an F-score, which is the harmonic mean of the two ($F = 2PR/(P + R)$). Matches between the proposed keyphrases and the known keyphrases can be checked after stemming or applying some other text normalization.

Designing a supervised keyphrase extraction system involves deciding on several choices (some of these apply to unsupervised, too). The first choice is exactly how to generate examples. Turney and others have used all possible unigrams, bigrams, and trigrams without intervening punctuation and after removing stopwords. Hulth showed that you can get some improvement by selecting examples to be sequences of tokens that match certain patterns of part-of-speech tags. Ideally, the mechanism for generating examples produces all the known labeled keyphrases as candidates, though this is often not the case. For example, if we use only unigrams, bigrams, and trigrams, then we will never be able to extract a known keyphrase containing four words. Thus, recall may suffer. However, generating too many examples can also lead to low precision.

We also need to create features that describe the examples and are informative enough to allow a learning algorithm to discriminate keyphrases from non- keyphrases. Typically features involve various term frequencies (how many times a phrase appears in the current text or in a larger corpus), the length of the example, relative position of the first occurrence, various boolean syntactic features (e.g., contains all caps), etc. The Turney paper used about 12 such features. Hulth uses a reduced set of features, which were found most successful in the KEA (Keyphrase Extraction Algorithm) work derived from Turney's seminal paper.

In the end, the system will need to return a list of keyphrases for a test document, so we need to have a way to limit the number. Ensemble methods (i.e., using votes from several classifiers) have been used to produce numeric scores that can be thresholded to provide a user-provided number of keyphrases. This is the technique used by Turney with C4.5 decision trees. Hulth used a single binary classifier so the learning algorithm implicitly determines the appropriate number.

Once examples and features are created, we need a way to learn to predict keyphrases. Virtually any supervised learning algorithm could be used, such as decision trees, Naive Bayes, and rule induction. In the case of Turney's GenEx algorithm, a genetic algorithm is used to learn parameters for a domain-specific keyphrase extraction algorithm. The extractor follows a series of heuristics to identify keyphrases. The genetic algorithm optimizes parameters for these heuristics with respect to performance on training documents with known key phrases.

Unsupervised Approach: Textrank

Another keyphrase extraction algorithm is TextRank. While supervised methods have some nice properties, like being able to produce interpretable rules for what features characterize a keyphrase, they also require a large amount of training data. Many documents with known keyphrases are needed. Furthermore, training on a specific domain tends to customize the extraction process to that domain, so the resulting classifier is not necessarily portable, as some of Turney's results

demonstrate. Unsupervised keyphrase extraction removes the need for training data. It approaches the problem from a different angle. Instead of trying to learn explicit features that characterize keyphrases, the TextRank algorithm exploits the structure of the text itself to determine keyphrases that appear "central" to the text in the same way that PageRank selects important Web pages. Recall this is based on the notion of "prestige" or "recommendation" from social networks. In this way, TextRank does not rely on any previous training data at all, but rather can be run on any arbitrary piece of text, and it can produce output simply based on the text's intrinsic properties. Thus the algorithm is easily portable to new domains and languages.

TextRank is a general purpose graph-based ranking algorithm for NLP. Essentially, it runs PageRank on a graph specially designed for a particular NLP task. For keyphrase extraction, it builds a graph using some set of text units as vertices. Edges are based on some measure of semantic or lexical similarity between the text unit vertices. Unlike PageRank, the edges are typically undirected and can be weighted to reflect a degree of similarity. Once the graph is constructed, it is used to form a stochastic matrix, combined with a damping factor (as in the "random surfer model"), and the ranking over vertices is obtained by finding the eigenvector corresponding to eigenvalue 1 (i.e., the stationary distribution of the random walk on the graph).

The vertices should correspond to what we want to rank. Potentially, we could do something similar to the supervised methods and create a vertex for each unigram, bigram, trigram, etc. However, to keep the graph small, the authors decide to rank individual unigrams in a first step, and then include a second step that merges highly ranked adjacent unigrams to form multi-word phrases. This has a nice side effect of allowing us to produce keyphrases of arbitrary length. For example, if we rank unigrams and find that "advanced", "natural", "language", and "processing" all get high ranks, then we would look at the original text and see that these words appear consecutively and create a final keyphrase using all four together. Note that the unigrams placed in the graph can be filtered by part of speech. The authors found that adjectives and nouns were the best to include. Thus, some linguistic knowledge comes into play in this step.

Edges are created based on word co-occurrence in this application of TextRank. Two vertices are connected by an edge if the unigrams appear within a window of size N in the original text. N is typically around 2–10. Thus, "natural" and "language" might be linked in a text about NLP. "Natural" and "processing" would also be linked because they would both appear in the same string of N words. These edges build on the notion of "text cohesion" and the idea that words that appear near each other are likely related in a meaningful way and "recommend" each other to the reader.

Since this method simply ranks the individual vertices, we need a way to threshold or produce a limited number of keyphrases. The technique chosen is to set a count T to be a user-specified fraction of the total number of vertices in the graph. Then the top T vertices/unigrams are selected based on their stationary probabilities. A post- processing step is then applied to merge adjacent instances of these T unigrams. As a result, potentially more or less than T final keyphrases will be produced, but the number should be roughly proportional to the length of the original text.

It is not initially clear why applying PageRank to a co-occurrence graph would produce useful keyphrases. One way to think about it is the following. A word that appears multiple times throughout a text may have many different co-occurring neighbors. For example, in a text about machine learning, the unigram "learning" might co-occur with "machine", "supervised", "un-supervised",

and “semi-supervised” in four different sentences. Thus, the “learning” vertex would be a central “hub” that connects to these other modifying words. Running PageRank/TextRank on the graph is likely to rank “learning” highly. Similarly, if the text contains the phrase “supervised classification”, then there would be an edge between “supervised” and “classification”. If “classification” appears several other places and thus has many neighbors, its importance would contribute to the importance of “supervised”. If it ends up with a high rank, it will be selected as one of the top T unigrams, along with “learning” and probably “classification”. In the final post-processing step, we would then end up with keyphrases “supervised learning” and “supervised classification”.

In short, the co-occurrence graph will contain densely connected regions for terms that appear often and in different contexts. A random walk on this graph will have a stationary distribution that assigns large probabilities to the terms in the centers of the clusters. This is similar to densely connected Web pages getting ranked highly by PageRank. This approach has also been used in document summarization, considered below.

Document Summarization

Like keyphrase extraction, document summarization aims to identify the essence of a text. The only real difference is that now we are dealing with larger text units—whole sentences instead of words and phrases.

Before getting into the details of some summarization methods, we will mention how summarization systems are typically evaluated. The most common way is using the so-called ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measure. This is a recall-based measure that determines how well a system-generated summary covers the content present in one or more human-generated model summaries known as references. It is recall-based to encourage systems to include all the important topics in the text. Recall can be computed with respect to unigram, bigram, trigram, or 4-gram matching. For example, ROUGE-1 is computed as division of count of unigrams in reference that appear in system and count of unigrams in reference summary.

If there are multiple references, the ROUGE-1 scores are averaged. Because ROUGE is based only on content overlap, it can determine if the same general concepts are discussed between an automatic summary and a reference summary, but it cannot determine if the result is coherent or the sentences flow together in a sensible manner. High-order n-gram ROUGE measures try to judge fluency to some degree. Note that ROUGE is similar to the BLEU measure for machine translation, but BLEU is precision- based, because translation systems favor accuracy.

A promising line in document summarization is adaptive document/text summarization. The idea of adaptive summarization involves preliminary recognition of document/text genre and subsequent application of summarization algorithms optimized for this genre. First summarizes that perform adaptive summarization have been created.

Supervised Learning Approaches

Supervised text summarization is very much like supervised keyphrase extraction. Basically, if you have a collection of documents and human-generated summaries for them, you can learn features of sentences that make them good candidates for inclusion in the summary. Features might

include the position in the document (i.e., the first few sentences are probably important), the number of words in the sentence, etc. The main difficulty in supervised extractive summarization is that the known summaries must be manually created by extracting sentences so the sentences in an original training document can be labeled as "in summary" or "not in summary". This is not typically how people create summaries, so simply using journal abstracts or existing summaries is usually not sufficient. The sentences in these summaries do not necessarily match up with sentences in the original text, so it would be difficult to assign labels to examples for training. Note, however, that these natural summaries can still be used for evaluation purposes, since ROUGE-1 only cares about unigrams.

Maximum Entropy-based Summarization

During the DUC 2001 and 2002 evaluation workshops, TNO developed a sentence extraction system for multi-document summarization in the news domain. The system was based on a hybrid system using a naive Bayes classifier and statistical language models for modeling salience. Although the system exhibited good results, the researchers wanted to explore the effectiveness of a maximum entropy (ME) classifier for the meeting summarization task, as ME is known to be robust against feature dependencies. Maximum entropy has also been applied successfully for summarization in the broadcast news domain.

TextRank and LexRank

The unsupervised approach to summarization is also quite similar in spirit to unsupervised keyphrase extraction and gets around the issue of costly training data. Some unsupervised summarization approaches are based on finding a "centroid" sentence, which is the mean word vector of all the sentences in the document. Then the sentences can be ranked with regard to their similarity to this centroid sentence.

A more principled way to estimate sentence importance is using random walks and eigenvector centrality. LexRank is an algorithm essentially identical to TextRank, and both use this approach for document summarization. The two methods were developed by different groups at the same time, and LexRank simply focused on summarization, but could just as easily be used for keyphrase extraction or any other NLP ranking task.

In both LexRank and TextRank, a graph is constructed by creating a vertex for each sentence in the document.

The edges between sentences are based on some form of semantic similarity or content overlap. While LexRank uses cosine similarity of TF-IDF vectors, TextRank uses a very similar measure based on the number of words two sentences have in common (normalized by the sentences' lengths). The LexRank paper explored using unweighted edges after applying a threshold to the cosine values, but also experimented with using edges with weights equal to the similarity score. TextRank uses continuous similarity scores as weights.

In both algorithms, the sentences are ranked by applying PageRank to the resulting graph. A summary is formed by combining the top ranking sentences, using a threshold or length cutoff to limit the size of the summary.

It is worth noting that TextRank was applied to summarization exactly as described here, while LexRank was used as part of a larger summarization system (MEAD) that combines the LexRank score (stationary probability) with other features like sentence position and length using a linear combination with either user-specified or automatically tuned weights. In this case, some training documents might be needed, though the TextRank results show the additional features are not absolutely necessary.

Another important distinction is that TextRank was used for single document summarization, while LexRank has been applied to multi-document summarization. The task remains the same in both cases—only the number of sentences to choose from has grown. However, when summarizing multiple documents, there is a greater risk of selecting duplicate or highly redundant sentences to place in the same summary. Imagine you have a cluster of news articles on a particular event, and you want to produce one summary. Each article is likely to have many similar sentences, and you would only want to include distinct ideas in the summary. To address this issue, LexRank applies a heuristic post-processing step that builds up a summary by adding sentences in rank order, but discards any sentences that are too similar to ones already placed in the summary. The method used is called Cross-Sentence Information Subsumption (CSIS).

These methods work based on the idea that sentences "recommend" other similar sentences to the reader. Thus, if one sentence is very similar to many others, it will likely be a sentence of great importance. The importance of this sentence also stems from the importance of the sentences "recommending" it. Thus, to get ranked highly and placed in a summary, a sentence must be similar to many sentences that are in turn also similar to many other sentences. This makes intuitive sense and allows the algorithms to be applied to any arbitrary new text. The methods are domain-independent and easily portable. One could imagine the features indicating important sentences in the news domain might vary considerably from the biomedical domain. However, the unsupervised "recommendation"-based approach applies to any domain.

Multi-document Summarization

Multi-document summarization is an automatic procedure aimed at extraction of information from multiple texts written about the same topic. Resulting summary report allows individual users, such as professional information consumers, to quickly familiarize themselves with information contained in a large cluster of documents. In such a way, multi-document summarization systems are complementing the news aggregators performing the next step down the road of coping with information overload. Multi-document summarization may also be done in response to a question.

Multi-document summarization creates information reports that are both concise and comprehensive. With different opinions being put together and outlined, every topic is described from multiple perspectives within a single document. While the goal of a brief summary is to simplify information search and cut the time by pointing to the most relevant source documents, comprehensive multi-document summary should itself contain the required information, hence limiting the need for accessing original files to cases when refinement is required. Automatic summaries present information extracted from multiple sources algorithmically, without any editorial touch or subjective human intervention, thus making it completely unbiased.

Incorporating Diversity

Multi-document extractive summarization faces a problem of potential redundancy. Ideally, we would like to extract sentences that are both "central" (i.e., contain the main ideas) and "diverse" (i.e., they differ from one another). LexRank deals with diversity as a heuristic final stage using CSIS, and other systems have used similar methods, such as Maximal Marginal Relevance (MMR), in trying to eliminate redundancy in information retrieval results. There is a general purpose graph-based ranking algorithm like Page/Lex/TextRank that handles both "centrality" and "diversity" in a unified mathematical framework based on absorbing Markov chain random walks. (An absorbing random walk is like a standard random walk, except some states are now absorbing states that act as "black holes" that cause the walk to end abruptly at that state.) The algorithm is called GRASSHOPPER. In addition to explicitly promoting diversity during the ranking process, GRASSHOPPER incorporates a prior ranking (based on sentence position in the case of summarization).

The state of the art results for multi-document summarization, however, are obtained using mixtures of submodular functions. These methods have achieved the state of the art results for Document Summarization Corpora, DUC 04 - 07. Similar results were also achieved with the use of determinantal point processes (which are a special case of submodular functions) for DUC-04.

Submodular Functions as Generic Tools for Summarization

The idea of a Submodular set function has recently emerged as a powerful modeling tool for various summarization problems. Submodular functions naturally model notions of *coverage*, *information*, *representation* and *diversity*. Moreover, several important combinatorial optimization problems occur as special instances of submodular optimization. For example, the set cover problem is a special case of submodular optimization, since the set cover function is submodular. The set cover function attempts to find a subset of objects which *cover* a given set of concepts. For example, in document summarization, one would like the summary to cover all important and relevant concepts in the document. This is an instance of set cover. Similarly, the facility location problem is a special case of submodular functions. The Facility Location function also naturally models coverage and diversity. Another example of a submodular optimization problem is using a Determinantal point process to model diversity. Similarly, the Maximum-Marginal-Relevance procedure can also be seen as an instance of submodular optimization. All these important models encouraging coverage, diversity and information are all submodular. Moreover, submodular functions can be efficiently combined together, and the resulting function is still submodular. Hence, one could combine one submodular function which models diversity, another one which models coverage and use human supervision to learn a right model of a submodular function for the problem.

While submodular functions are fitting problems for summarization, they also admit very efficient algorithms for optimization. For example, a simple greedy algorithm admits a constant factor guarantee. Moreover, the greedy algorithm is extremely simple to implement and can scale to large datasets, which is very important for summarization problems.

Submodular functions have achieved state-of-the-art for almost all summarization problems. For example, work by Lin and Bilmes, 2012 shows that submodular functions achieve the best results to date on DUC-04, DUC-05, DUC-06 and DUC-07 systems for document summarization. Similarly, work by Lin and Bilmes, 2011, shows that many existing systems for automatic summarization

are instances of submodular functions. This was a break through result establishing submodular functions as the right models for summarization problems.

Submodular Functions have also been used for other summarization tasks. Tschiatschek et al., 2014 show that mixtures of submodular functions achieve state-of-the-art results for image collection summarization. Similarly, Bairi et al., 2015 show the utility of submodular functions for summarizing multi-document topic hierarchies. Submodular Functions have also successfully been used for summarizing machine learning datasets.

Evaluation Techniques

The most common way to evaluate the informativeness of automatic summaries is to compare them with human-made model summaries.

Evaluation techniques fall into intrinsic and extrinsic, inter-textual and intra-textual.

Intrinsic and Extrinsic Evaluation

An intrinsic evaluation tests the summarization system in and of itself while an extrinsic evaluation tests the summarization based on how it affects the completion of some other task. Intrinsic evaluations have assessed mainly the coherence and informativeness of summaries. Extrinsic evaluations, on the other hand, have tested the impact of summarization on tasks like relevance assessment, reading comprehension, etc.

Inter-textual and Intra-textual

Intra-textual methods assess the output of a specific summarization system, and the inter-textual ones focus on contrastive analysis of outputs of several summarization systems.

Human judgement often has wide variance on what is considered a "good" summary, which means that making the evaluation process automatic is particularly difficult. Manual evaluation can be used, but this is both time and labor-intensive as it requires humans to read not only the summaries but also the source documents. Other issues are those concerning coherence and coverage.

One of the metrics used in NIST's annual Document Understanding Conferences, in which research groups submit their systems for both summarization and translation tasks, is the ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation). It essentially calculates n-gram overlaps between automatically generated summaries and previously-written human summaries. A high level of overlap should indicate a high level of shared concepts between the two summaries. Note that overlap metrics like this are unable to provide any feedback on a summary's coherence. Anaphor resolution remains another problem yet to be fully solved. Similarly, for image summarization, Tschiatschek et al., developed a Visual-ROUGE score which judges the performance of algorithms for image summarization.

Current Challenges in Evaluating Summaries Automatically

Evaluating summaries, either manually or automatically, is a hard task. The main difficulty in evaluation comes from the impossibility of building a fair gold-standard against which the results

of the systems can be compared. Furthermore, it is also very hard to determine what a correct summary is, because there is always the possibility of a system to generate a good summary that is quite different from any human summary used as an approximation to the correct output.

Content selection is not a deterministic problem. People are subjective, and different authors would choose different sentences. And individuals may not be consistent. A particular person may chose different sentences at different times. Two distinct sentences expressed in different words can express the same meaning. This phenomenon is known as paraphrasing. We can find an approach to automatically evaluating summaries using paraphrases (ParaEval).

Most summarization systems perform an extractive approach, selecting and copying important sentences from the source documents. Although humans can also cut and paste relevant information of a text, most of the times they rephrase sentences when necessary, or they join different related information into one sentence.

Domain Specific Versus Domain Independent Summarization Techniques

Domain independent summarization techniques generally apply sets of general features which can be used to identify information-rich text segments. Recent research focus has drifted to domain-specific summarization techniques that utilize the available knowledge specific to the domain of text. For example, automatic summarization research on medical text generally attempts to utilize the various sources of codified medical knowledge and ontologies.

Evaluating Summaries Qualitatively

The main drawback of the evaluation systems existing so far is that we need at least one reference summary, and for some methods more than one, to be able to compare automatic summaries with models. This is a hard and expensive task. Much effort has to be done in order to have corpus of texts and their corresponding summaries. Furthermore, for some methods, not only do we need to have human-made summaries available for comparison, but also manual annotation has to be performed in some of them (e.g. SCU in the Pyramid Method). In any case, what the evaluation methods need as an input, is a set of summaries to serve as gold standards and a set of automatic summaries. Moreover, they all perform a quantitative evaluation with regard to different similarity metrics. To overcome these problems, we think that the quantitative evaluation might not be the only way to evaluate summaries, and a qualitative automatic evaluation would be also important.

Examples of Data Mining

Data mining has been used in many applications. Some notable examples of usage are:

Games

Since the early 1960s, with the availability of oracles for certain combinatorial games, also called tablebases (e.g. for 3x3-chess) with any beginning configuration, small-board dots-and-boxes, small-board-hex, and certain endgames in chess, dots-and-boxes, and hex; a new area for data

mining has been opened. This is the extraction of human-usable strategies from these oracles. Current pattern recognition approaches do not seem to fully acquire the high level of abstraction required to be applied successfully. Instead, extensive experimentation with the tablebases – combined with an intensive study of tablebase-answers to well designed problems, and with knowledge of prior art (i.e., pre-tablebase knowledge) – is used to yield insightful patterns. Berlekamp (in dots-and-boxes, etc.) and John Nunn (in chess endgames) are notable examples of researchers doing this work, though they were not – and are not – involved in tablebase generation.

Business

In business, data mining is the analysis of historical business activities, stored as static data in data warehouse databases. The goal is to reveal hidden patterns and trends. Data mining software uses advanced pattern recognition algorithms to sift through large amounts of data to assist in discovering previously unknown strategic business information. Examples of what businesses use data mining is to include performing market analysis to identify new product bundles, finding the root cause of manufacturing problems, to prevent customer attrition and acquire new customers, cross-selling to existing customers, and profiling customers with more accuracy.

- In today's world raw data is being collected by companies at an exploding rate. For example, Walmart processes over 20 million point-of-sale transactions every day. This information is stored in a centralized database, but would be useless without some type of data mining software to analyze it. If Walmart analyzed their point-of-sale data with data mining techniques they would be able to determine sales trends, develop marketing campaigns, and more accurately predict customer loyalty.
- Categorization of the items available in the e-commerce site is a fundamental problem. A correct item categorization system is essential for user experience as it helps determine the items relevant to him for search and browsing. Item categorization can be formulated as a supervised classification problem in data mining where the categories are the target classes and the features are the words composing some textual description of the items. One of the approaches is to find groups initially which are similar and place them together in a latent group. Now given a new item, first classify into a latent group which is called coarse level classification. Then, do a second round of classification to find the category to which the item belongs to.
- Every time a credit card or a store loyalty card is being used, or a warranty card is being filled, data is being collected about the users behavior. Many people find the amount of information stored about us from companies, such as Google, Facebook, and Amazon, disturbing and are concerned about privacy. Although there is the potential for our personal data to be used in harmful, or unwanted, ways it is also being used to make our lives better. For example, Ford and Audi hope to one day collect information about customer driving patterns so they can recommend safer routes and warn drivers about dangerous road conditions.
- Data mining in customer relationship management applications can contribute significantly to the bottom line. Rather than randomly contacting a prospect or customer through a call center or sending mail, a company can concentrate its efforts on prospects that are predicted to have a high likelihood of responding to an offer. More sophisticated methods may be used to optimize resources across campaigns so that one may predict to which

channel and to which offer an individual is most likely to respond (across all potential offers). Additionally, sophisticated applications could be used to automate mailing. Once the results from data mining (potential prospect/customer and channel/offer) are determined, this "sophisticated application" can either automatically send an e-mail or a regular mail. Finally, in cases where many people will take an action without an offer, "uplift modeling" can be used to determine which people have the greatest increase in response if given an offer. Uplift modeling thereby enables marketers to focus mailings and offers on persuadable people, and not to send offers to people who will buy the product without an offer. Data clustering can also be used to automatically discover the segments or groups within a customer data set.

- Businesses employing data mining may see a return on investment, but also they recognize that the number of predictive models can quickly become very large. For example, rather than using one model to predict how many customers will churn, a business may choose to build a separate model for each region and customer type. In situations where a large number of models need to be maintained, some businesses turn to more automated data mining methodologies.
- Data mining can be helpful to human resources (HR) departments in identifying the characteristics of their most successful employees. Information obtained – such as universities attended by highly successful employees – can help HR focus recruiting efforts accordingly. Additionally, Strategic Enterprise Management applications help a company translate corporate-level goals, such as profit and margin share targets, into operational decisions, such as production plans and workforce levels.
- Market basket analysis, relates to data-mining use in retail sales. If a clothing store records the purchases of customers, a data mining system could identify those customers who favor silk shirts over cotton ones. Although some explanations of relationships may be difficult, taking advantage of it is easier. The example deals with association rules within transaction-based data. Not all data are transaction based and logical, or inexact rules may also be present within a database.
- Market basket analysis has been used to identify the purchase patterns of the Alpha Consumer. Analyzing the data collected on this type of user has allowed companies to predict future buying trends and forecast supply demands.
- Data mining is a highly effective tool in the catalog marketing industry. Catalogers have a rich database of history of their customer transactions for millions of customers dating back a number of years. Data mining tools can identify patterns among customers and help identify the most likely customers to respond to upcoming mailing campaigns.
- Data mining for business applications can be integrated into a complex modeling and decision making process. LIONsolver uses Reactive business intelligence (RBI) to advocate a "holistic" approach that integrates data mining, modeling, and interactive visualization into an end-to-end discovery and continuous innovation process powered by human and automated learning.
- In the area of decision making, the RBI approach has been used to mine knowledge that is progressively acquired from the decision maker, and then self-tune the decision meth-

od accordingly. The relation between the quality of a data mining system and the amount of investment that the decision maker is willing to make was formalized by providing an economic perspective on the value of "extracted knowledge" in terms of its payoff to the organization This decision-theoretic classification framework was applied to a real-world semiconductor wafer manufacturing line, where decision rules for effectively monitoring and controlling the semiconductor wafer fabrication line were developed.

- An example of data mining related to an integrated-circuit (IC) production line is described in the paper "Mining IC Test Data to Optimize VLSI Testing." In this paper, the application of data mining and decision analysis to the problem of die-level functional testing is described. Experiments mentioned demonstrate the ability to apply a system of mining historical die-test data to create a probabilistic model of patterns of die failure. These patterns are then utilized to decide, in real time, which die to test next and when to stop testing. This system has been shown, based on experiments with historical test data, to have the potential to improve profits on mature IC products. Other examples of the application of data mining methodologies in semiconductor manufacturing environments suggest that data mining methodologies may be particularly useful when data is scarce, and the various physical and chemical parameters that affect the process exhibit highly complex interactions. Another implication is that on-line monitoring of the semiconductor manufacturing process using data mining may be highly effective.

Science and Engineering

In recent years, data mining has been used widely in the areas of science and engineering, such as bioinformatics, genetics, medicine, education and electrical power engineering.

- In the study of human genetics, sequence mining helps address the important goal of understanding the mapping relationship between the inter-individual variations in human DNA sequence and the variability in disease susceptibility. In simple terms, it aims to find out how the changes in an individual's DNA sequence affects the risks of developing common diseases such as cancer, which is of great importance to improving methods of diagnosing, preventing, and treating these diseases. One data mining method that is used to perform this task is known as multifactor dimensionality reduction.
- In the area of electrical power engineering, data mining methods have been widely used for condition monitoring of high voltage electrical equipment. The purpose of condition monitoring is to obtain valuable information on, for example, the status of the insulation (or other important safety-related parameters). Data clustering techniques – such as the self-organizing map (SOM), have been applied to vibration monitoring and analysis of transformer on-load tap-changers (OLTCS). Using vibration monitoring, it can be observed that each tap change operation generates a signal that contains information about the condition of the tap changer contacts and the drive mechanisms. Obviously, different tap positions will generate different signals. However, there was considerable variability amongst normal condition signals for exactly the same tap position. SOM has been applied to detect abnormal conditions and to hypothesize about the nature of the abnormalities.
- Data mining methods have been applied to dissolved gas analysis (DGA) in power transformers. DGA, as a diagnostics for power transformers, has been available for many years.

Methods such as SOM has been applied to analyze generated data and to determine trends which are not obvious to the standard DGA ratio methods (such as Duval Triangle).

- In educational research, where data mining has been used to study the factors leading students to choose to engage in behaviors which reduce their learning, and to understand factors influencing university student retention. A similar example of social application of data mining is its use in expertise finding systems, whereby descriptors of human expertise are extracted, normalized, and classified so as to facilitate the finding of experts, particularly in scientific and technical fields. In this way, data mining can facilitate institutional memory.
- Data mining methods of biomedical data facilitated by domain ontologies, mining clinical trial data, and traffic analysis using SOM.
- In adverse drug reaction surveillance, the Uppsala Monitoring Centre has, since 1998, used data mining methods to routinely screen for reporting patterns indicative of emerging drug safety issues in the WHO global database of 4.6 million suspected adverse drug reaction incidents. Recently, similar methodology has been developed to mine large collections of electronic health records for temporal patterns associating drug prescriptions to medical diagnoses.
- Data mining has been applied to software artifacts within the realm of software engineering: Mining Software Repositories.

Human Rights

Data mining of government records – particularly records of the justice system (i.e., courts, prisons) – enables the discovery of systemic human rights violations in connection to generation and publication of invalid or fraudulent legal records by various government agencies.

Medical Data Mining

Some machine learning algorithms can be applied in medical field as second-opinion diagnostic tools and as tools for the knowledge extraction phase in the process of knowledge discovery in databases. One of these classifiers (called *Prototype exemplar learning classifier* (PEL-C) is able to discover syndromes as well as atypical clinical cases.

In 2011, the case of Sorrell v. IMS Health, Inc., decided by the Supreme Court of the United States, ruled that pharmacies may share information with outside companies. This practice was authorized under the 1st Amendment of the Constitution, protecting the "freedom of speech." However, the passage of the Health Information Technology for Economic and Clinical Health Act (HITECH Act) helped to initiate the adoption of the electronic health record (EHR) and supporting technology in the United States. The HITECH Act was signed into law on February 17, 2009 as part of the American Recovery and Reinvestment Act (ARRA) and helped to open the door to medical data mining. Prior to the signing of this law, estimates of only 20% of United States-based physicians were utilizing electronic patient records. Søren Brunak notes that "the patient record becomes as information-rich as possible" and thereby "maximizes the data mining opportunities." Hence, electronic patient records further expands the possibilities regarding medical data mining thereby opening the door to a vast source of medical data analysis.

Spatial Data Mining

Spatial data mining is the application of data mining methods to spatial data. The end objective of spatial data mining is to find patterns in data with respect to geography. So far, data mining and Geographic Information Systems (GIS) have existed as two separate technologies, each with its own methods, traditions, and approaches to visualization and data analysis. Particularly, most contemporary GIS have only very basic spatial analysis functionality. The immense explosion in geographically referenced data occasioned by developments in IT, digital mapping, remote sensing, and the global diffusion of GIS emphasizes the importance of developing data-driven inductive approaches to geographical analysis and modeling.

Data mining offers great potential benefits for GIS-based applied decision-making. Recently, the task of integrating these two technologies has become of critical importance, especially as various public and private sector organizations possessing huge databases with thematic and geographically referenced data begin to realize the huge potential of the information contained therein. Among those organizations are:

- offices requiring analysis or dissemination of geo-referenced statistical data
- public health services searching for explanations of disease clustering
- environmental agencies assessing the impact of changing land-use patterns on climate change
- geo-marketing companies doing customer segmentation based on spatial location.

Challenges in Spatial mining: Geospatial data repositories tend to be very large. Moreover, existing GIS datasets are often splintered into feature and attribute components that are conventionally archived in hybrid data management systems. Algorithmic requirements differ substantially for relational (attribute) data management and for topological (feature) data management. Related to this is the range and diversity of geographic data formats, which present unique challenges. The digital geographic data revolution is creating new types of data formats beyond the traditional "vector" and "raster" formats. Geographic data repositories increasingly include ill-structured data, such as imagery and geo-referenced multi-media.

There are several critical research challenges in geographic knowledge discovery and data mining. Miller and Han offer the following list of emerging research topics in the field:

- Developing and supporting geographic data warehouses (GDW's): Spatial properties are often reduced to simple aspatial attributes in mainstream data warehouses. Creating an integrated GDW requires solving issues of spatial and temporal data interoperability – including differences in semantics, referencing systems, geometry, accuracy, and position.
- Better spatio-temporal representations in geographic knowledge discovery: Current geographic knowledge discovery (GKD) methods generally use very simple representations of geographic objects and spatial relationships. Geographic data mining methods should recognize more complex geographic objects (i.e., lines and polygons) and relationships (i.e., non-Euclidean distances, direction, connectivity, and interaction through attributed geographic space such as terrain). Furthermore, the time dimension needs to be more fully

integrated into these geographic representations and relationships.

- Geographic knowledge discovery using diverse data types: GKD methods should be developed that can handle diverse data types beyond the traditional raster and vector models, including imagery and geo-referenced multimedia, as well as dynamic data types (video streams, animation).

Temporal Data Mining

Data may contain attributes generated and recorded at different times. In this case finding meaningful relationships in the data may require considering the temporal order of the attributes. A temporal relationship may indicate a causal relationship, or simply an association.

Sensor Data Mining

Wireless sensor networks can be used for facilitating the collection of data for spatial data mining for a variety of applications such as air pollution monitoring. A characteristic of such networks is that nearby sensor nodes monitoring an environmental feature typically register similar values. This kind of data redundancy due to the spatial correlation between sensor observations inspires the techniques for in-network data aggregation and mining. By measuring the spatial correlation between data sampled by different sensors, a wide class of specialized algorithms can be developed to develop more efficient spatial data mining algorithms.

Visual Data Mining

In the process of turning from analog into digital, large data sets have been generated, collected, and stored discovering statistical patterns, trends and information which is hidden in data, in order to build predictive patterns. Studies suggest visual data mining is faster and much more intuitive than is traditional data mining.

Music Data Mining

Data mining techniques, and in particular co-occurrence analysis, has been used to discover relevant similarities among music corpora (radio lists, CD databases) for purposes including classifying music into genres in a more objective manner.

Surveillance

Data mining has been used by the U.S. government. Programs include the Total Information Awareness (TIA) program, Secure Flight (formerly known as Computer-Assisted Passenger Prescreening System (CAPPS II)), Analysis, Dissemination, Visualization, Insight, Semantic Enhancement (ADVISE), and the Multi-state Anti-Terrorism Information Exchange (MATRIX). These programs have been discontinued due to controversy over whether they violate the 4th Amendment to the United States Constitution, although many programs that were formed under them continue to be funded by different organizations or under different names.

In the context of combating terrorism, two particularly plausible methods of data mining are "pattern mining" and "subject-based data mining".

Pattern Mining

"Pattern mining" is a data mining method that involves finding existing patterns in data. In this context *patterns* often means association rules. The original motivation for searching association rules came from the desire to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. For example, an association rule "beer ☐ potato chips (80%)" states that four out of five customers that bought beer also bought potato chips.

In the context of pattern mining as a tool to identify terrorist activity, the National Research Council provides the following definition: "Pattern-based data mining looks for patterns (including anomalous data patterns) that might be associated with terrorist activity — these patterns might be regarded as small signals in a large ocean of noise." Pattern Mining includes new areas such a Music Information Retrieval (MIR) where patterns seen both in the temporal and non temporal domains are imported to classical knowledge discovery search methods.

Subject-based Data Mining

"Subject-based data mining" is a data mining method involving the search for associations between individuals in data. In the context of combating terrorism, the National Research Council provides the following definition: "Subject-based data mining uses an initiating individual or other datum that is considered, based on other information, to be of high interest, and the goal is to determine what other persons or financial transactions or movements, etc., are related to that initiating datum."

Knowledge Grid

Knowledge discovery "On the Grid" generally refers to conducting knowledge discovery in an open environment using grid computing concepts, allowing users to integrate data from various online data sources, as well make use of remote resources, for executing their data mining tasks. The earliest example was the Discovery Net, developed at Imperial College London, which won the "Most Innovative Data-Intensive Application Award" at the ACM SC02 (Supercomputing 2002) conference and exhibition, based on a demonstration of a fully interactive distributed knowledge discovery application for a bioinformatics application. Other examples include work conducted by researchers at the University of Calabria, who developed a Knowledge Grid architecture for distributed knowledge discovery, based on grid computing.

References

- Battiti, Roberto; and Brunato, Mauro; Reactive Business Intelligence. From Data to Models to Insight, Reactive Search Srl, Italy, February 2011. ISBN 978-88-905795-0-9.
- Monk, Ellen; Wagner, Bret (2006). Concepts in Enterprise Resource Planning, Second Edition. Boston, MA: Thomson Course Technology. ISBN 0-619-21663-8. OCLC 224465825.
- Zhu, Xingquan; Davidson, Ian (2007). Knowledge Discovery and Data Mining: Challenges and Realities. New York, NY: Hershey. pp. 163–189. ISBN 978-1-59904-252-7.
- Haag, Stephen; Cummings, Maeve; Phillips, Amy (2006). Management Information Systems for the information age. Toronto: McGraw-Hill Ryerson. p. 28. ISBN 0-07-095569-7. OCLC 63194770.
- Good, P. I.; Hardin, J. W. (2009). Common Errors in Statistics (And How to Avoid Them) (3rd ed.). Hoboken, New Jersey: Wiley. p. 211. ISBN 978-0-470-45798-6.

- Fotheringham, A. Stewart; Brunsdon, Chris; Charlton, Martin (2002). Geographically weighted regression: the analysis of spatially varying relationships (Reprint ed.). Chichester, England: John Wiley. ISBN 978-0-471-49616-8.
- Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich. Introduction to Information Retrieval. Cambridge University Press. ISBN 978-0-521-86571-5.
- Bailey, Ken (1994). "Numerical Taxonomy and Cluster Analysis". Typologies and Taxonomies. p. 34. ISBN 9780803952591.
- Hájek, Petr; Feglar, Tomas; Rauch, Jan; and Coufal, David; The GUHA method, data preprocessing and mining, Database Support for Data Mining Applications, Springer, 2004, ISBN 978-3-540-22479-2
- Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin (2005). "Chapter 6. Association Analysis: Basic Concepts and Algorithms" (PDF). Introduction to Data Mining. Addison-Wesley. ISBN 0-321-32136-7.
- Angiulli, F.; Pizzuti, C. (2002). Fast Outlier Detection in High Dimensional Spaces. Principles of Data Mining and Knowledge Discovery. Lecture Notes in Computer Science. 2431. p. 15. doi:10.1007/3-540-45681-3_2. ISBN 978-3-540-44037-6.
- Witten, Ian H.; Frank, Eibe; Hall, Mark A. (30 January 2011). Data Mining: Practical Machine Learning Tools and Techniques (3 ed.). Elsevier. ISBN 978-0-12-374856-0.
- Mena, Jesús (2011). Machine Learning Forensics for Law Enforcement, Security, and Intelligence. Boca Raton, FL: CRC Press (Taylor & Francis Group). ISBN 978-1-4398-6069-4.
- Kantardzic, Mehmed (2003). Data Mining: Concepts, Models, Methods, and Algorithms. John Wiley & Sons. ISBN 0-471-22852-4. OCLC 50055336.