Data Preparation

Dr. Cahit Karakuş

What is Big Data

- Big data is defined as datasets that are too large and complex for businesses' existing systems to handle using their traditional capabilities to capture, store, manage and analyze the data sets.
- Big data is no different than traditional data because if it can't be analyzed to provide insight and help with decision making, its value is limited.
- Volume, velocity, veracity and variety are often used to represent the defining features of big data.
    - Volume refers to the massive amount of data involved.
    - Velocity refers to the fact that the data comes in at quick speeds or in real time, such as streaming videos and news feeds.
    - Variety refers to unstructured and unprocessed data, such as comments in social media, emails, global positioning system (GPS) measurements, etc.
    - Veracity refers to the quality of the data including extent of cleanliness (without errors or data integrity issues), reliability and representationally faithful.

What is Data Analytics

- Data analytics is defined as the science of examining raw data, removing excess noise, and organizing the data with the purpose of drawing conclusions for decision making.
- Data analytics often involves the technologies, systems, practices, methodologies, databases, and applications used to analyze diverse business data to help organizations make sound and timely business decisions.
- The intent of data analytics is to transform raw data into valuable information.
- Data analytics is used in today’s business world by examining the data to generate models for predictions of patterns and trends.
- When used effectively, data analytics gives us the ability to search through large and unstructured data to identify unknown patterns or relationships, which when organized, is used to provide useful information.

The Power of Data Analytics

- With a wealth of data on their hands, companies are empowered by using data analytics to discover various patterns, investigate anomalies, forecast future behavior, and so forth.
- Patterns discovered from historical data enable businesses to identify future opportunities and risks.
- In addition to producing more value externally, studies show that data analytics affects internal processes, improving productivity, utilization, and growth.

Benefits and Costs of Data Analytics

- Reformatting, cleaning, and consolidating large volumes of data from multiple sources and platforms can be especially time consuming.
- It is estimated that data analytics professionals spend 50 percent to 90 percent of their time cleaning data for analysis.
- The cost to scrub the data includes the salaries of the data analytics scientists and the cost of the technology to prepare and analyze the data.
- As with other information, there is a cost to produce data.

The impact of Data Analytics

- Many companies address the likely possibility that the data their organizations hold influence their market value.
- Facebook, for example, has a large amount of its market value driven by the number of users on the platform and the amount of data those users contribute which is sold to third parties.
- Data analytics often also involves data management and business intelligence with knowledge of business functional areas. Today there is an increasing number of investments in data analytics and increasing demand for data analytics–related tasks
- The real value of data comes from the use of data analytics.
- Companies are getting much smarter about using data analytics to discover various patterns, investigate anomalies, forecast future behavior, and so forth.
- For example, companies can use their data to do more directed marketing campaigns based on patterns observed in their data.
- That can give them a competitive advantage and it can also be used on historical data to enable businesses to identify future opportunities and risks.

The Impact of Data Analytics

- We refer to reporting as the responsibilitis of issuing the statements and the reports.
- The reporting includes a number of estimates and valuations that can be evaluated through use of data analytics.
- Many statements are just estimates and someone can use data analytics to evaluate those estimates.
- Data analytics may be used to scan the environment—that is, by scanning social media to identify potential risks and opportunities to the firm.
- Data analytics plays a very critical role in the future of audit.
- By using data analytics, auditors can spend less time looking for evidence, which will allow more time for presenting their findings and making judgments.
- Data analytics also expands auditors’ capabilities in services such as testing for fraudulent transactions and automating compliance-monitoring activities.

Cognitive ability -1

- Cognitive ability is considered our general intelligence level and provides us the ability to think abstractly, comprehend complex situations, problem solve, and gather and retain information from our life experiences (Plomin, 1999).

- Cognitive ability includes components such as mechanical reasoning, spatial awareness, numerical reasoning, critical thinking, and general intelligence (Davies, 2017).
- This ability drives the processes of how we choose to engage in learning and how we learn, remember, and solve problems of various levels of complexity, and the best learners do not just memorize random pieces of information (Nordin & Dakwah, 2015).

- When our cognitive abilities are developed at an advanced level, the process of learning is more straightforward for us. In contrast, when cognitive abilities are not as developed, the learning process is more challenging (Bhat, 2016).

Cognitive ability -2

- The cycle of growth for cognitive development starts when an individual is around two years old and moves through a series of ups and downs in performance levels, which continue into early adulthood (Fischer & Bidell, 2006).

- Brain and human behaviors research indicate that the capacity to engage in reflective judgment through advanced levels of abstract thinking does not emerge until early adulthood.
- Fischer’s (1980) model of cognitive skill theory (Skill Theory) added a background to how complex reasoning is developed.
- Skill Theory outlines the professional maturation of individuals and how an individual’s environment contributes to the development of skills.
- The developmental levels of Skill Theory occur as the brain conducts re-organizations of behavior.

Cognitive ability -3

- This reorganization of behavior facilitates the use of new higher-order cognitive ability levels built upon combinations of previously constructed lower-order cognitive abilities.
- As individuals develop complex reasoning, their developmental range will fluctuate between functional and optimal skill levels based on their environment.
- An individual must solidify those less complex cognitive skills, and then they will be able to develop more complex cognitive skills.
- Reflective judgment requires an individual to coordinate multiple views, so this skill cannot develop until adults can engage in abstract thought (Fischer & Pruyne, 2003).
- Individuals move through periods when their skills grow at a faster pace because they are in an environment that supports optimal performance
- During their functional performance, they are not pushing the limits of their cognitive ability (optimal performance), and they grow slowly or do not progress at all (Fischer, 2008).

Data analytics processes

- Data scrubbing and data preparation
- Data quality
- Descriptive data analysis
- Data analysis through data manipulation
- Define and address problems through statistical analysis
- Data visualization and data reporting

How does higher education help

- A primary goal of education is to promote the thinking skills of students.
- Universities recognize the importance of enabling cognitive development in their students and the impact on a student’s level of cognitive ability.
- For students within accounting programs, this is especially important because the accounting profession requires more creativity and innovative thinking to stay competitive within the market (Thompson & Washington, 2015).
- Cognitive ability goes beyond just basic memorization or imitation; it gives a person the capacity to comprehend situations and determine how to assess and resolve them (Plomin & Von Stumm, 2018).
- Cognitive ability is closely associated with higher achievement in education, occupations, and better health outcomes.

It’s a shared responsibility

- Business students, especially those in accounting, finance, and audit positions, are expected to have higher levels of cognitive ability (Reding & Newman, 2017).
- The focus of today’s accounting and business professionals is to provide value-added services, and higher education serves as a pipeline for businesses to obtain new talent.
- With the increasing reliance on big data to help within the decision-making process, higher levels of cognitive ability are critical in today's business world, and especially important for graduating students.
- Students entering the business world with lower cognitive ability with have direct impact on the profitability and productivity of organizations.
- Cognitive development occurs when individuals allow their growth and experiences to enhance their cognitive ability, maintain an adequate level of engagement until they reach their optimal level of cognitive ability, and work to maintain that optimal level.
- Business leaders and their organizations have a vested interest in continuing to support and foster the growth of cognitive ability within their employees.
- Without that ongoing support, those employees will revert to their “functional” level and will become static in their growth.

Data Preparation

What is a Dataset?

- A Dataset is a grouped set of data into a collection that developers can work with to achieve their goals.
- In a dataset, the rows represent the number of data points and the columns represent the features of the Dataset. These are often used to gain insights, make informed decisions, or train algorithms in areas such as machine learning, business, and government.
- Datasets can vary in size and complexity and often require cleaning and preprocessing to ensure data quality and suitability for analysis or modeling.

Dataset Types

- Numeric Dataset: Contains numerical data points that can be solved with equations. These include temperature, humidity, signs, etc.
- Categorical Dataset: Contains categories like color, gender, occupation, games, sports, etc.
- Web Dataset: These contain datasets created by calling APIs using HTTP requests and filling them with values for data analysis. These are mostly stored in JSON (JavaScript Object Notation) formats.
- Time Series Dataset: These contain datasets between a specific period, for example, changes in geographical terrain over time.
- Image Dataset: Contains a dataset consisting of images. This is mostly used to distinguish types of diseases, heart conditions, etc.
- Ordinal Dataset: These datasets contain data sorted according to rank, for example, customer reviews, movie ratings, etc.
- Partitioned Dataset: In these datasets, data points are divided into different members or different sections.
- File-Based Datasets: These datasets are stored in files, .csv or .xlsx files in Excel.
- Bivariate Dataset: In this dataset, 2 classes or features are directly related to each other. For example, height and weight in a dataset are directly related to each other.
- Multivariate Dataset: In this type of dataset, as the name suggests, 2 or more classes are directly related to each other. For example, attendance and homework grades are directly related to a student’s overall grade.

Dataset Properties

- **Data center:** This refers to the “middle” value of the data, usually measured by the mean, median, or mode. It helps to understand where most of the data points are concentrated.
- **Data skewness:** This indicates how symmetric the distribution of the data is. A perfectly symmetric distribution (like a normal distribution) has a skewness of 0. Positive skew means the data is clustered to the left, while negative skew means the data is clustered to the right.
- **Spread among data members:** This describes how far the data points diverge from the center. Common measurements include standard deviation or variance, which measure how much individual points deviate from the mean.
- **Existence of outliers:** These are data points that fall significantly outside the overall pattern. Identifying outliers can be important because they can affect the results of the analysis and may require further investigation.
- **Correlation between data:** This refers to the strength and direction of relationships between different variables in the data set. A positive correlation indicates that values in one variable tend to increase as the other increases, while a negative correlation indicates that they move in opposite directions. No correlation means that there is no linear relationship between the variables. The type of probability distribution the data follows: Understanding the distribution (e.g., normal, uniform, binomial) helps us estimate the probability of finding certain values in the data and choose appropriate statistical methods for analysis.

Properties of a Dataset

- The properties of a dataset can refer to the columns present in the dataset. The properties of a dataset are the most critical aspect of the dataset because, based on the properties of each existing data point, will there be a possibility to deploy models to find outputs to predict the properties of any new data points that may be added to the dataset?
- It is possible to determine standard properties only from some datasets because their functionality and data will be completely different when compared to other datasets. Some possible properties of a dataset are:
- Numerical Properties: These can include numerical values like height, weight, etc. These can be continuous or discrete variables in a range.
- Categorical Properties: These include multiple classes/categories like gender, color, etc.
- Metadata: Contains a general description of a dataset. Usually in very large datasets, having an idea/description will save a lot of time and increase efficiency when the dataset is passed on to a new developer.
- Data Size: It refers to the number of inputs and features present in the file containing the dataset.
- Data Formatting: Datasets available online are available in various formats. Some of these are JSON (JavaScript Object Notation), CSV (Comma Separated Value), XML (Extensible Markup Language), DataFrame and Excel Files (xlsx or xlsm). Especially for large datasets containing images for disease detection, when downloading files from the internet, they come in zip files that will be required to extract them into separate components in the system.
- Target Variable: It is the feature whose values/attributes are referenced to get output from other features with machine learning techniques.
- Data Inputs: These refer to the individual values of the data in the Dataset. They play a major role in data analysis.

Methods Used in Data Sets

- 1. Loading and Reading Datasets: A set of methods used to initially load and read datasets to perform the required tasks.
- 2. Exploratory Data Analysis: We run these functions on a dataset to perform Data Analysis and visualize it.
- 3. Data Preprocessing: Before a dataset is analyzed, it is preprocessed using certain methods to remove erroneous values and mislabeled data points.
- 4. Data Manipulation: Data points in the dataset are edited/rearranged to change the features. At some points, even the features of the dataset are changed to reduce computational complexity, etc. This may include methods or functions that merge columns, add new data points, etc.
- 5. Data Visualization: Methods used to explain the dataset to non-technical people; for example, using bar charts and graphs to provide a pictorial representation of the company/business's dataset.
- 6. Data Indexing, Data Subsets: We use data indexing or creating precise subsets to express a specific feature in a dataset.
- 7. Export Data: Methods used to export the data you are working with in different formats according to need.

Data - Datasets - Database

- Data: It includes facts like numerical data, categorical data, features, etc. However, data alone cannot be used properly. To perform analysis, a large amount of data needs to be collected.
- Datasets: A dataset is a collection of data that contains data specific to its category and nothing else. This is used to develop Machine Learning models that perform Data Analysis, Data and Feature Engineering. Datasets can be structured (Height, weight analysis) or unstructured (audio files, videos, images).
- Database: A database contains multiple datasets. It is possible for a database to host several Datasets that may not be related to each other. The data in databases can be queried to perform various applications.
- There are various types of databases to host various types of data, structured or unstructured data. These are divided into SQL databases and NoSQL databases.

Data - Datasets - Database

Data	Datasets	Database
Contains only raw facts or information	It has a data collection or data entry structure.	It consists of collections stored in an orderly manner.
It is devoid of any context and is unedited.	Organizes data into rows and columns	Data is organized into tables that can span multiple dimensions.
It contains the foundations of knowledge and provides the foundation/backbone of datasets/databases.	It structures the data and provides meaningful inferences from it.	Structured data exists and relationships between features are defined in detail.
It cannot be manipulated due to a structural deficiency.	Can be manipulated with the help of Python Libraries.	It can be manipulated through a series of queries, operations, or scripts.
It needs to be preprocessed and converted before going any further.	It can be used for Data Analysis, Data Modeling and Data Visualization.	Data can be processed through Queries or Transactions.

Training Dataset-1

- Machine learning models rely on access to high-quality training data. Understanding how to effectively collect, prepare, and test your data will help you unlock the full value of AI.
- It requires an initial dataset, called a training dataset, to act as a foundation for further implementation and use.
- This dataset is the foundation of the program’s growing library of knowledge. The training dataset must be labeled correctly so that the model can process and learn from it.

Training Dataset-2

- Machine Learning algorithms learn from data.
- They find relationships between labeled data structures, develop understanding, make decisions, and evaluate their confidence in the training data they are given.
- The better the training data, the better the model will perform.
- In fact, the quality and quantity of your machine learning training data has as much to do with the success of your data project as the algorithms themselves.
- First, it’s important to have a common understanding of what we mean by a dataset. The definition of a dataset is that it has both rows and columns, where each row contains an observation.
- This observation could be an image, an audio clip, text, or vide

Training Dataset-3

- Now, even if you have a large amount of well-structured data stored in your dataset, it may not actually be labeled enough to serve as a training dataset for your model.
- For example, autonomous vehicles don’t just need images of roads; they need labeled images that annotate each car, pedestrian, street sign, and more.
- Sentiment analysis projects require labels that help an algorithm understand when someone is using slang or sarcasm.
- Chatbots need entity extraction and careful syntactic analysis, not just raw language. In other words, the data you want to use for training often needs to be enriched or labeled.
- You may also need to collect more to power your algorithms. Chances are, the data you have stored isn’t quite ready to be used to train machine learning algorithms.

Sufficient training data

- There’s no hard and fast rule for how much data you need. After all, different use cases will require different amounts of data.
- Models where you need your model to be confident (like self-driving cars) will require a lot of data, while a very narrow text-based sentiment model will require much less data.
- As a general rule, you’ll need more data than you think.

What is the difference between training data and big data?

- Big data and training data are not the same thing. Gartner calls big data “high volume, high velocity, and/or high variety,” and this information usually needs to be processed in some way to be truly useful. Training data, as mentioned above, is labeled data used to teach AI models or machine learning algorithms.

Determining the Need for Training Data

- There are many factors at play in deciding how much machine learning training data you need.
- First and foremost, how important is accuracy? Let’s say you’re building a sentiment analysis algorithm. Your problem is complex, yes, but it’s not a matter of life or death. A sentiment algorithm that’s 85% or 90% accurate is more than enough for most people’s needs, and a false positive or negative here won’t significantly change anything.
- Now, a cancer detection model or a self-driving car algorithm? That’s a different story. A cancer detection model that might miss important indicators is literally a matter of life or death.
- Of course, more complex use cases usually require more data than less complex ones. A computer vision that’s trying to identify objects, as opposed to a computer vision that’s just trying to identify food, will generally need less training data. The more classes you hope your model can identify, the more examples it will need.
- It’s important to remember that there really is no such thing as too much high-quality data. Better training data, and more of it, will improve your models. Of course, there is a point where the marginal gains from adding more data become very small, so you want to keep an eye on that and your data budget. The threshold for success needs to be determined, but with careful iteration, it can be overcome with more and better data.

Preparing Training Data

- The truth is, most data is messy or incomplete.
- Take a picture, for example. To a machine, an image is just a bunch of pixels. Some might be green, some might be brown, but a machine doesn’t know it’s a tree until it has a label that essentially says that this collection of pixels is a tree. If a machine sees enough labeled images of a tree, it can start to understand that similar groups of pixels in an unlabeled image also make up a tree.
- So how do you prepare the training data so that it has the features and labels that it needs to do well?
- The best way is with a human in the loop. Or, more accurately, humans in the loop.
- Ideally, you’ll have a variety of annotators (in some cases, you might need domain experts) who can label the data accurately and efficiently.
- Humans can also look at an output, such as a model’s prediction about whether an image is actually a dog, and verify or correct that output (i.e., “yes, that’s a dog” or “no, that’s a cat”). This is known as ground truth tracking and is part of the human process in the iterative cycle. The more accurate your training data labels are, the better your model will perform. It can be helpful to find a data partner who can provide annotation tools and access to crowd workers for the often time-consuming data labeling process.

Test Set

- To build a machine learning algorithm, you need both training and test data.
- After a model is trained on a training set, it is typically evaluated on a test set. Often, these sets are drawn from the same general data set, but the training set needs to be labeled or enriched to increase the confidence and accuracy of an algorithm.
- How should you split a data set into test and training sets?
- In general, training data is split more or less randomly, ensuring that it captures important classes that you know about in advance. For example, if you’re trying to build a model that can read receipt images from multiple stores, you’ll want to avoid training your algorithm on images from a single franchise. This will make your model more robust and help prevent overfitting.

Determining whether the test dataset is biased

- This is an important question as companies work to make AI more ethical and effective for everyone. Bias can occur at many stages of the AI creation process, so bias must be reduced at every step of the way.
- When collecting training data, it is important to ensure that the data is representative of all use cases and end users. To reduce the potential for bias at this stage, you want to make sure that there is a diverse group of people labeling the data and monitoring model performance.
- Finally, key performance indicators include bias as a measurable factor.

Testing and Evaluating Training Data

- Typically, when you build a model, you split your labeled dataset into training and test sets (although sometimes your test set may be unlabeled).
- And of course, you train your algorithm on the former and validate its performance on the latter.
- What happens when your validation set doesn’t give you the results you’re looking for? You’ll need to update your weights, drop or add labels, try different approaches, and retrain your model. When you do this, it’s incredibly important to do it with your datasets split in the same way.
- Why? This is the best way to judge success. You’ll be able to see the labels and decisions it made and where it fell apart. Different training sets can lead to dramatically different results on the same algorithm, so when testing different models, you need to use the same training data to see if you’re really getting better.
- Your test data won’t have an equal amount of every category you’re hoping to identify. To use a simple example: if your computer vision algorithm sees 10,000 examples of dogs and only 5 examples of cats, it’s likely to have trouble identifying cats. The important thing to keep in mind here is what success means for your model in the real world. If your classifier is only trying to identify dogs, poor performance at identifying cats is probably not a deal breaker. However, you will want to evaluate model success on the labels you will need in production. What if you don’t have enough information to achieve the level of accuracy you want? Chances are, you will need more training data. Models built on a few thousand rows are generally not robust enough to be successful for large-scale business applications.

Data Sources for Applications, Programs and Analytics Tools

- It can be external, such as sensors, trackers, web logs, computer system logs and feeds.
- It can be machines that provide data from data-generating programs.
- The data sources can be structured, semi-structured, highly structured, or unstructured.
- The data sources can be social media (L4 Data Processing Layer)
- A source can be internal.
- The sources can be data repositories, such as a database, relational database, flat file, spreadsheet, mail server, web server, directory services.
- It can be files, such as text or comma-separated values (CSV).
- The source can be a data store for applications (L4 Data Processing Layer)

Structured Data Sources

- SQL Server, MySQL, Microsoft Access database, Oracle DBMS, IBMDB2, Informix, Amazon SimpleDB, or a file collection directory on a server
- Data dictionary – which provides references for accessing data, consists of a set of main lookup tables.

Unstructured Data Sources

- Distributed data over high-speed networks that require high-speed processing

- Sources are distributed file systems. Sources are file types such as .txt (text file), .csv (comma-separated values file).

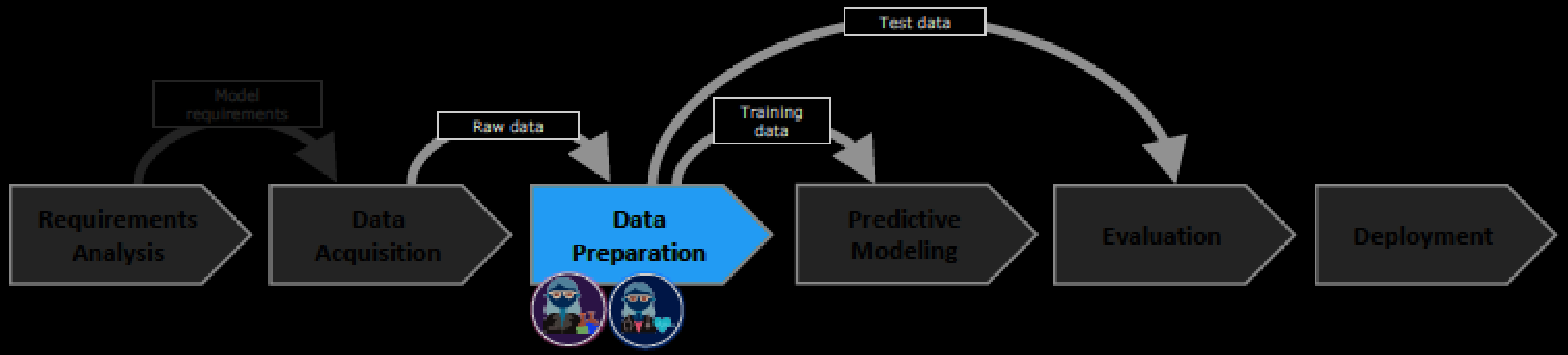
- Data can be in the form of key-value pairs such as hash key-value pairs.
- Data can have internal structures such as email, Facebook pages, twitter messages, etc.

- Data does not model, does not expose relationships, hierarchy relationships, or object-oriented features such as extensibility.

Testing, Evaluation and Validation

- Testing is trying to find out something about it ("to establish evidence; to prove by experiment the truth, reality, or quality of it") according to the Collaborative International Dictionary of English, and verifying is proving that something is valid ("To confirm; to make valid" according to the Collaborative International Dictionary of English).
- In both industry and academia, they are sometimes used interchangeably, with the idea being that different models are tested to improve an internal process (test set as a development set) and the final model is the one that needs to be validated before actual use on unseen data (validation set).
- “The machine learning literature often reverses the meaning of ‘validation’ and ‘test’ sets. This is the most obvious example of the terminological confusion that pervades AI research. However, the important concept to retain is that the final set, whether called test or validation, should be used only in the final experiment. To obtain more stable results and use all valuable data for training, a dataset can be repeatedly split into several training and one validation dataset. This process is known as cross-validation . An additional test dataset, normally obtained from cross-validation, is used to verify model performance.

Data Preparation



Model requirements

Raw data

Test data

Training data

Requirements Analysis

Data Acquisition

Data Preparation

Predictive Modeling

Evaluation

Deployment

Data Preparation

- Exploration
- Quality assessment
- Cleansing
- Labeling
- Imputation
- Feature engineering

Roles Data Scientist Domain Expert (Data) Engineer

Data Preparation

- Data preparation can either make or break the predictive ability of the model!

- Data preparation is the process of adding, deleting, or transforming the training set data.

- Sometimes, preprocessing the data can lead to unexpected improvements in model accuracy.

- Data preparation is an important step and one should try the appropriate data preprocessing steps for the data to see if this desired increase in model accuracy can be achieved.

Data Preparation Steps

- How do I clean data? Data Cleansing
- How do I get the right data? Data Transformation
- How do I include and adjust data? Data Integration
- How do I combine and scale data? Data normalization
- How do I handle missing data? Missing Data Correction
- How do I detect and manage noise? Noise Identification

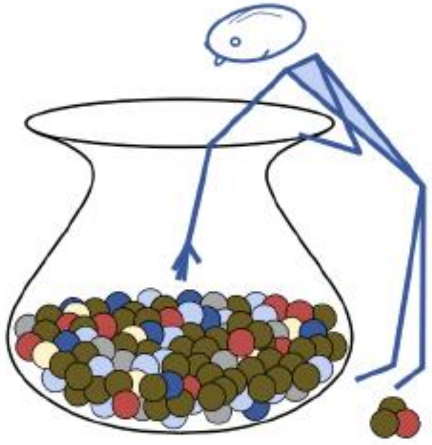
Data Preparation Process

It can also be summarized as the process of preparing data for a machine learning algorithm:

- Data is selected.
- Data is preprocessed.
- Data is transformed.

Data Selection

- There is always a strong desire to include all the data that is available, the adage “more is better” holds true.
- It may or may not be true.
- Think about what data you actually need to address the question or problem you are working on.
- Questions to help you think:
  - What is the scope of the data you have?
  - What data is not available that you would like to have?
  - What data do you not need to solve the problem?



http://uniquerecall.com/

Data Preprocessing

Data Preprocessing

- In supervised learning, anomaly detection is often an important step in data preprocessing to provide the learning algorithm with a suitable dataset to learn from. This is also known as data cleaning. After detecting anomalous examples, classifiers remove them. However, sometimes corrupted data can still provide useful examples for learning. A common method to find suitable examples to use is to identify noisy data. One approach to finding noisy values is to build a probabilistic model from the data using the intact data and corrupted data models.

- In data mining, high-dimensional data will also present high computational challenges with massively large datasets. By removing a large number of examples that may find themselves irrelevant to a classifier or detection algorithm, the runtime can be significantly reduced even on the largest datasets.

Veri Ön İşleme'ye Giriş

- Veri Entegrasyonu
- Veri Dönüşümleri-Veri Ayrıklaştırma-Veri Kodlama
- Veri temizleme
- Veri boyutu Azaltma

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data. e.g., occupation=“ ”
  - noisy: containing errors or outliers. e.g., Salary=“ 10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Data Preprocessing?

- Eksik veriler gelebilir
    - Toplandığında geçerli olmayan veri değerleri
    - Verilerin toplandığı ve analiz edildiği zaman arasında farklılıklar, çelişkiler.
    - İnsan/donanım/yazılım sorunları
- Gürültülü veriler (yanlış değerler) aşağıdakilerden gelebilir:
    - Hatalı veri toplama araçları
    - Veri girişinde insan veya bilgisayar hatası
    - Veri aktarımındaki hatalar
- Tutarsız veriler şunlardan gelebilir:
    - Farklı veri kaynakları
    - İşlevsel bağımlılık ihlali (ör. bazı bağlantılı verileri değiştirin)
- Yinelenen kayıtlar ayrıca veri temizliğine ihtiyaç duyar

Why Is Data Preprocessing Important?

- Kaliteli veri yok ise kaliteli madencilik yok!
  - Kaliteli kararlar kalite verilerine dayanmalıdır. örneğin, mükerrer veya eksik veriler, yanlış ve hattalı verile yanıltıcı istatistiklere neden olabilir.
  - Veri ambarı, kaliteli verilerin tutarlı bir şekilde entegrasyonuna ihtiyaç duyar
- Veri çıkarma, temizleme ve dönüştürme, bir veri ambarı oluşturma işinin çoğunu içerir.

Major Tasks in Data Preprocessing

- Data integration: Integration of multiple databases, data cubes or files
- Data transformation: Normalization and Aggregation. Coding and Binning.
- Data cleaning: Filling in missing values, correcting noisy data, identifying or removing outliers and resolving inconsistencies.
- Data reduction: Achieves a reduced representation in volume but produces the same or similar analytical results

Major Tasks in Data Preprocessing

- Data cleaning
    - Fill in missing values, smooth noisy data, identify or remove outliers and noisy data, and resolve inconsistencies
- Data integration
    - Integration of multiple databases, or files
- Data transformation
    - Normalization and aggregation
- Data reduction
    - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization (for numerical data)

Data Preprocessing

- Data Preprocessing: An Overview
  - Data Quality
  - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization

Data Preprocessing

- A key step in the L2 ingestion layer in the Data Processing Architecture
- Required before running a Machine Learning (ML) algorithm and Analytics
- Required before data is transferred to a data store or cloud service
- Transfer Formats from the data store, analytics application, service or cloud.

Data Preprocessing Needs

- (i) Out-of-range, inconsistent and outlier values
- (ii) Filtering out unreliable, irrelevant and unnecessary information
- (iii) Data cleaning, organizing, reducing and/or discussing
- (iv) Data validation, transformation or transcoding
- (v) ELT processing: Extract, Load and Transform.
- (vi) Enrichment, Organizing, Discussing, Reducing

Daha İyi Veriler > Daha İyi Algoritmalar

- **Biçimlendirme:** Seçilen veriler uygun bir biçimde olmayabilir
- **Temizleme**: Eksik verilerin kaldırılması veya düzeltilmesi
  - Sorunu çözmek için veri taşınmaz ve tamamlanmaz.
  - Hassas bilgiler anonimleştirilir veya kaldırılır.
  - Verilerin eksik, yanlış, hatalı, alakasız kısımları belirlenir.
- **Örnekleme:** Gerekenden daha fazla seçilmiş veri mevcut
  - Algoritmalar için daha uzun çalışma süreleri
  - Daha büyük hesaplama ve bellek gereksinimleri
  - Tüm veri setini değerlendirmeden önce daha küçük temsili örnek alınır.

Rasgele (dummy) Değişkenler

- Kategorik (kuşku bırakmayan, açık, kesin) öznitelik sayısal özniteliğe dönüştürülür.
- Her öznitelik 0 veya 1 değerine sahip olacaktır.
- Tam Rasgele Değişkenler: Her düzey için bir değişken olmak üzere n rasgele değişken kullanarak n kategori temsil edilir.
- Referans Gruplu Rasgele Değişkenler: Kategorik değişken n-1 rasgele değişken kullanarak n kategoriyle temsil edilir.
- Referans Gruplu Sıralı Kategorik Değişken için Rasgele Değişkenler: Matematiksel sıralamalar Küçük < Orta < Büyük olarak varsayılır.
- Sıralamayı belirtmek için daha yüksek kategoriler için daha fazla 1’ler kullanılır.

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, …
  - Consistency: some modified but some not, dangling, …
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

Data Quality

- Data quality is high when it represents the real-world structure that is referenced.
- High quality means data that accurately enables all necessary operations, analysis, decisions, planning, and knowledge discovery.
- A definition for high-quality data, especially for AI applications, might be “data with the five Rs: relevance, timeliness, range, robustness, and reliability.”
- Relevance is extremely important.

Major Tasks in Data Preprocessing

- **Data cleaning**
    - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
    - Integration of multiple databases, data cubes, or files
- **Data reduction**
    - Dimensionality reduction
    - Numerosity reduction
    - Data compression
- **Data transformation and data discretization**
    - Normalization
    - Concept hierarchy generation

Extracting Data

- Finding and removing duplicates in a stack
- Identifying Noisy and Error
- Identifying unnecessary, meaningless data
- Sources of systematic error: Missing data, missing data, bias, unknown, uncertainty,
- Error: Bias or systematic error, random errors, precision, variability.
- Perfect accuracy, precision and specificity are never possible. Biases are usually “unknowns”.

- Confidence intervals are important. Systematic errors (biases) are difficult to detect because they are often not noticed.

Data Integrity

- It refers to maintaining consistency and accuracy in data throughout its usable life.
- Software that stores, processes or retrieves data must maintain the integrity of the data.

Contradiction (Çelişki)

- Outliers – A factor affecting quality
- Refers to data that does not appear to belong to the dataset
- For example, data that is outside the expected range.
- True outliers must be removed from the dataset, otherwise the result will be affected by a small or large amount.
- An outlier, if real, can be useful in detecting anomalies, not due to error.

Missing values

- Missing value – A factor affecting data quality
- Implies data that does not appear in the dataset.

Duplicate Values

- Duplicate value – A factor affecting data quality
- It refers to the same data appearing two or more times in a data set.
- Duplicate values play an important role in determining the manipulation game.
- Frequency, frequency analysis should be performed. Missing data, biased data, anomaly values are determined with FFT.

Deviation – Anomaly Detection in Data Analysis

- In data analysis, anomaly detection (also known as outlier detection) is the identification of rare items, events, or observations that differ significantly from the majority of the data and are suspicious.[1] Typically, anomalous items will amount to some kind of problem, such as bank fraud, a structural defect, medical issues, or errors in a text. Anomalies are also referred to as outliers, novelty, noise, deviations, and exceptions.

- In the context of exploitation and network intrusion detection in particular, interesting objects are often not rare objects but rather unexpected bursts of activity. This pattern does not fit the general statistical definition of an outlier as a rare object, and many outlier detection methods (especially unsupervised methods) fail on such data unless properly clustered. Instead, a cluster analysis algorithm can detect microclusters of these patterns.

- There are three broad categories of anomaly detection techniques:
  - Unsupervised anomaly detection techniques detect anomalies in an unlabeled test dataset by searching for examples that least match the rest of the dataset, assuming that the majority of the examples in the dataset are normal.
  - Supervised anomaly detection techniques require a dataset labeled as “normal” and “abnormal” and involve training a classifier (the key difference from many other statistical classification problems is the inherently unbalanced nature of outlier detection).
  - Semi-supervised anomaly detection techniques generate a model representing normal behavior from a given normal training dataset and then test the probability that a test example is generated by the model used.

Deviation – Anomaly Detection Applications

- It can be applied in various areas such as anomaly detection, intrusion detection, fraud detection, fault detection, system health monitoring, event detection in sensor networks, detection of ecosystem disturbances, and defect detection in images using machine vision.

- It is usually used in preprocessing to remove anomalous data from a dataset.

- In supervised learning, removing anomalous data from a dataset usually results in a statistically significant increase in accuracy.

Mistakes

Intentional errors. Unnoticed systematic errors. Individual errors. Software errors: mathematical modeling, algorithms, coding; incorrect data entry.

- Systematic error: Random, Measurement error, Sampling error.
- Missing Data
- Incomplete Data
- Bias
- Unknown
- Uncertainty
- Sensitivity
- Variability:
- Interference: noise, interference
- Deviation

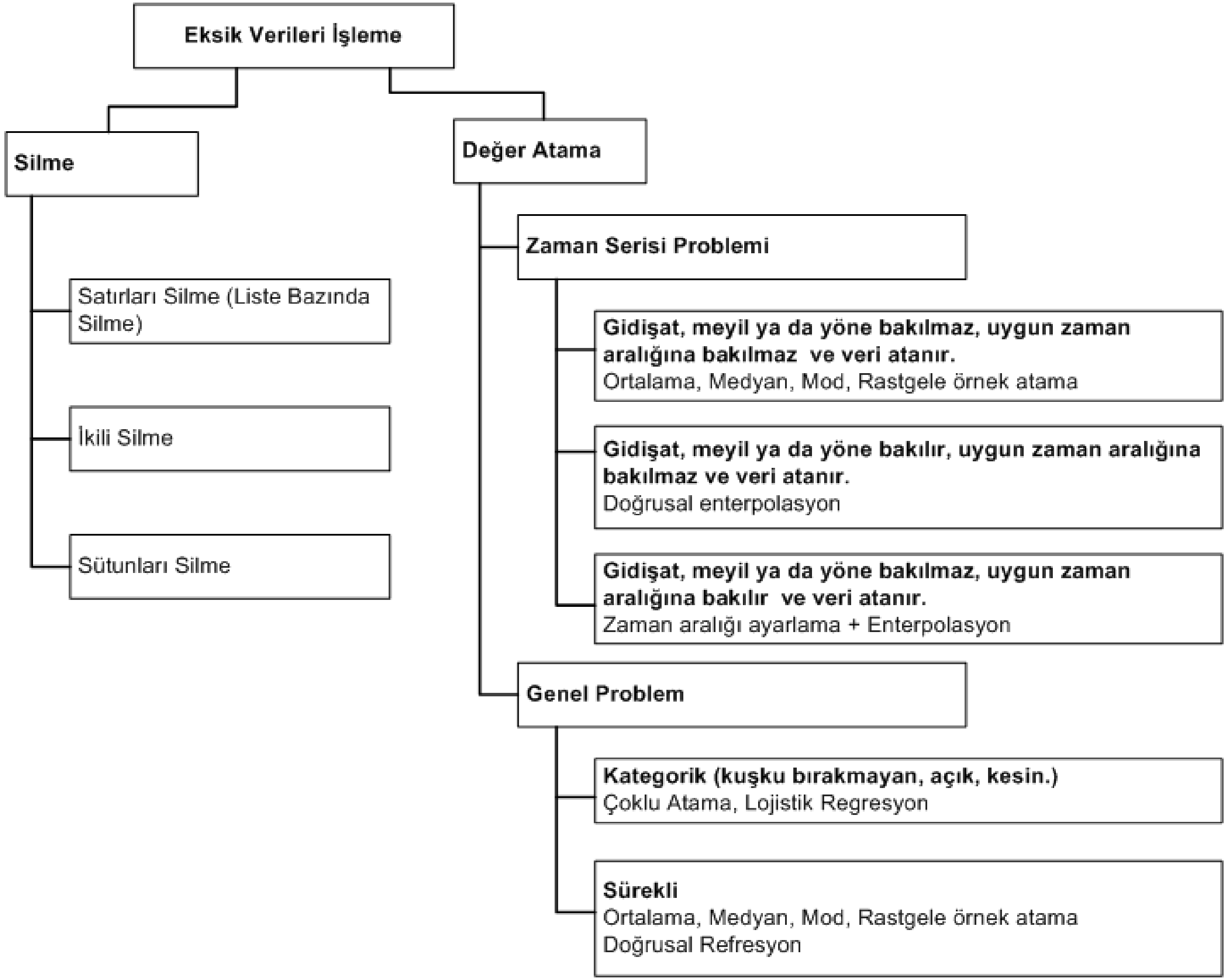
Verileri Ön İşleme

Eksik Veriler Nasıl İşlenir?

- Eksik verilerle başa çıkmanın iyi bir yolu YOKTUR!
- Sorunun türüne bağlı olarak veri ataması için farklı çözümler: Fouier Dönüşümü, Zaman Serisi Analizi, ML, Regresyon vb.
- Genel çözüm yok

Verileri Ön İşleme

Eksik Veriler Nasıl İşlenir?



- **Eksik Verileri İşleme**
  - **Silme**
    - Satırları Silme (Liste Bazında Silme)
    - İkili Silme
    - Sütunları Silme
  - **Değer Atama**
    - **Zaman Serisi Problemi**
      - **Gidişat, meyil ya da yöne bakılmaz, uygun zaman aralığına bakılmaz ve veri atanır.**
        Ortalama, Medyan, Mod, Rastgele örnek atama
      - **Gidişat, meyil ya da yöne bakılır, uygun zaman aralığına bakılmaz ve veri atanır.**
        Doğrusal enterpolasyon
      - **Gidişat, meyil ya da yöne bakılmaz, uygun zaman aralığına bakılır ve veri atanır.**
        Zaman aralığı ayarlama + Enterpolasyon
    - **Genel Problem**
      - **Kategorik (kuşku bırakmayan, açık, kesin.)**
        Çoklu Atama, Lojistik Regresyon
      - **Sürekli**
        Ortalama, Medyan, Mod, Rastgele örnek atama
        Doğrusal Refresyon

Data Cleaning as a Process

- Data discrepancy detection
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check field overloading
  - Check uniqueness rule, consecutive rule and null rule
  - Use commercial tools
    - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
    - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
- Data migration and integration
  - Data migration tools: allow transformations to be specified
  - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
- Integration of the two processes
  - Iterative and interactive (e.g., Potter's Wheels)

Data Preprocessing - Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=“ ” (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*=“−10” (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*=“42”, *Birthday*=“03/07/2010”
    - Was rating “1, 2, 3”, now rating “A, B, C”
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone’s birthday?

Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula or decision tree

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

Noisy

- Noise – It is one of the factors affecting data quality.
- Noise – refers to data that provides additional meaningless information along with the correct (real/necessary) information.

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Preprocessing - Data Integration

- **Data integration**:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

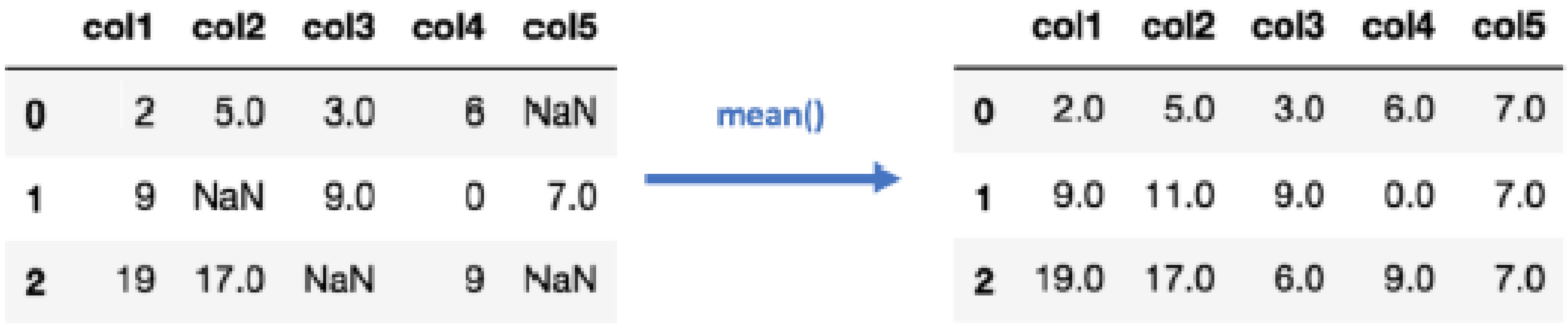
- Redundant data occur often when integration of multiple databases
  - *Object identification*: The same attribute or object may have different names in different databases
  - *Derivable data:* One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Varyans - Standart Sapma

- Aritmetik Ortalama: Alınan örnekleme değerlerinden bir ya da iki tanesi çok yüksek ya da düşük olursa aritmetik ortalama davranışın eğilimini yansıtmaz.
- Varyans - Standart Sapma: Standart sapma, değerlerin aritmetik ortalamasından kaynaklanan kök ortalama karesi (RMS) sapmasıdır. Olasılık ve istatistikte, bir olasılık dağılımının standart sapması, rasgele değişken veya popülasyon veya değerlerin yayılmasının bir ölçüsüdür. Genellikle σ harfi ile belirtilir (küçük harf sigma). Standat sapma, varyansın karekökü olarak tanımlanır. Varyans, veriler ile aritmetik ortalama farklarının karlerinin toplamıdır. Ölçülen verilerin ortalamaya yayılmasını ölçer. Standart sapma, aritmetik ortalamadan olan sapmayı verir.
- Veri değerleri aritmetik ortalamaya yakınsa, standart sapma küçüktür. Ayrıca, birçok veri noktası ortalamanın uzağındaysa, standart sapma büyüktür. Tüm veri değerleri eşitse, standart sapma sıfırdır.
- Bir veri dağılımındaki değişimin önemli bir ölçüsü varyanstır. Varyansın karekökü alınarak standart sapma elde edilir.
- Standart sapma dizideki herbir değerin aritmetik ortalamaya yakınlığını gösterir. Standart sapmanın küçük olması ortalamalarda sapmaların ve riskin az olduğunu, standart sapmanın büyük olması ortalamalarda sapmaların ve riskin çok olduğunu gösterir.

Veri Tahmini/Atama (Ortalama/Medyan) Değerleri

- Bir sütundaki eksik olmayan değerlerin ortalaması/medyanı hesaplanır.
- Artıları:
  - Kolay ve Hızlı
  - Küçük sayısal veri kümeleriyle iyi çalışır
- Eksileri:
  - Özellikler arasındaki korelasyonları etkilemez. Yalnızca sütun düzeyinde çalışır.
  - Kodlanmış kategorik özelliklerde kötü sonuçlar verir (kategorik özelliklerde KULLANMAYIN)
  - çok doğru değil
  - Tahminlerdeki belirsizliği hesaba katmaz

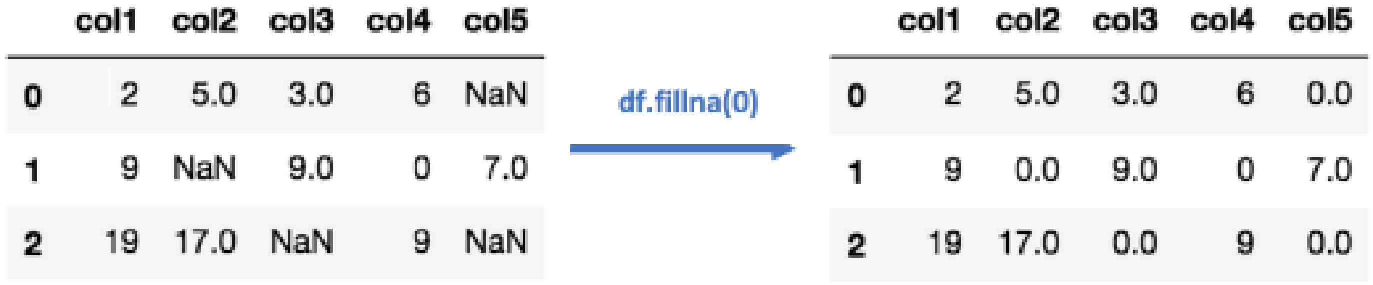


	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

Veri Tahmini/Atama (En Sık) veya (Sıfır/Sabit) Değerler

- Eksik değerleri yüklemek için en sık kullanılan istatistiksel strateji
- Eksik verileri her sütunda en sık görülen değerlerle değiştirme
- Sıfır veya Sabit atama, eksik değerler sıfır veya belirtilen herhangi bir sabit değerle değiştirilir
- Artıları:
  - Kategorik özelliklerle iyi çalışır
- Eksileri:
  - Ayrıca özellikler arasındaki korelasyonları da etkilemez.
  - Verilerde önyargı oluşturabilir



	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

df.fillna(0)

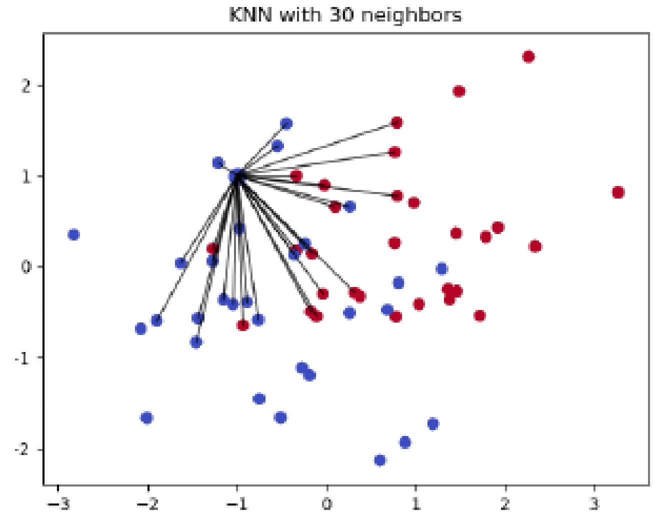
	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	0.0
1	9	0.0	9.0	0	7.0
2	19	17.0	0.0	9	0.0

Veri Tahmini/Atama: k-NN

- K-en yakın komşular basit sınıflandırma için kullanılan bir algoritmadır
- Algoritma, herhangi bir yeni veri noktasının değerlerini tahmin etmek için 'özellik benzerliğini' kullanır
- Yeni noktaya, eğitim kümesindeki noktalara ne kadar benzediğine bağlı olarak bir değer atanır.

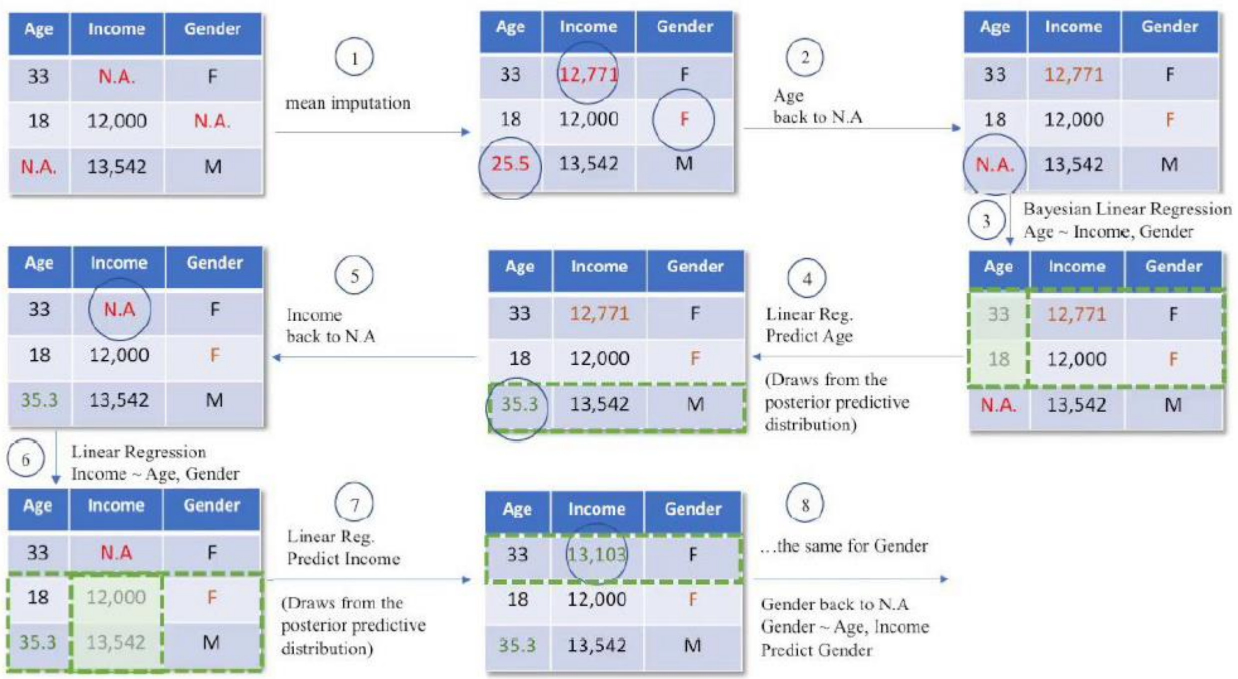
- Artıları:
  - Ortalama, medyan veya en sık kullanılan atama yöntemlerinden çok daha doğru olabilir (Veri kümesine bağlıdır)

- Eksileri:
  - Hesaplamalı olarak pahalı.
  - KNN, tüm eğitim veri setini bellekte saklayarak çalışır.
  - K-NN, verilerdeki aykırı değerlere karşı oldukça hassastır (SVM'den farklı olarak)



Veri Tahmini / Atama: Çok Değişkenli Atama

- Eksik verilerin birden çok kez doldurulması.
- Çoklu atamalarlar , eksik değerlerin belirsizliğini daha iyi bir şekilde ölçtüğü için tek bir atamadan çok daha iyidir.
- Zincirli denklemler yaklaşımı da çok esnektir ve farklı veri tiplerinin farklı değişkenlerini işleyebilir.



Multiple Imputation by Chained Equations (MICE) – Single Iteration

Age	Income	Gender
33	N.A.	F
18	12,000	N.A.
N.A.	13,542	M

1 mean imputation

Age	Income	Gender
33	12,771	F
18	12,000	F
25.5	13,542	M

2 Age back to N.A

Age	Income	Gender
33	12,771	F
18	12,000	F
N.A.	13,542	M

3 Bayesian Linear Regression Age ~ Income, Gender

Age	Income	Gender
33	12,771	F
18	12,000	F
N.A.	13,542	M

4 Linear Reg. Predict Age (Draws from the posterior predictive distribution)

Age	Income	Gender
33	12,771	F
18	12,000	F
35.3	13,542	M

5 Income back to N.A

Age	Income	Gender
33	N.A	F
18	12,000	F
35.3	13,542	M

6 Linear Regression Income ~ Age, Gender

Age	Income	Gender
33	N.A	F
18	12,000	F
35.3	13,542	M

7 Linear Reg. Predict Income (Draws from the posterior predictive distribution)

Age	Income	Gender
33	13,103	F
18	12,000	F
35.3	13,542	M

8 ...the same for Gender

Gender back to N.A Gender ~ Age, Income Predict Gender

Correlation Analysis (Nominal Data)

- **X^2 (chi-square) test**

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the X^2 value, the more likely the variables are related
- The cells that contribute the most to the X^2 value are those whose actual count is very different from the expected count
- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

Chi-Square Calculation: An Example

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- X^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

Correlation Analysis (Numeric Data)

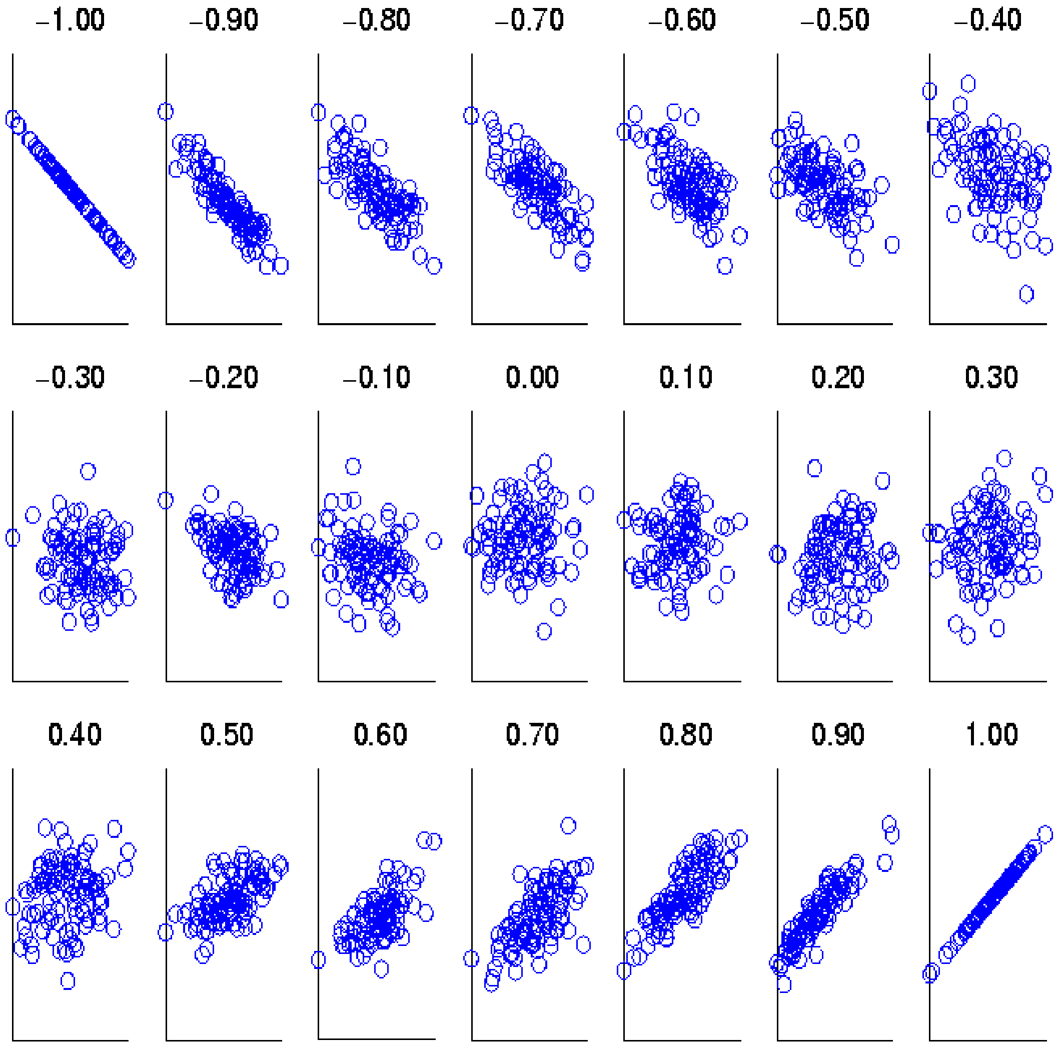
- Correlation coefficient (also called Pearson's product moment coefficient)

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\overline{A}\overline{B}}{(n-1)\sigma_A \sigma_B}$$

where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Correlation (viewed as linear relationship)

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, A and B, and then take their dot product

$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

$$correlation(A, B) = A' \bullet B'$$

Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$$

where n is the number of tuples, and $\overline{A}$ are the $\overline{B}$ respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:
  - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Co-Variance: An Example

$$Cov(A,B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n}$$

- It can be simplified in computation as

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4
  - E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6
  - Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since Cov(A, B) > 0.

Data Preprocessing - Data Reduction Strategies

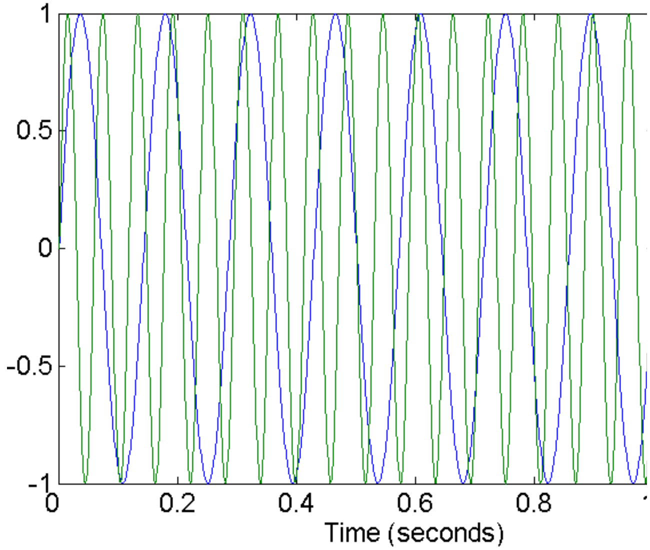
- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - Numerosity reduction (some simply call it: Data Reduction)
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
    - Data cube aggregation
  - Data compression

Data Reduction 1: Dimensionality Reduction

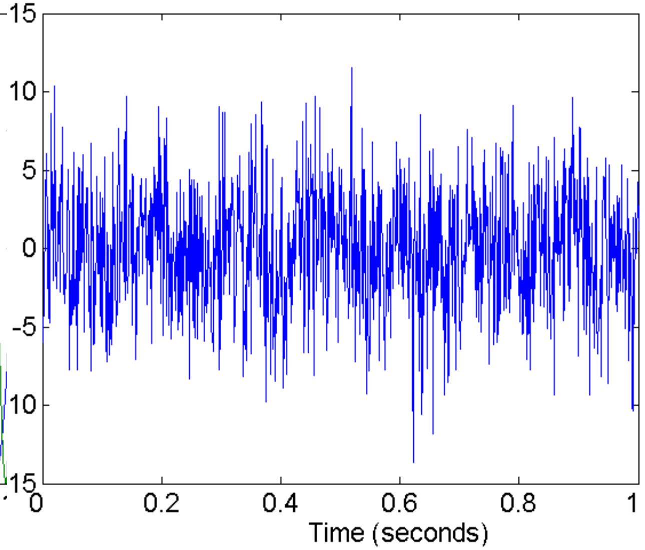
- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

Mapping Data to a New Space

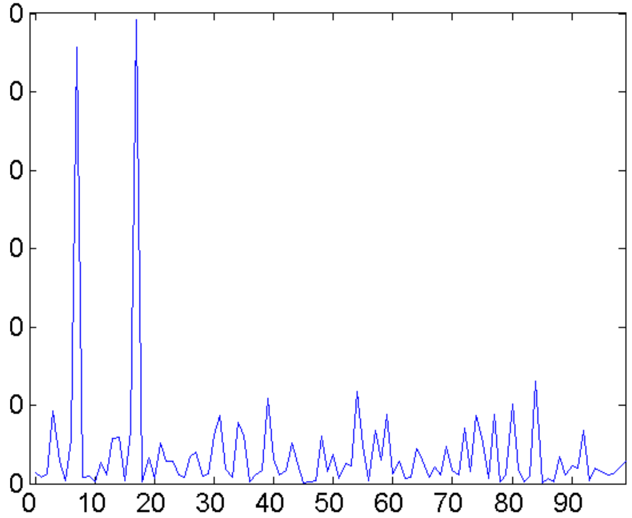
- **Fourier transform**
- **Wavelet transform**



Two Sine Waves



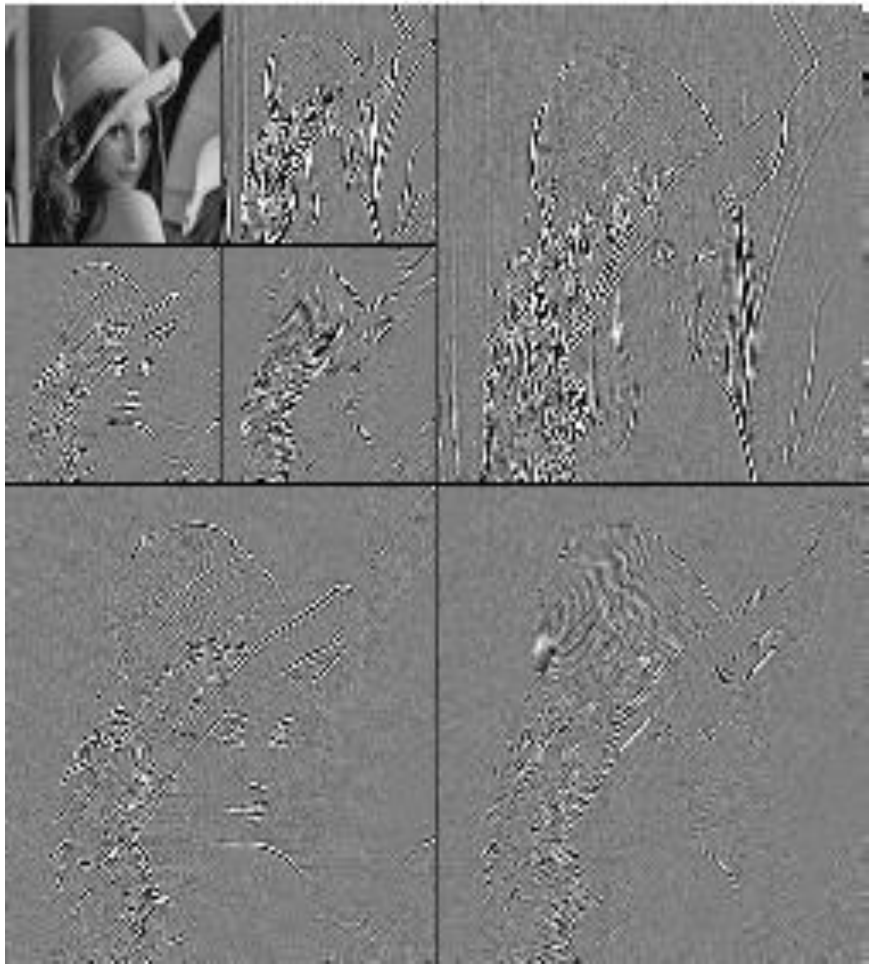
Two Sine Waves + Noise



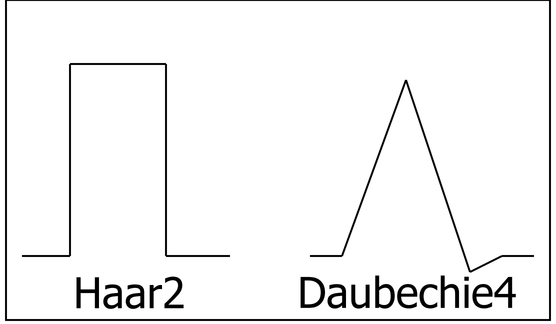
Frequency

What Is Wavelet Transform?

- Decomposes a signal into different frequency subbands
  - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression



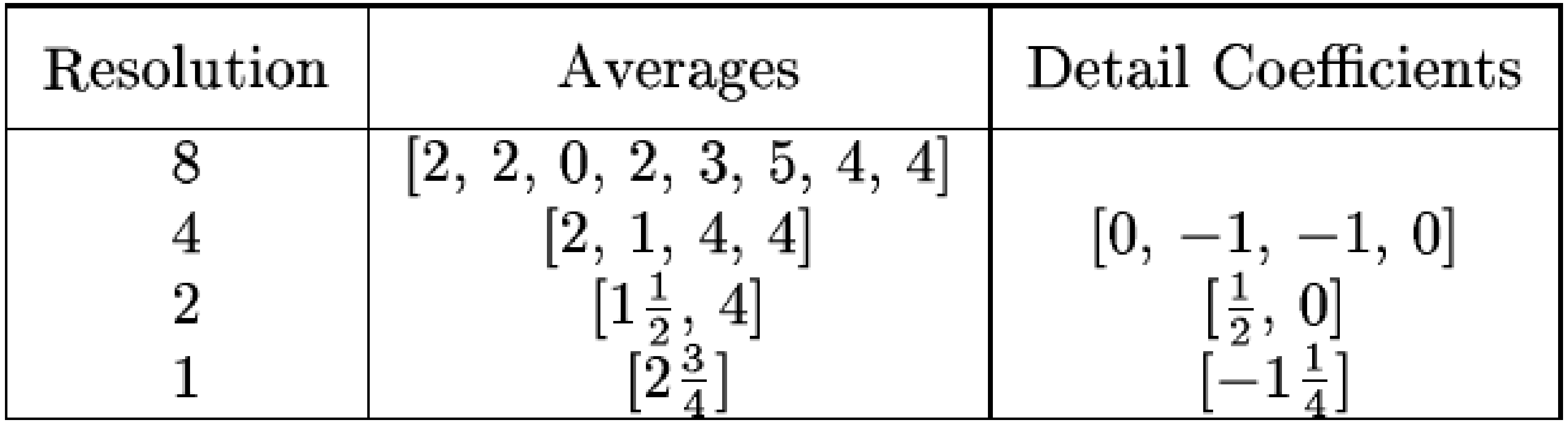
Wavelet Transformation



- Discrete wavelet transform (DWT) for linear signal processing, multi-resolution analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
  - Length, L, must be an integer power of 2 (padding with 0’s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length L/2
  - Applies two functions recursively, until reaches the desired length

Wavelet Decomposition

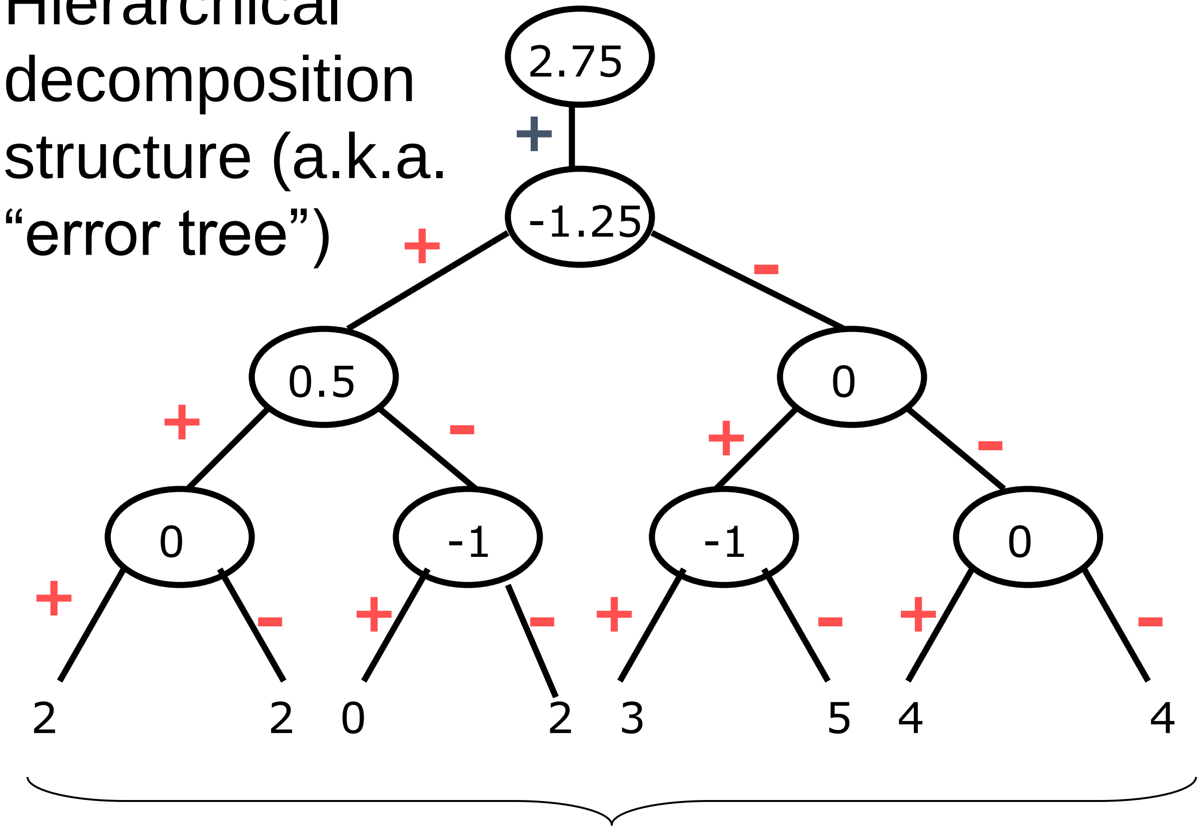
- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- S = [2, 2, 0, 2, 3, 5, 4, 4] can be transformed to $S_{\wedge} = [2^{3}/_{4}, -1^{1}/_{4}, {}^{1}/_{2}, 0, 0, -1, -1, 0]$
- Compression: many small detail coefficients can be replaced by 0's, and only the significant coefficients are retained



Resolution	Averages	Detail Coefficients
8	$[2, 2, 0, 2, 3, 5, 4, 4]$	
4	$[2, 1, 4, 4]$	$[0, -1, -1, 0]$
2	$[1\frac{1}{2}, 4]$	$[\frac{1}{2}, 0]$
1	$[2\frac{3}{4}]$	$[-1\frac{1}{4}]$

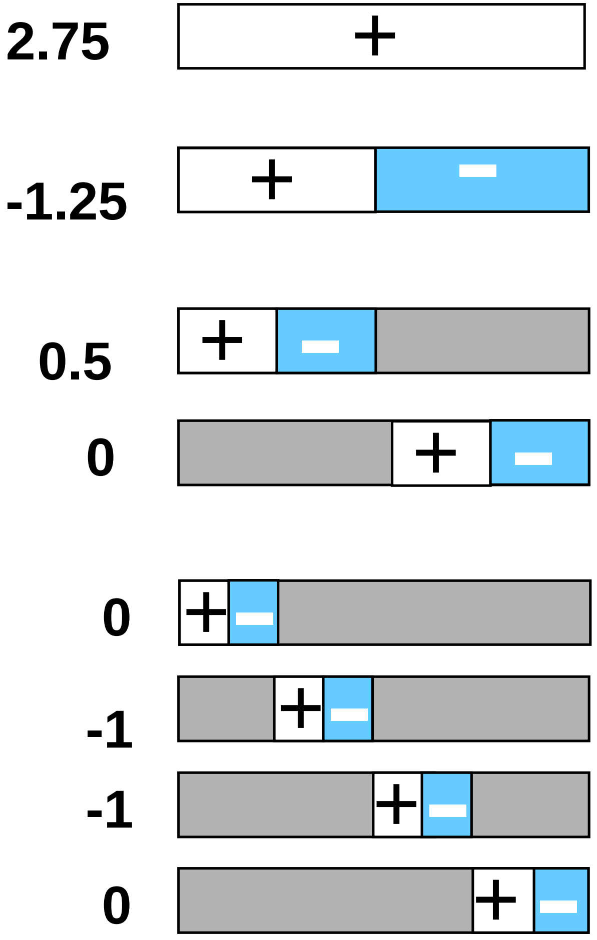
Haar Wavelet Coefficients

Hierarchical decomposition structure (a.k.a. “error tree”)



Original frequency distribution

Coefficient “Supports”

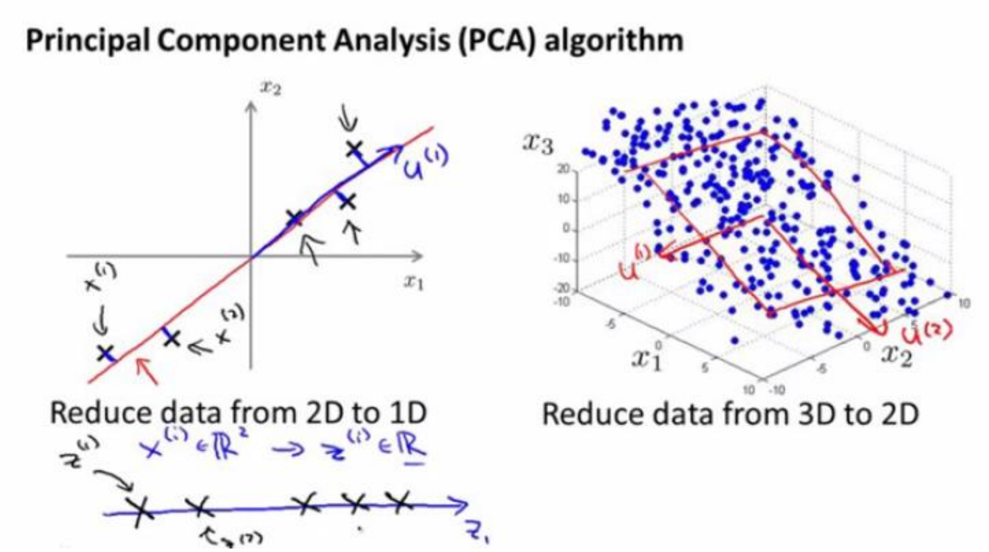


Why Wavelet Transform?

- Use hat-shape filters
  - Emphasize region where points cluster
  - Suppress weaker information in their boundaries
- Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- Multi-resolution
  - Detect arbitrary shaped clusters at different scales
- Efficient
  - Complexity O(N)
- Only applicable to low dimensional data

PCA -nPrincipal Component Analysis

- As the amount of data in the world increases, the size of datasets available for machine learning development also increases
- Dimensionality reduction involves transforming data into new dimensions that make it easier to drop some dimensions without losing any important information.
- Large-scale problems bring with them a variety of dimensions that can be very difficult to visualize.
- Some of these dimensions can be easily reduced for better visualization.

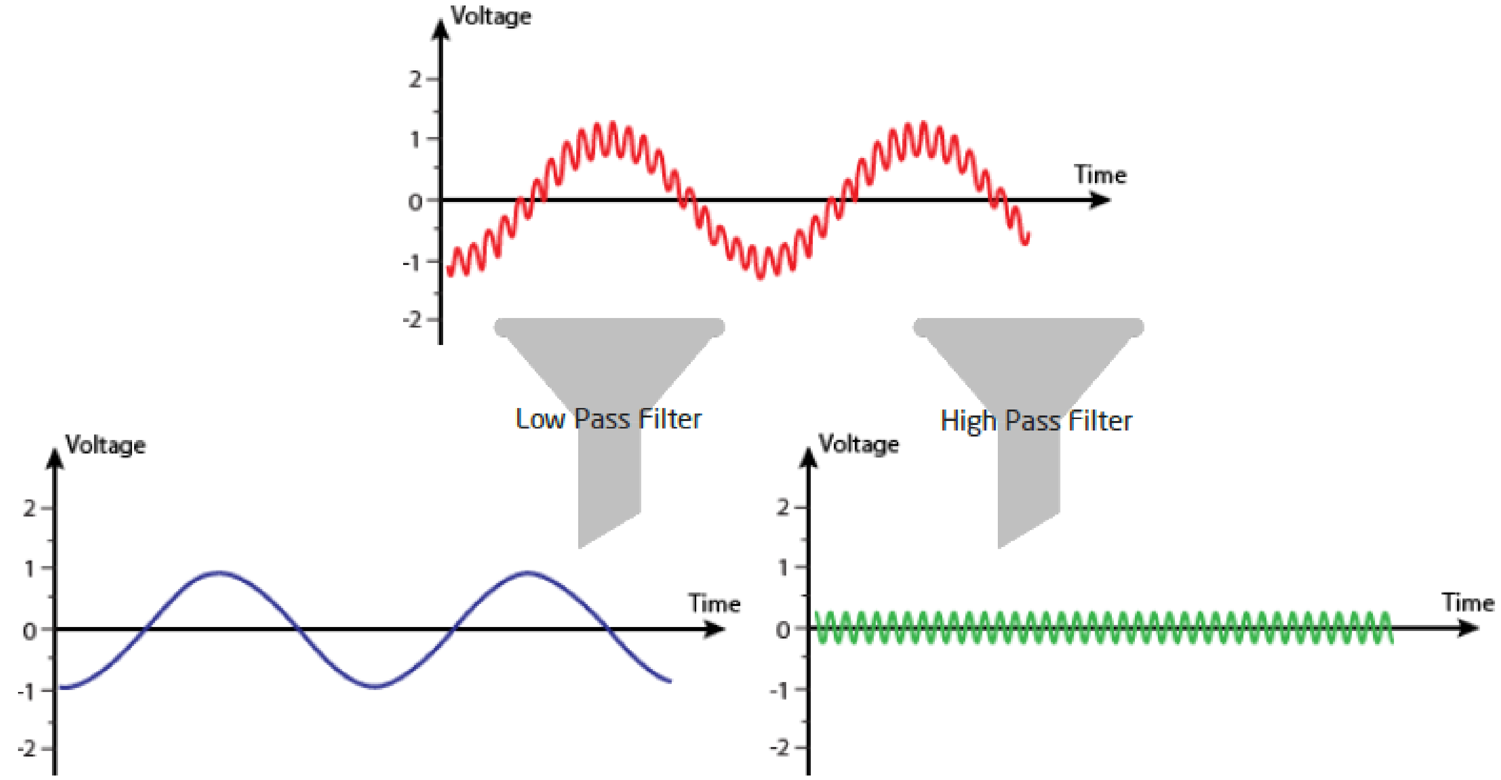


Principal Component Analysis (PCA) algorithm

Reduce data from 2D to 1D

Reduce data from 3D to 2D

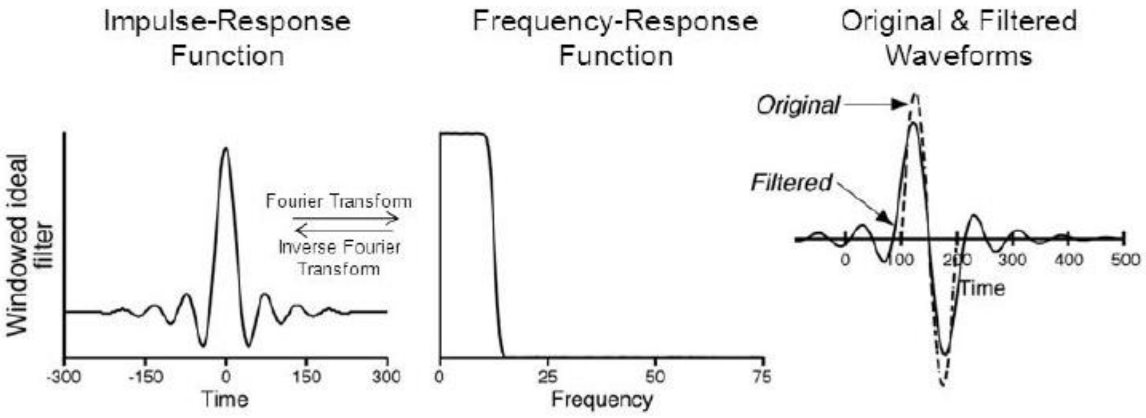
Filter



Changes in the frequency domain are obtained with FFT. For LPF, the upper frequencies are suppressed. For HPF, the lower frequencies are suppressed. With BBF, the signal is allowed to pass in the band gap.

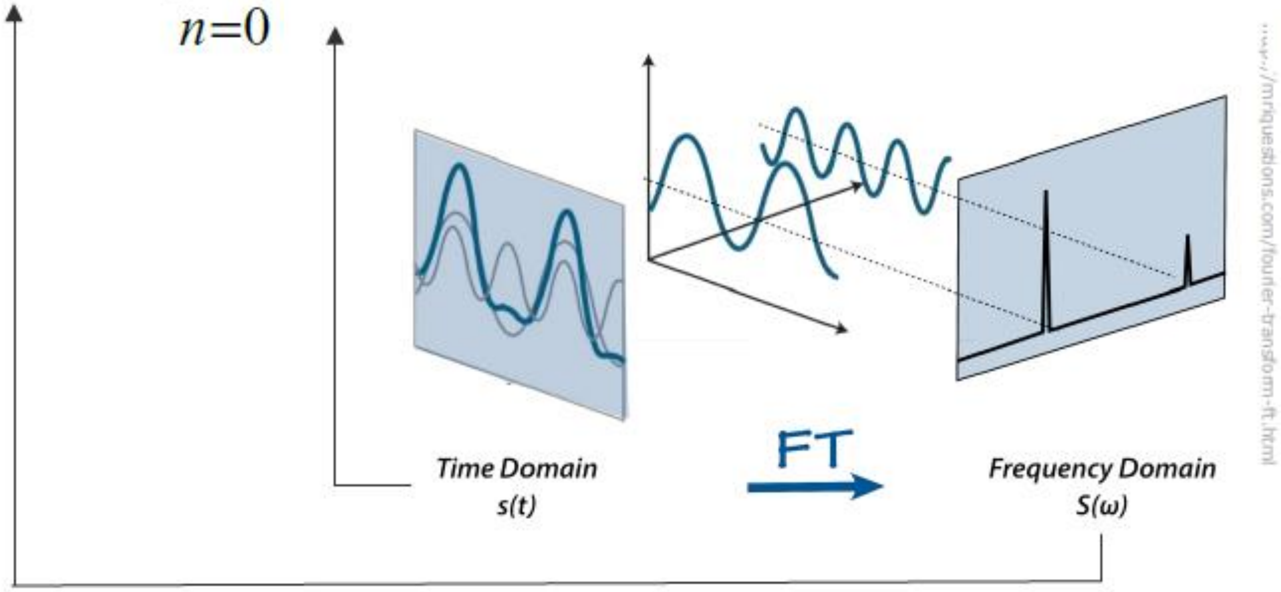
Fourier Transformation

- It is an important signal processing tool.
- It is used to decompose a signal into its sine and cosine components.
- The output of the transform represents the signal in the Fourier or frequency domain.
- Mathematical operations are applied to eliminate certain frequency domains very easily.
- Inverse Fourier transform is applied to recover the original time signal.



Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N} n}$$

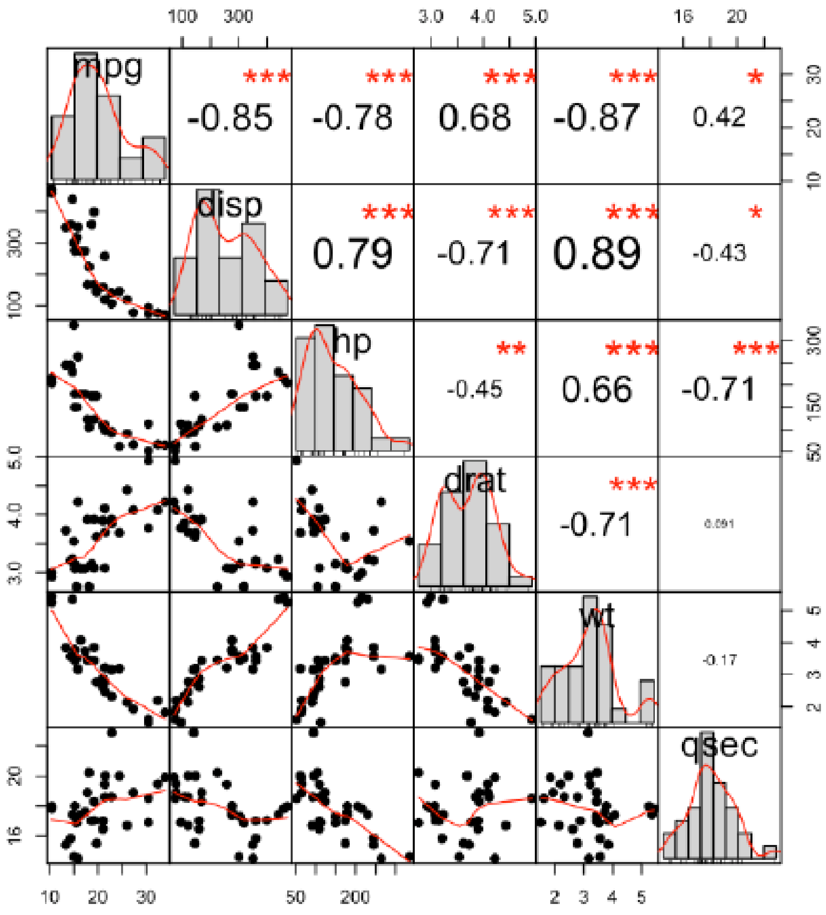


mriquestions.com/fourier-transform-ft.html

Fourier series in 1822

Correlation

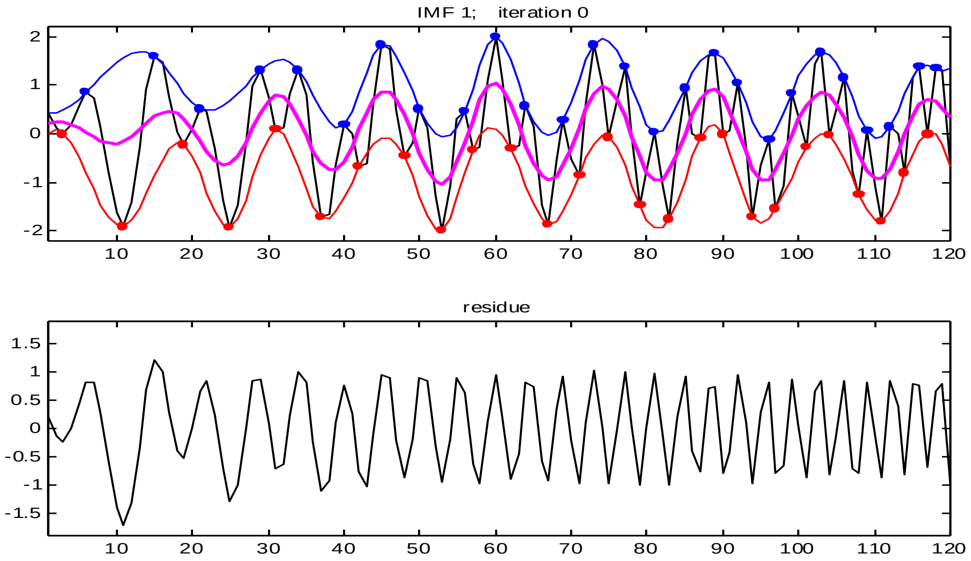
- A way to understand the relationship between multiple variables and attributes in your dataset
- Using correlation, you can gain insights such as:
  - One or more attributes depend on another
  - One or more attributes are correlated with other attributes
- Can help predict one attribute from another (a great way to impute missing values)
- Can (sometimes) indicate the existence of a causal relationship

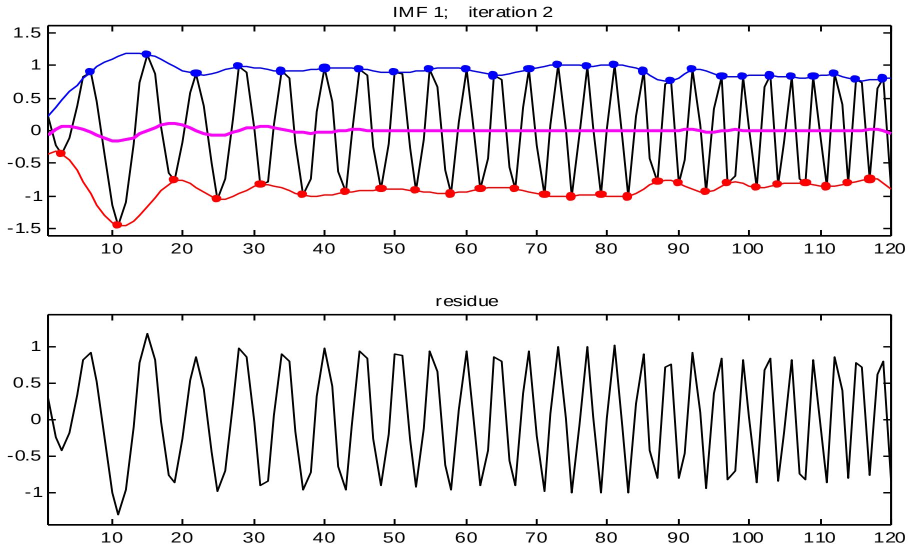


Autocorrelation

- Used extensively in time series analysis and forecasting
- A measure of the relationship between lagged values of a time series
- Uncover hidden patterns in data
- Determine seasonality and trend in our time series data

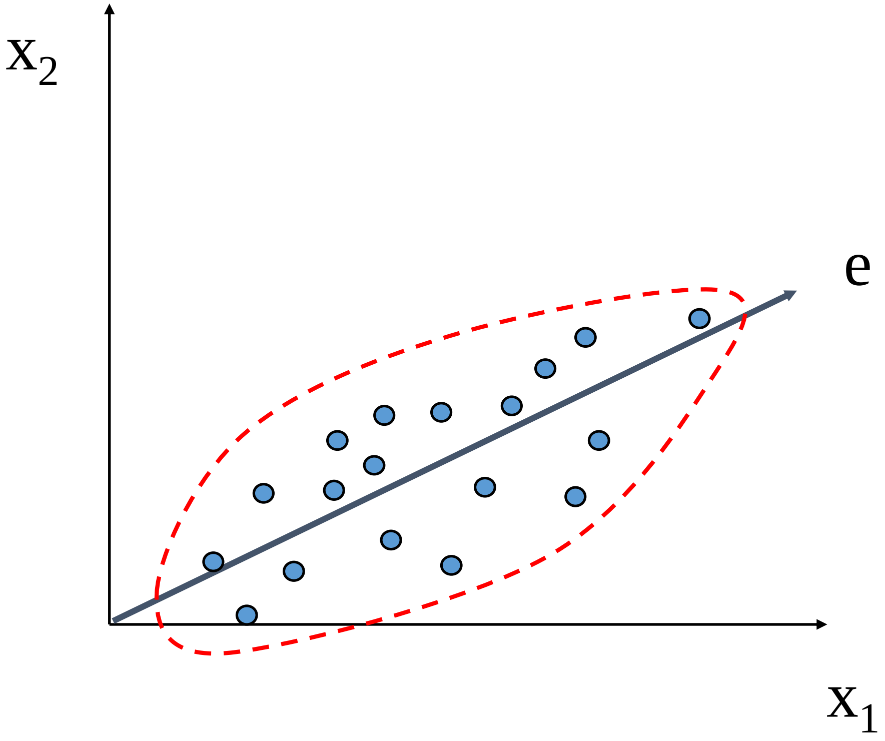
Hilbert Huang Transform





Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Principal Component Analysis (Steps)

- Given *N* data vectors from *n*-dimensions, find *k* ≤ *n* orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the *k* principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

Attribute Subset Selection

- Another way to reduce dimensionality of data
- Redundant attributes
    - Duplicate much or all of the information contained in one or more other attributes
    - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
    - Contain no information that is useful for the data mining task at hand
    - E.g., students' ID is often irrelevant to the task of predicting students' GPA

Heuristic Search in Attribute Selection

- There are *2^d* possible attribute combinations of *d* attributes
- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first
    - Then next best attribute condition to the first, ...
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute
  - Best combined attribute selection and elimination
  - Optimal branch and bound:
    - Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
  - Attribute extraction
    - Domain-specific
  - Mapping data to new space (see: data reduction)
    - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
  - Attribute construction
    - Combining features (see: discriminative frequent patterns in Chapter 7)
    - Data discretization

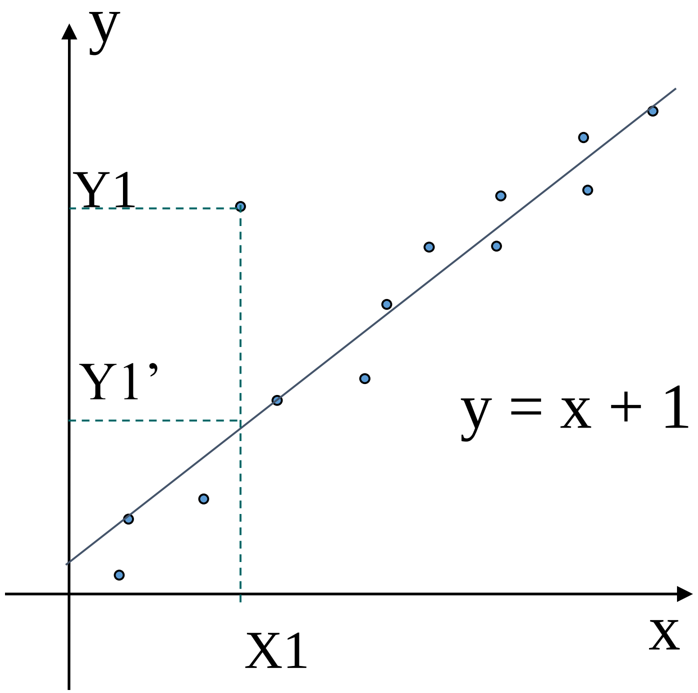
Data Reduction 2: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Ex.: Log-linear models—obtain value at a point in *m*-D space as the product on appropriate marginal subspaces
- **Non-parametric** methods
  - Do not assume models
  - Major families: histograms, clustering, sampling, …

Parametric Data Reduction: Regression and Log-Linear Models

- **Linear regression**
    - Data modeled to fit a straight line
    - Often uses the least-square method to fit the line
- **Multiple regression**
    - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- **Log-linear model**
    - Approximates discrete multidimensional probability distributions

Regression Analysis



- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a ***dependent variable*** (also called ***response variable*** or *measurement*) and of one or more *independent variables* (aka. ***explanatory variables*** or ***predictors***)
- The parameters are estimated so as to give a "**best fit**" of the data
- Most commonly the best fit is evaluated by using the ***least squares method***, but other criteria have also been used

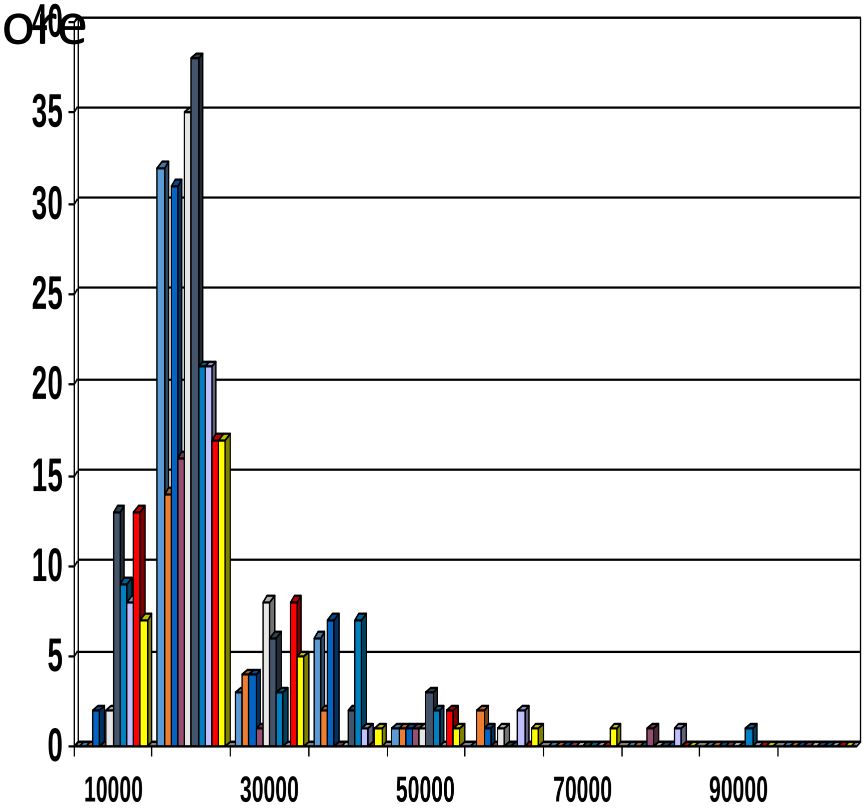
- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

Regress Analysis and Log-Linear Models

- Linear regression: $Y = w X + b$
  - Two regression coefficients, w and b, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots.$
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - Many nonlinear functions can be transformed into the above
- Log-linear models:
  - Approximate discrete multidimensional probability distributions
  - Estimate the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations
  - Useful for dimensionality reduction and data smoothing

Histogram Analysis

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)



Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 10

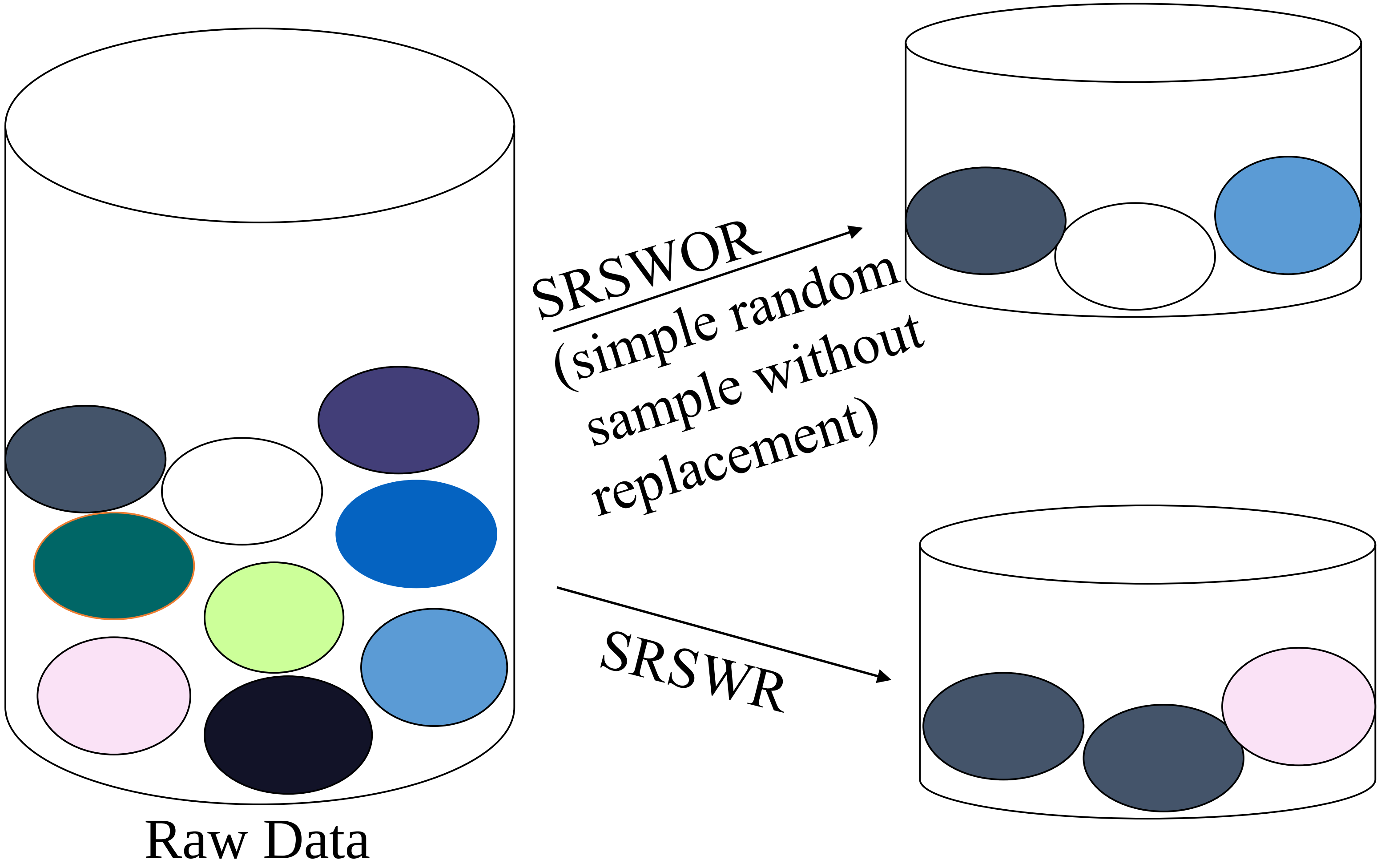
Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling:
- Note: Sampling may not reduce database I/Os (page at a time)

Types of Sampling

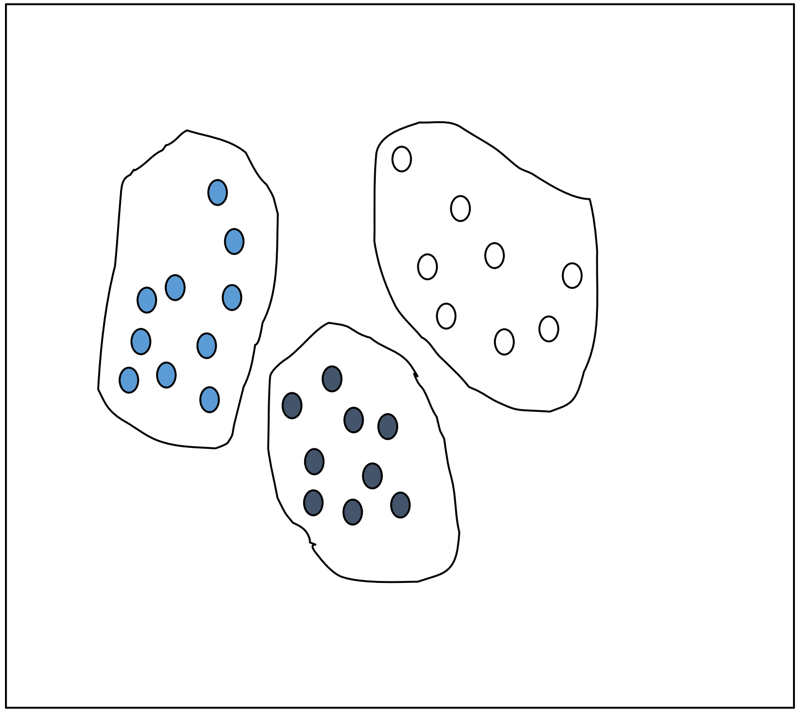
- **Simple random sampling**
  - There is an equal probability of selecting any particular item
- **Sampling without replacement**
  - Once an object is selected, it is removed from the population
- **Sampling with replacement**
  - A selected object is not removed from the population
- **Stratified sampling:**
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
  - Used in conjunction with skewed data

Sampling: With or without Replacement

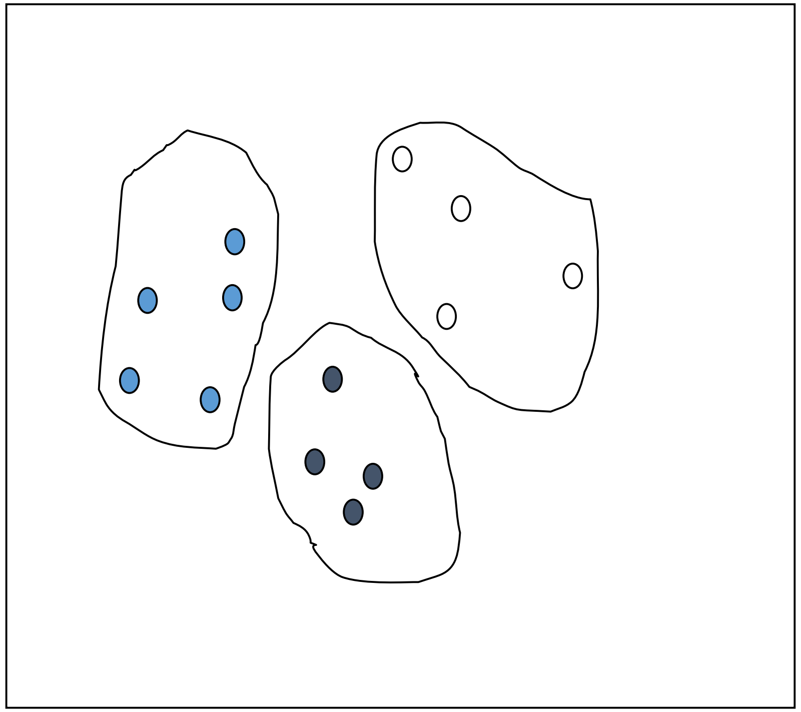


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



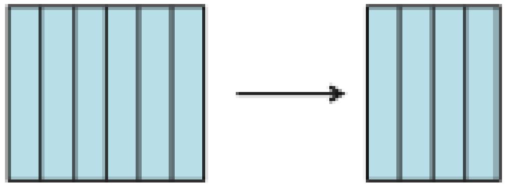
Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

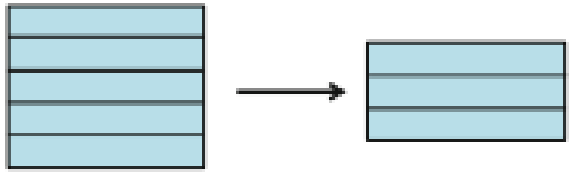
Veri Azaltma

- Verilerin boyutsallığı nasıl azaltılabilir? Özellik Seçimi (Feature Selection - FS)
- Gereksiz ve/veya çelişkili örnekler nasıl kaldırılır? Örnek Seçimi (Instance Selection - IS)
- Bir özniteliğin etki alanı nasıl basitleştirilir? Ayrıklaştırma (Discretization)
- Verilerdeki boşluklar nasıl doldurulur? Özellik Çıkarma ve/veya Örnek Oluşturma (Feature Extraction and/or Instance Generation)

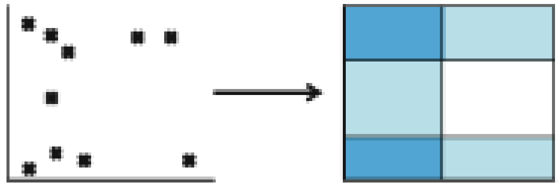
Feature Selection



Instance Selection



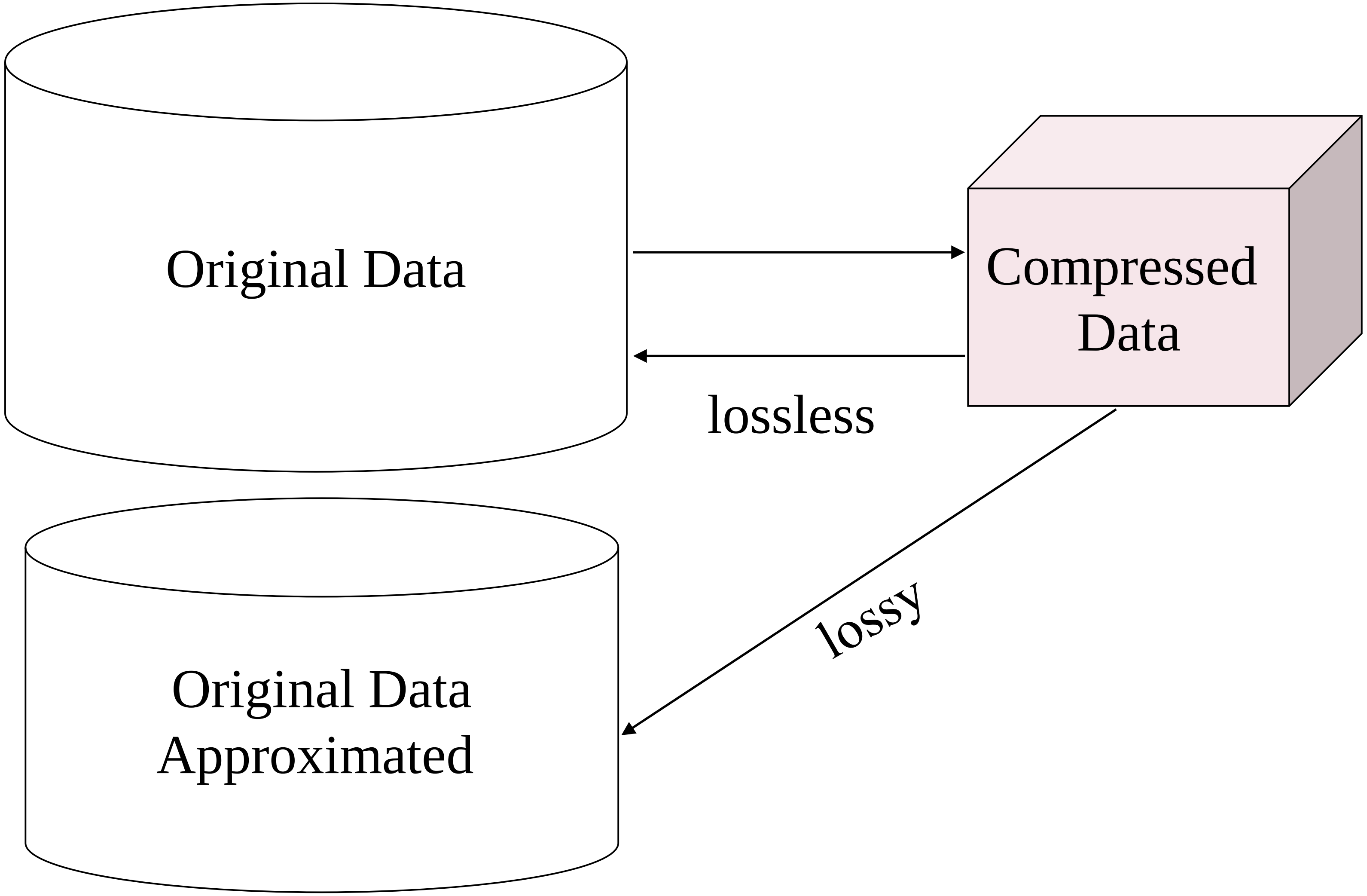
Discretization



Data Reduction: Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
  - Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression

Data Compression



Discretization

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning
    - Top-down split, unsupervised
  - Histogram analysis
    - Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Simple Discretization: Binning

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

Binning Methods for Data Smoothing

❑Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into equal-frequency (**equi-depth**) bins:

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

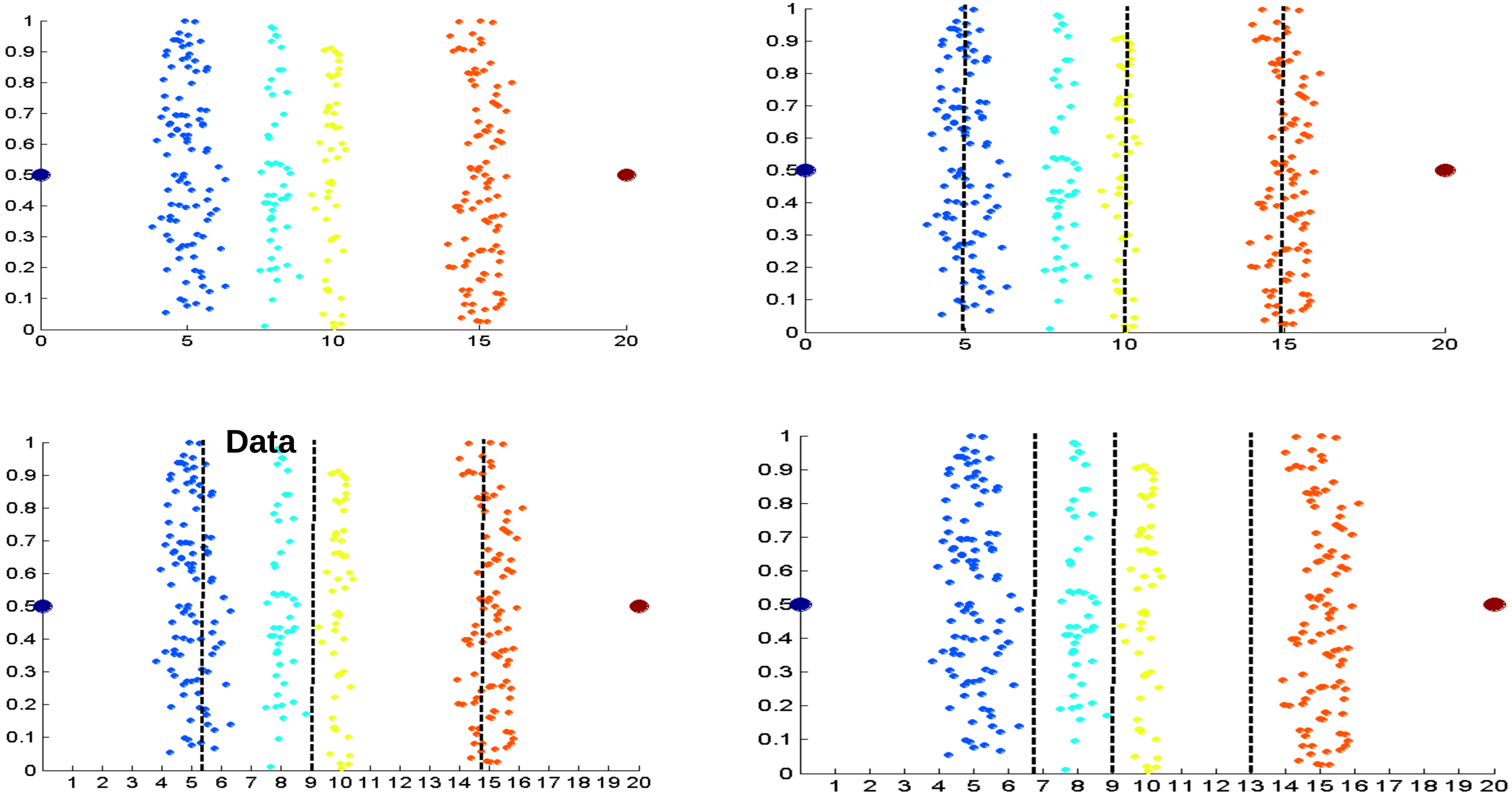
* Smoothing by **bin means**:

- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

* Smoothing by **bin boundaries**:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Discretization Without Using Class Labels (Binning vs. Clustering)



Equal frequency (binning)

K-means clustering leads to better results

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in Chapter 7
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation

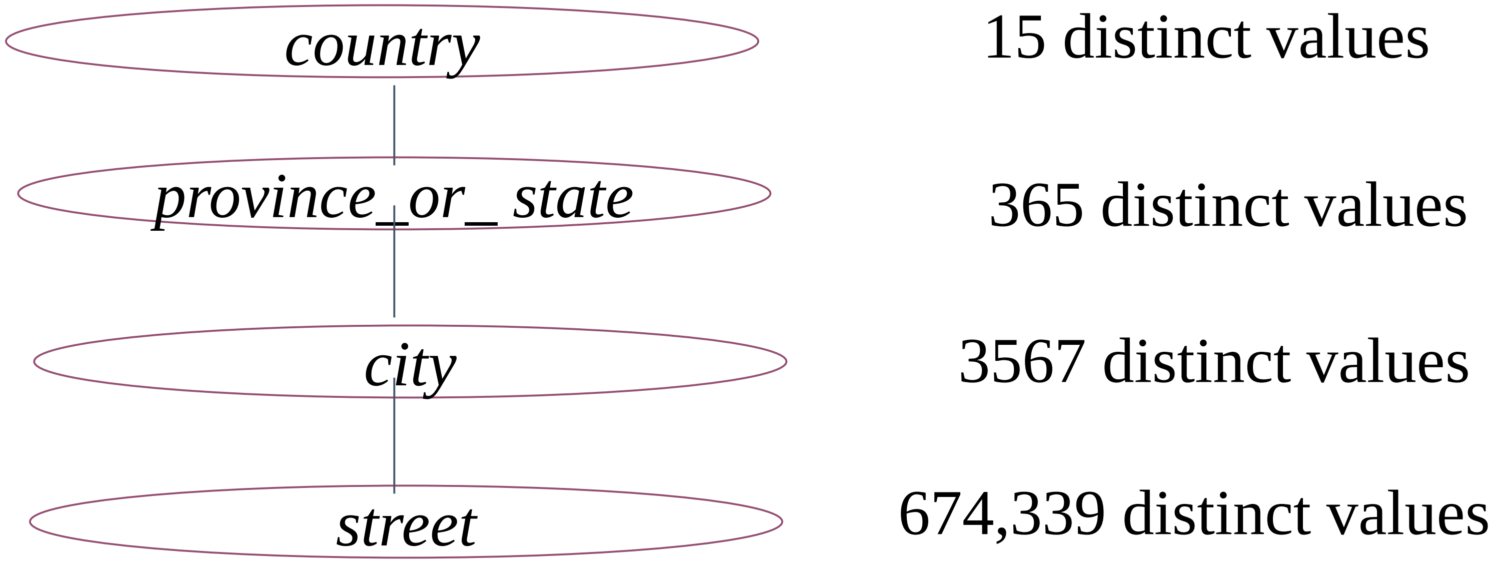
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth, adult*, or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data. For numeric data, use discretization methods shown.

Concept Hierarchy Generation for Nominal Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - *street* < *city* < *state* < *country*
- Specification of a hierarchy for a set of values by explicit data grouping
  - {Urbana, Champaign, Chicago} < Illinois
- Specification of only a partial set of attributes
  - E.g., only *street* < *city*, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - E.g., for a set of attributes: {*street, city, state, country*}

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year



Data Preprocessing - Summary

- **Data quality**: accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning**: e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
  - Entity identification problem
  - Remove redundancies
  - Detect inconsistencies
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

Data Transformation

Data Transformation

- Smoothing: remove noise from data
- Normalization: scaled to fall within a small, specified range
- Attribute/feature construction
  - New attributes constructed from the given ones
- Aggregation: summarization
- Generalization: concept hierarchy climbing

Data Transformation: Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max($|v'|$)<1

Data Preprocessing - Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values
- Methods
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range
    - min-max normalization
    - z-score normalization
    - normalization by decimal scaling
  - Discretization: Concept hierarchy climbing

Normalization

- **Min-max normalization**: to [new_min$_A$, new_max$_A$]

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ: mean, σ: standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

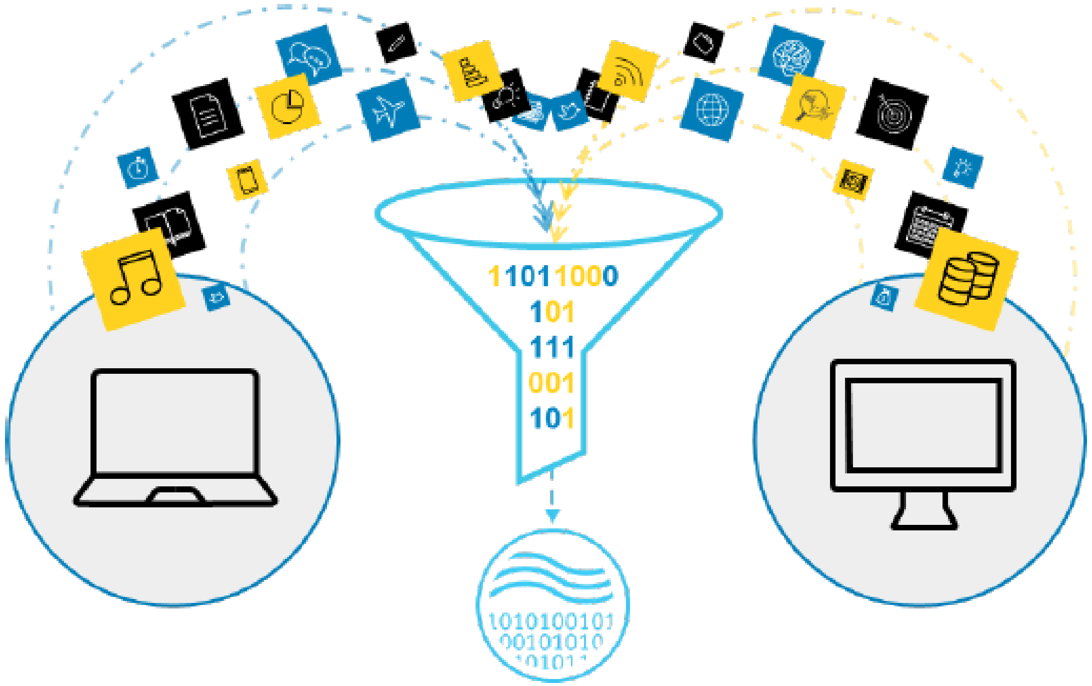
  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(|ν'|) < 1

Transform Data

- Scaling: The preprocessed data may contain attributes with a mixtures of scales for various quantities. Many machine learning methods like data attributes to have the same scale
- Decomposition: There may be features that represent a complex concept that may be more useful to a machine learning method when split into the constituent parts, Example -> Date
- Aggregation: There may be features that can be aggregated into a single feature

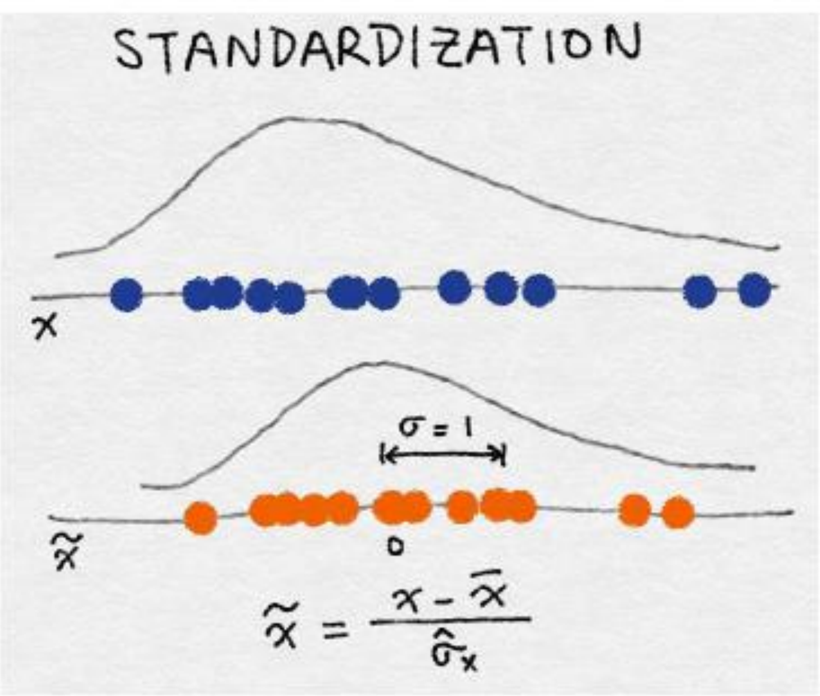


Transform Data

Standardization (Variance Scaling)

- Özelliğin ortalamasını (tüm veri noktalarından) çıkarır ve varyansa böler
- Ayrıca varyans ölçekleme olarak da adlandırılabilir, sonuçta ölçeklenen özelliğin ortalaması 0 ve varyansı 1'dir.
- Orijinal özelliğin bir Gauss dağılımı varsa, ölçeklenen özellik de öyledir

$$\tilde{x} = \frac{x - \mathrm{mean}(x)}{\mathrm{sqrt}(\mathrm{var}(x))}$$

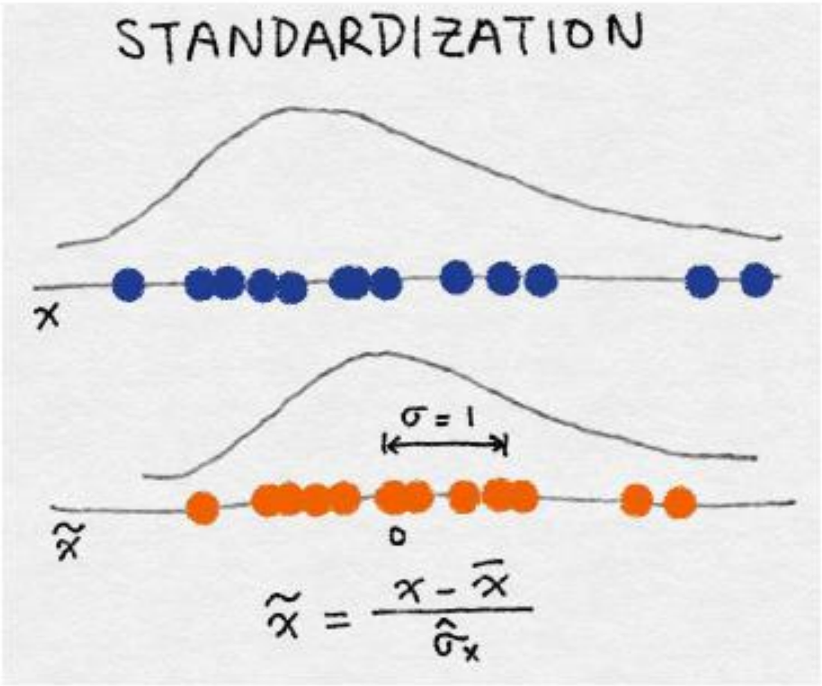


Transform Data

Min-Max Scaling

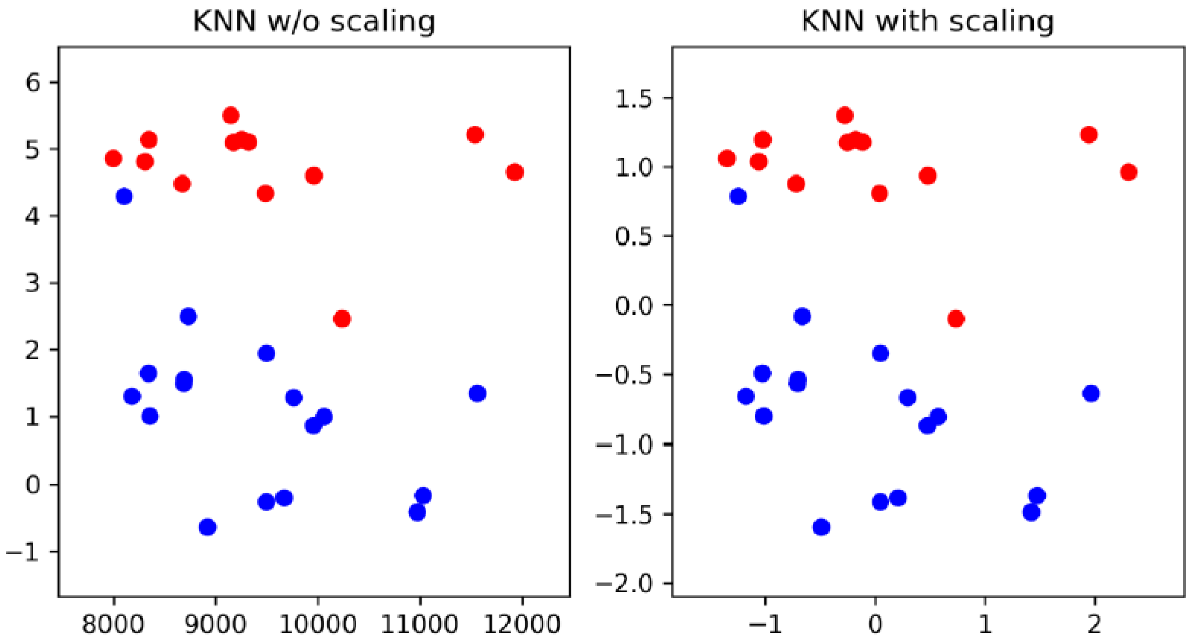
- Let x be an individual feature value (i.e., a value of the feature in some data point)
- min (x) and max (x), respectively, be the minimum and maximum values of this feature over the entire dataset
- Min-max scaling squeezes (or stretches) all feature values to be within the range of [0, 1]

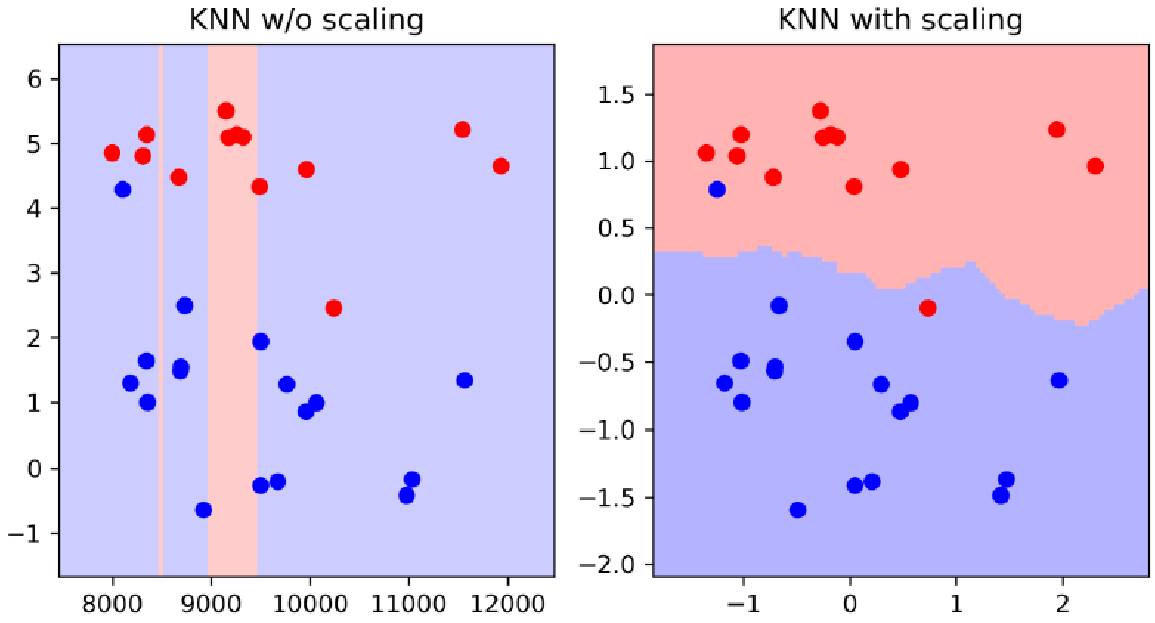
$$\tilde{x} = \frac{x - \text{mean}(x)}{\text{sqrt}(\text{var}(x))}$$



Transform Data

Why Scaling?

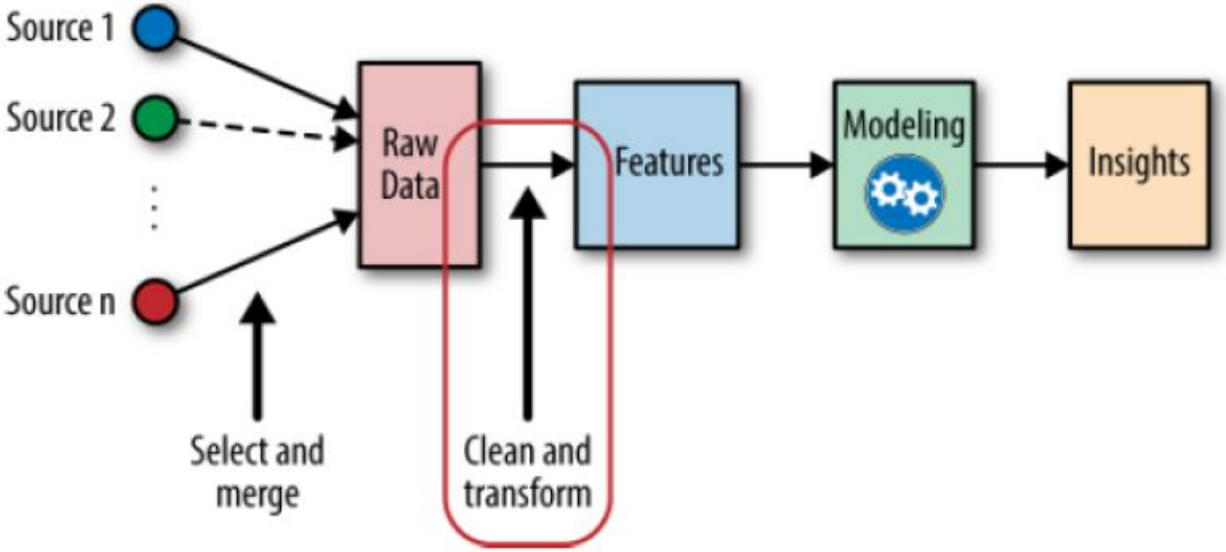




Transform Data

Feature Engineering

- Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.
- The features you use influence more than everything else the result. No algorithm alone, to my knowledge, can supplement the information gain given by correct feature engineering.
- Good data preparation and feature engineering is integral to better prediction.



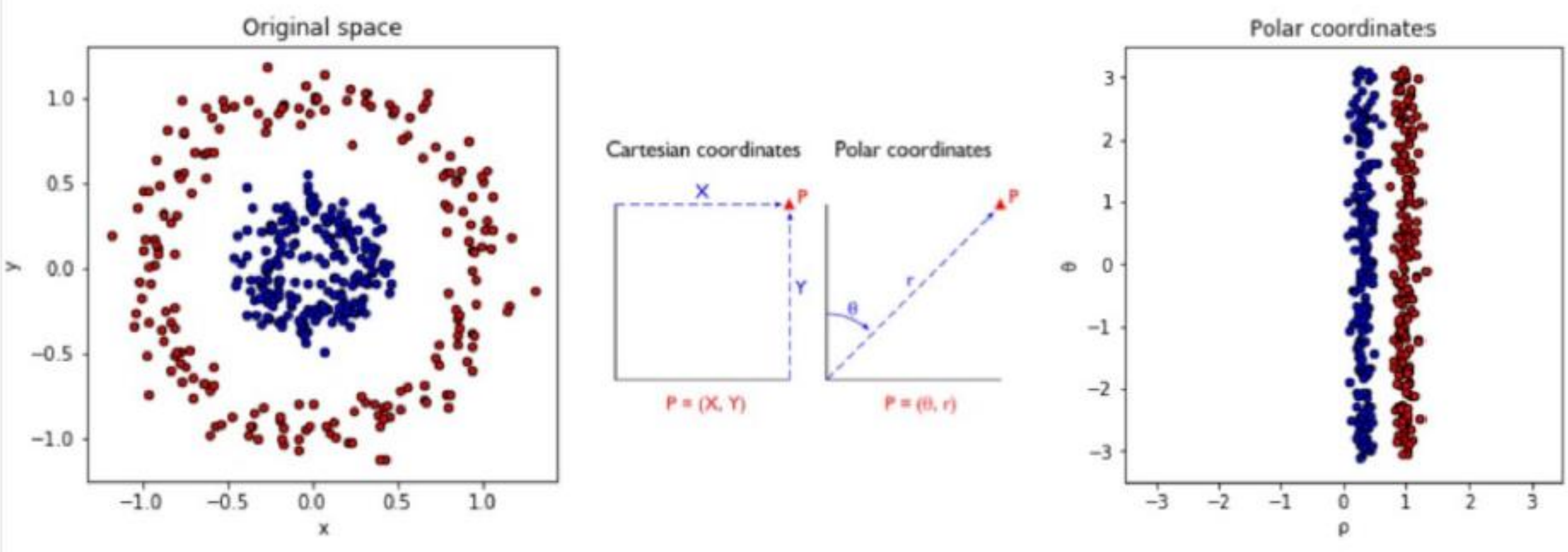
Transform Data

Feature Engineering

- Verilerin anlaşılması ve işlenmesi zor olabilir
- Makine öğrenimi modellerimiz için verilerin okunmasını kolaylaştırmak için özellik mühendisliği yapılır
- Özellik Mühendisliği, verilen verileri yorumlanması daha kolay bir forma dönüştürme işlemidir.
- Genel olarak: Veri ile ilgili arka planı olmayan kişiler için hazırlanan veri görselleştirmesinin daha sindirilebilir olması için özellikler oluşturulabilir.
- Farklı modeller genellikle farklı veri türleri için farklı yaklaşımlar gerektirir.

Transform Data

Feature Engineering Example: Coordinate Transformation



Transform Data

Özellik Mühendisliğinin Yinelemeli Süreci

- Beyin fırtınası özellikleri: Gerçekten problemin içine girilir, birçok veriye bakılır, diğer problemler üzerinde özellik mühendisliği incelenir ve neler alınabileceği görülür.
- Özellikler geliştir: Sorununa bağlıdır, ancak otomatik özellik çıkarma, manuel özellik oluşturma ve ikisinin karışımı kullanılabilir.
- Özelliklerin seçilmesi: Modellerinizin üzerinde çalışacağı bir veya daha fazla "görünüm" hazırlamak için farklı özellik önem puanlamalarını ve özellik seçim yöntemlerini kullanılır.
- Modellerin değerlendirilmesi: Seçilen özellikleri kullanarak görünmeyen verilerde model doğruluğu tahmin edilir.

Transform Data

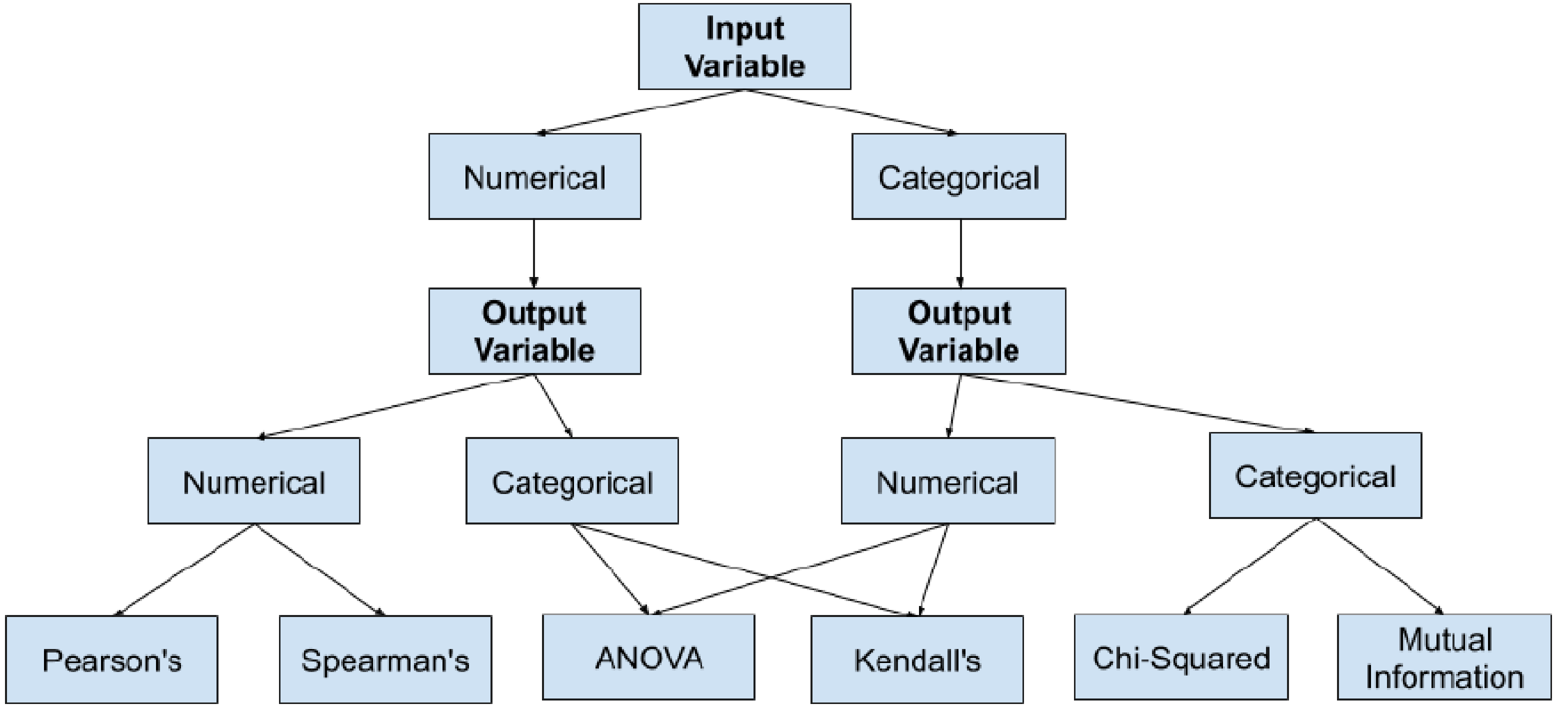
Özellik Mühendisliğinin Yönleri

- Özellik Seçimi: En kullanışlı ve ilgili özellikler mevcut verilerden seçilir.
- Özellik Çıkarma: Mevcut özellikler, daha kullanışlı özellikler geliştirmek için birleştirilir.
- Özellik Ekleme: Yeni veriler toplanarak yeni özellikler oluşturulur.
- Özellik Filtreleme: Modelleme adımını kolaylaştırmak için alakasız özellikler filtrelenir.

Transform Data

Öznitelik Seçimi

- Verilerde, ilgilenilen tahmin değişkenine veya çıktıya en çok katkıda bulunan özelliklerin otomatik olarak seçildiği süreç.
- Verilerde alakasız özelliklerin olması, birçok modelin, özellikle lineer ve lojistik regresyon gibi lineer algoritmaların doğruluğunu azaltabilir.



Data Aggregation and Boosting

Motivation

- Many models can be trained on the same data
- Typically none is strictly better than others
    - Recall “no free lunch theorem”
- Can we “combine” predictions from multiple models?

- Yes, typically with significant reduction of error!

Motivation

- Combined prediction using Adaptive Basis Functions

$$f(x) = \sum_{i=1}^{M} w_m \phi_m(x; v_m)$$

- M basis functions with own parameters
- Weight / confidence of each basis function
- Parameters including M trained using data

- Another interpretation: automatically learning best representation of data for the task at hand

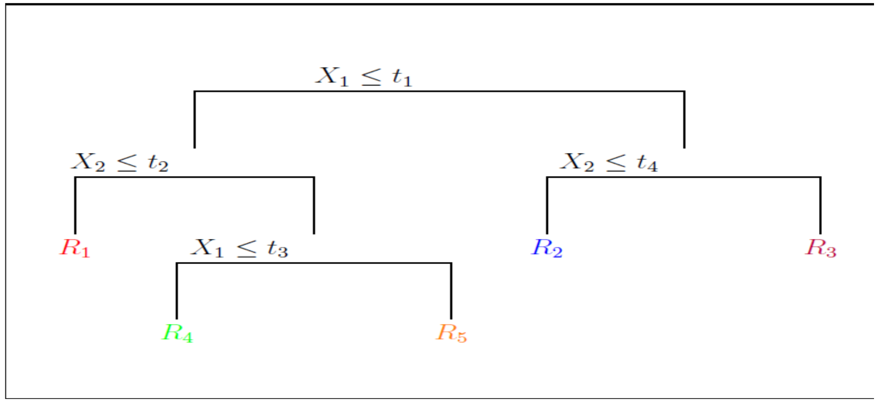
- Difference with mixture models?

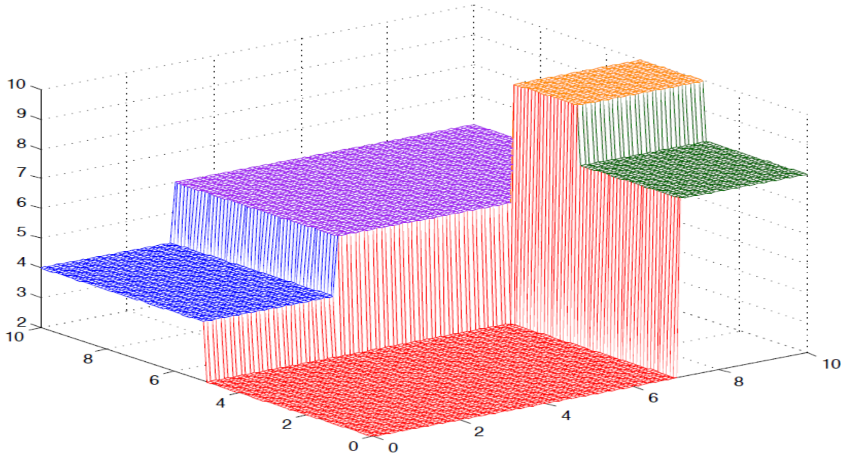
Examples of Model Combinations

- Also called Ensemble Learning
- Decision Trees
- Bagging
- Boosting
- Committee / Mixture of Experts
- Feed forward neural nets / Multi-layer perceptrons
- …

Decision Trees

- Partition input space into cuboid regions
- Simple model for each region
  - Classification: Single label; Regression: Constant real value
- Sequential process to choose model per instance
  - Decision tree





Learning Decision Trees

- Decision for each region
  - Regression: Average of training data for the region
  - Classification: Most likely label in the region
- Learning tree structure and splitting values
  - Learning optimal tree intractable
- Greedy algorithm
  - Find (node, dim., value) w/ largest reduction of "error"
    - Regression error: residual sum of squares
    - Classification: Misclassification error, entropy, …
  - Stopping condition
- Preventing overfitting: Pruning using cross validation

Pros and Cons of Decision Trees

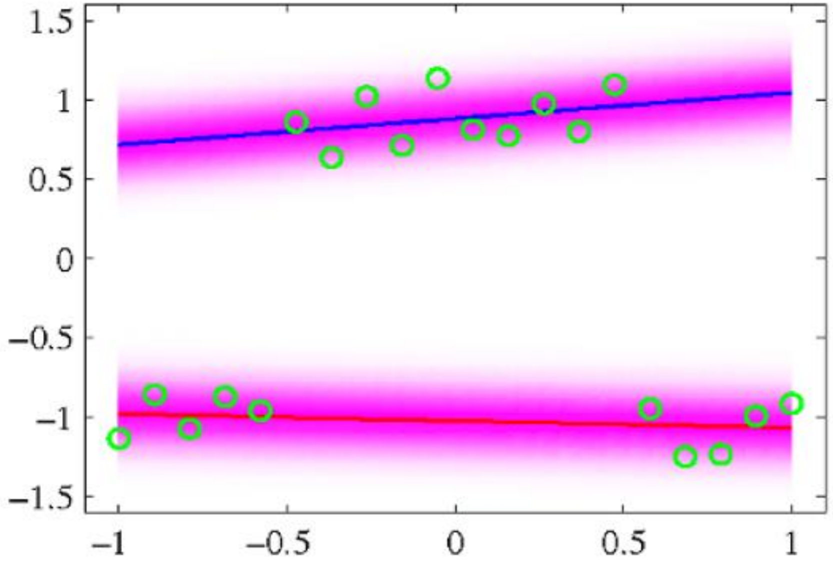
- Easily interpretable decision process
    - Widely used in practice, e.g. medical diagnosis

- Not very good performance
    - Restricted partition of space
    - Restricted to choose one model per instance
    - Unstable

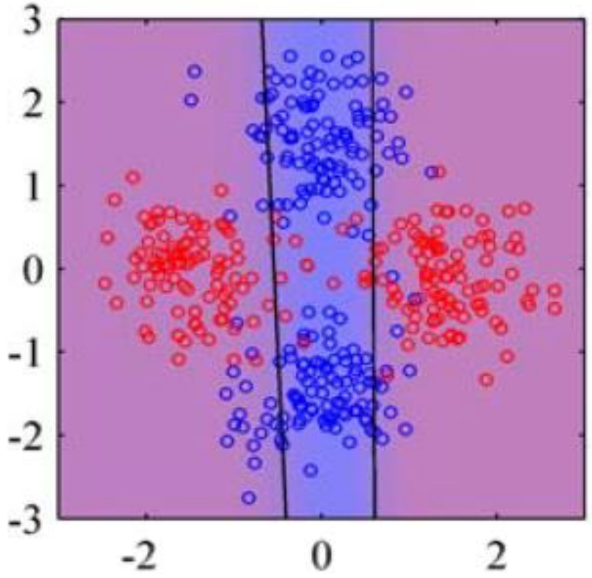
Mixture of Supervised Models

$$f(x) = \sum_{i} \pi_k \phi_k(x, w)$$

Mixture of linear regression models



Mixture of logistic regression models

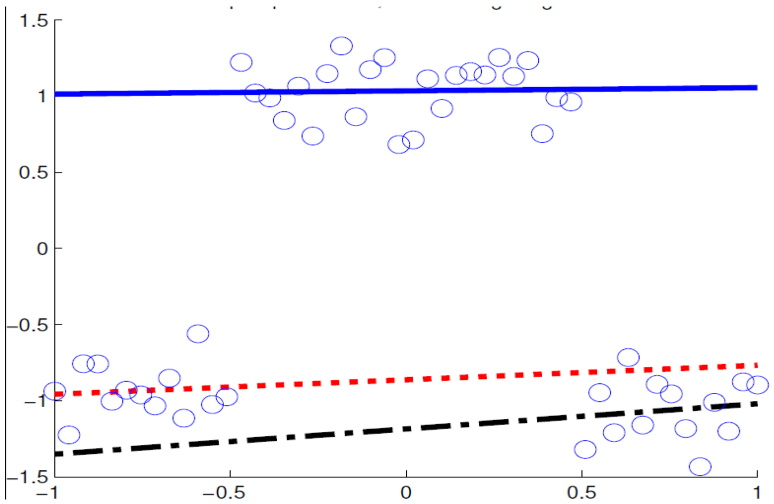


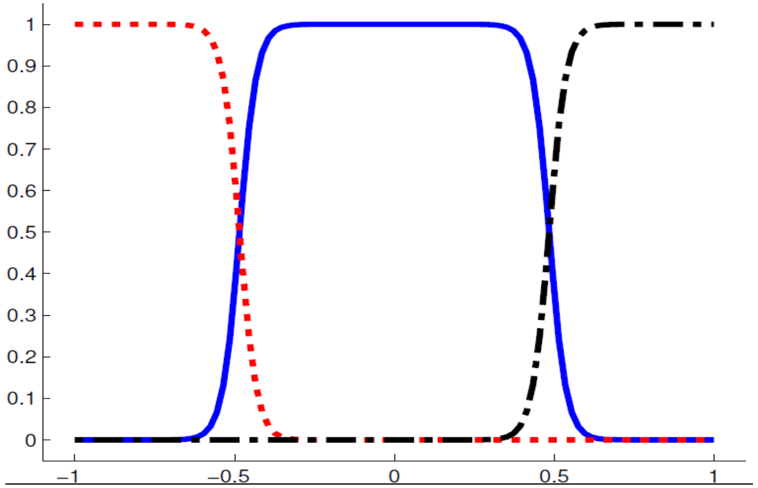
- Training using EM

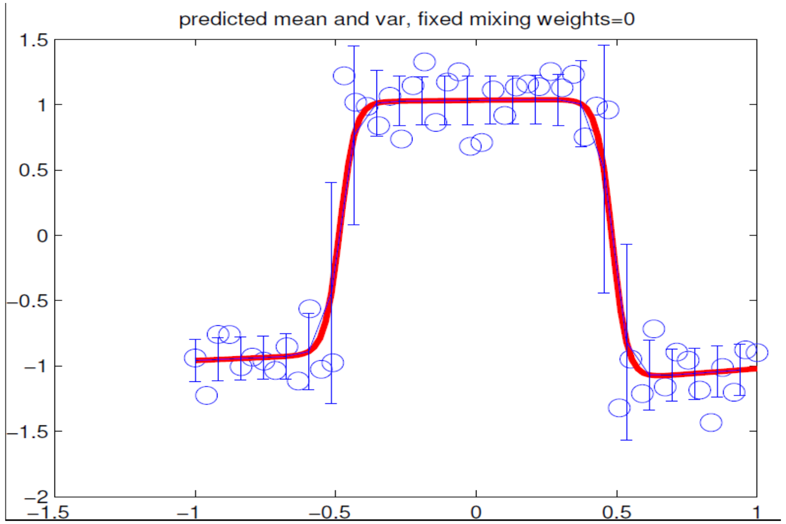
Conditional Mixture of Supervised Models

- Mixture of experts

$$f(x) = \sum_{i} \pi_k(x)\phi_k(x, w)$$







Bootstrap Aggregation / Bagging

- Individual models (e.g. decision trees) may have high variance along with low bias

- Construct M bootstrap datasets
- Train separate copy of predictive model on each
- Average prediction over copies

$$f(x) = \frac{1}{M}\sum_{i} f_m(x)$$

- If the errors are uncorrelated, then bagged error reduces linearly with M

Random Forests

- Training same algorithm on bootstraps creates correlated errors

- Randomly choose (a) subset of variables and (b) subset of training data

- Good predictive accuracy
- Loss in interpretability

Boosting

- Combining weak learners, ϵ-better than random
  - E.g. Decision stumps

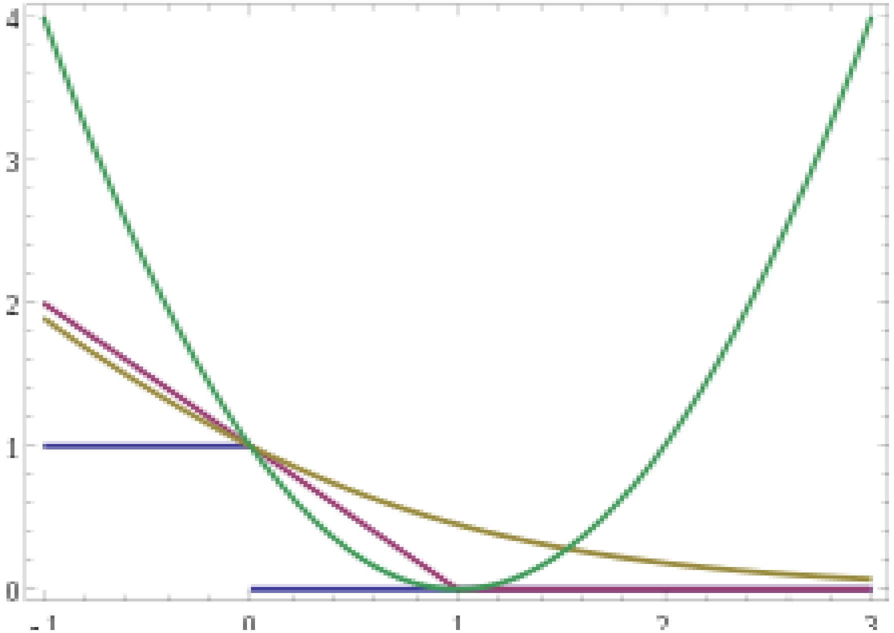
- Sequence of weighted datasets
- Weight of data point in each iteration proportional to no of misclassifications in earlier iterations

- Specific weighting scheme depends on loss function
- Theoretical bounds on error

Example loss functions and algorithms

- Squared error $\left(y_i - f(x_i)\right)^2$
- Absolute error $|y_i - f(x_i)|$

- Squared loss $(1 - \tilde{y}_i f(x_i))^2$
- 0-1 loss $I(\tilde{y}_i \neq f(x_i))$
- Exponential loss $\exp(-\tilde{y}_i f(x_i))$
- Logloss $\frac{1}{\log 2} \log(1 + e^{-\tilde{y}_i f(x_i)})$
- Hinge loss $|1 - \tilde{y}_i f(x_i)|_+$



Example: AdaBoost

- Binary classification problem + Exponential loss

1. Initialize $w_n^{(1)} = \frac{1}{N}$
2. Train classifier $y_m(x)$ minimizing $\sum_n w_n^{(m)} I(y_m(x_n) \neq y_n)$
3. Evaluate $\epsilon_m = \frac{\sum_n w_n^{(m)} I(y_m(x_n) \neq y_n)}{\sum_n w_n^{(m)}}$ and $\alpha_m = \log \frac{1-\epsilon_m}{\epsilon_m}$
4. Update wts $w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(x_n) \neq y_n)\}$
5. Predict $f_M(x) = sgn(\sum_{i=1}^{M} \alpha_m y_m(x))$

Neural networks: Multilayer Perceptrons

- Multiple layers of logistic regression models
- Parameters of each optimized by training

- Motivated by models of the brain
- Powerful learning model regardless

LR and R remembered …

- Linear models wi ons

$$y(x,w) = f(\sum_{i} w_i \phi_i(x))$$

- Fixed basis functions
- Non-linear transformation

- ϕ_i linear n

$$\hat{y}(x;w,v) = h(\sum_{j=1\ to\ M} w_{kj}\ g(\sum_{i=1\ to\ D} v_{ji}x_i))$$

Feed-forward network functions

- M linear combinations of input variables

$$a_j = \sum_{i=1\ to\ D} v_{ji} x_i$$

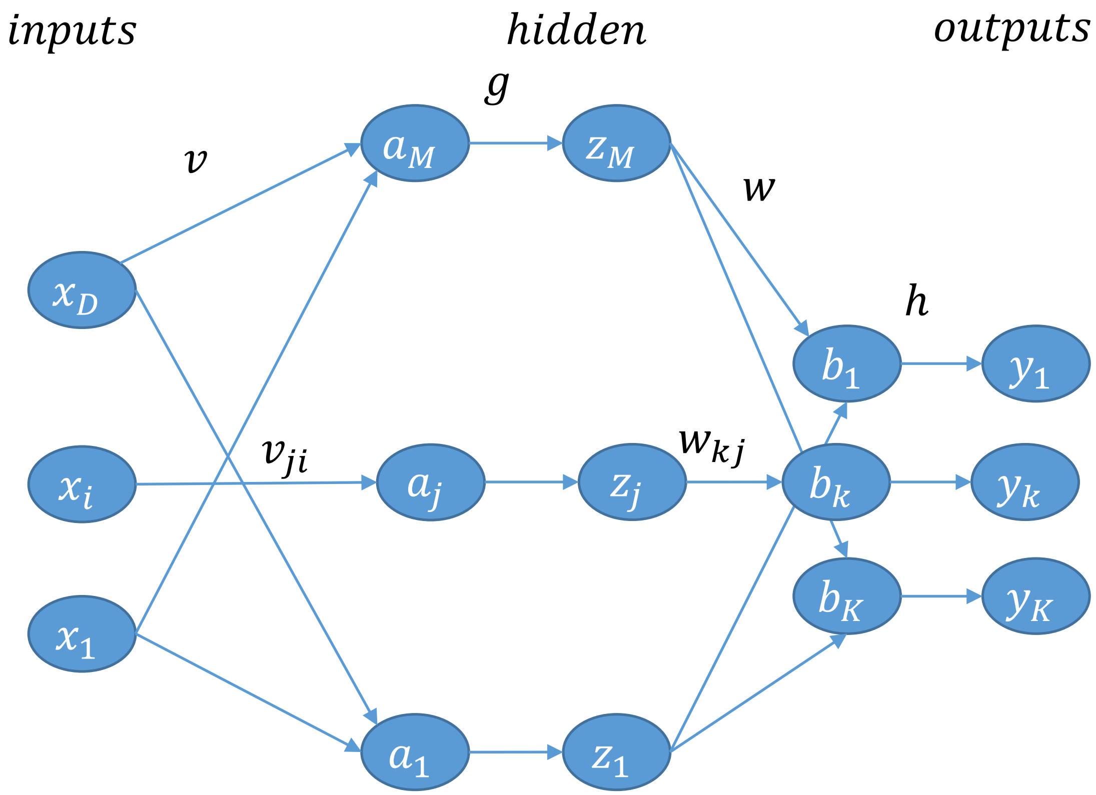
- Apply non-linear activati $z_j = g(a_j)$ on

- Linear combinations tc tivations

$$b_k = \sum_{j=1\ to\ M} w_{kj} z_j$$

- Apply output activation fu $y_k = h(b_k)$ get outputs

Network Representation



Easy to generalize to multiple layers

Power of feed-forward networks

- Universal approximators

A 2 layer network with linear outputs can uniformly approximate any smooth continuous function with arbitrary accuracy given sufficient number of nodes in hidden layer

- Why are >2 layers needed?

Training

- Formulate error function in terms of weights

$$E(w,v) = \sum_{i=1\ to\ N} \left|\left|\hat{y}(x_n;w,v) - y_n\right|\right|^2$$

- Optimize weights using gradient descent

$$(w,v)^{(t+1)} = (w,v)^{(t)} - \eta \nabla E((w,v)^{(t)})$$

- Deriving the gradient looks complicated because of feed-forward …

Error Backpropagation

- Full gradient: sequence of local computations and propagations over the network

Output layer

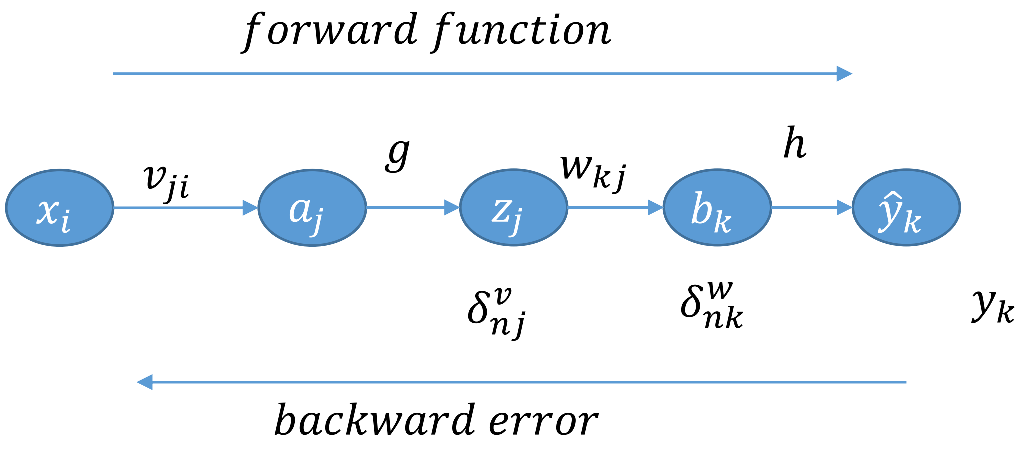
$$\frac{\partial E_n}{\partial w_{kj}} = \frac{\partial E_n}{\partial b_{nk}} \frac{\partial b_{nk}}{\partial w_{kj}} = \delta^w_{nk} z_{nj}$$

$$\delta^w_{nk} = \hat{y}_{nk} - y_{nk}$$

Hidden layer

$$\frac{\partial E_n}{\partial v_{ji}} = \frac{\partial E_n}{\partial a_{nj}} \frac{\partial a_{nj}}{\partial v_{ji}} = \delta^v_{nj} x_{ni}$$

$$\delta^v_{nj} = \sum_k \frac{\partial E_n}{\partial b_{nk}} \frac{\partial b_{nk}}{\partial a_{nj}} = \sum_k \delta^w_{nk} w_{kj} g'(a_{nj})$$



$$\frac{\partial E}{\partial w} = \sum_n \frac{\partial E_n}{\partial w}$$

Backpropagation Algorithm

1. Apply input vector x_n and compute derived variables $a_j, z_j, b_k, \hat{y}_k$
2. Compute δ_{nk}^{w} at all output nodes
3. Back propagate δ_{nk}^{w} to compute δ_{nj}^{v} at all hidden nodes
4. Compute derivatives $\frac{\partial E_n}{\delta w_{kj}}$ and $\frac{\partial E_n}{\delta v_{ji}}$
5. Batch: Sum derivatives over all input vectors

- Vanishing gradient problem

Neural Network Regularization

- Given such a large number of parameters, preventing overfitting is vitally important

- Choosing the number of layers + no of hidden nodes
- Controlling the weights
  - Weight decay
- Early stopping
- Weight sharing
- Structural regularization
  - Convolutional neural networks for invariances in image data

So… Which classifier is the best in practice?

- Low dimensions (9-200)

1. Boosted decision trees
2. Random forests
3. Bagged decision trees
4. SVM
5. Neural nets
6. K nearest neighbors
7. Boosted stumps
8. Decision tree
9. Logistic regression
10. Naïve Bayes

- High dimensions (500-100K)

1. HMC MLP
2. Boosted MLP
3. Bagged MLP
4. Boosted trees
5. Random forests

References

- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- J. Devore and R. Peck. *Statistics: The Exploration and Analysis of Data*. Duxbury Press, 1997.
- H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. *VLDB'01*
- J. E. Olson. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann, 2003
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB’2001
- T. Redman. *Data Quality: The Field Guide*. Digital Press (Elsevier), 2001

Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,
Dr. Cahit Karakuş
cahitkarakus@gmail.com

Thank You!