Data Preparation

Chapter · January 2017

DOI: 10.1007/978-1-4899-7687-1_62

3 authors, including:

Zahraa Said Abdallah
Monash University (Australia)
16 PUBLICATIONS **146** CITATIONS
SEE PROFILE

Geoffrey I Webb
Monash University (Australia)
380 PUBLICATIONS **7,084** CITATIONS
SEE PROFILE

To be published in C Sammut and G I Webb (Eds) Encyclopedia of Machine Learning and Data Mining, Springer, 2017.

Data Preparation

Zahraa S. Abdallah, Monash University
Lan Du, Monash University
Geoffrey I. Webb, Monash University

Synonyms

Data preprocessing.

Data Wrangling

1. Summary

Before data can be analyzed they must be organized into an appropriate form. Data preparation is the process of manipulating and organizing data prior to analysis.

Data preparation is typically an iterative process of manipulating raw data, which is often unstructured and messy, into a more structured and useful form that is ready for further analysis. The whole preparation process consists of a series of major activities (or tasks) including data profiling, cleansing, integration and transformation.

2. Motivation and Background

Data are collected for many purposes, not necessarily with machine learning or data mining in mind. Consequently, there is often a need to identify and extract relevant data for the given analytic purpose. Every learning system has specific requirements about how data must be presented for analysis and hence data must be transformed to fulfill those requirements. Further, the selection of the specific data to be analyzed can greatly affect the models that are learned. For these reasons, data preparation is a critical part of any machine learning exercise, and is often the most time-consuming part of any non-trivial machine learning or data mining project.

In most cases, the preparation process consists of dozens of transformations and needs to be repeated several times. Despite advances in technologies for working with data, each of those transformations may involve much-handcrafted work and can consume a significant amount of time and effort. Thus, working with huge and diverse data remains a challenge. It is often agreed that data wrangling/preparation is the most tedious and time-consuming aspect of data analysis. It has become a big bottleneck or "iceberg" for performing advanced data analysis, particularly on big data. A recent article in the New York Times [3] reported that the whole process of data wrangling could account up to 80% of the time in the analysis cycle. In other words, there is only a small fraction of time for data analysts and scientists to do analysis work. According to the data science report [8], published by Crown in 2015, messy and disorganized data are the number one obstacle holding data scientists back. The same study reports that 70% of a data scientist’s time is spent in cleaning data.

3. Processes and Techniques

The manner in which data are prepared varies greatly depending upon the analytic objectives for which they are required and the specific learning techniques and software by which they are to be analyzed. The following are a number of key processes and techniques.

3.1. Data Profiling: Sourcing, selecting and auditing appropriate data

It is necessary to review the data that are already available, assess their suitability to the task at hand and investigate the feasibility of sourcing new data collected specifically for the desired task. It is also important to assess whether there are sufficient data to realistically obtain the desired machine learning outcomes.

Data quality should also be investigated, as data sets are often of low quality. Those responsible for manual data collection may have little commitment to assuring data accuracy and may take shortcuts in data entry. For example, when default values are provided by a system, these tend to be substantially over-represented in the collected data. Automated data collection processes might be faulty, resulting in inaccurate or incorrect data. The precision of a measuring instrument may be lower than desirable. Data may be out-of-date and no longer correct.

Assuring and improving data quality are two of the primary reasons for data preprocessing. There are common criteria to measure and evaluate the quality of data, which can be categorized into two main elements; accuracy and uniqueness [9] as explained in Figure 1.

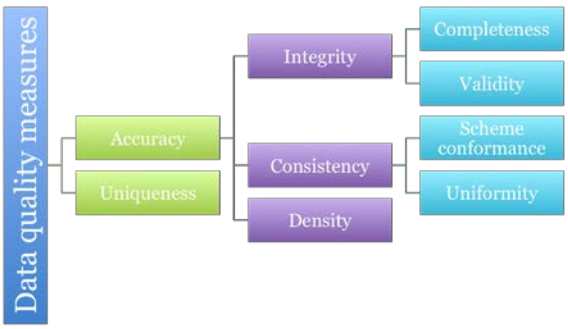

Figure 1: Data Quality Measures (adapted from [9])

Accuracy is described as an aggregated value over the quality criteria: Integrity, Consistency, and Density. Intuitively this describes the extent to which the data are an

exact, uniform and complete representation of the *mini-world*: the aspects of the world that the data describe. We describe each accuracy criterion as follows:

- **Integrity:** An integral data collection contains representations of all the entities in the mini-world and only of those. Integrity requires both completeness and validity.
    - **Completeness:** Complete data give a comprehensive representation of the mini-world and contain no missing values. We achieve completeness within data cleansing by correcting anomalies and not just deleting them. It is also possible that additional data are generated, representing existing entities that are currently unrepresented in the data. A problem with assessing completeness is that you don't know what you don't know. As a result, there are no known gold standard data, which can be used as a reference to measure completeness.
    - **Validity:** Data are valid when there are no constraints violated. There are numerous mechanisms to increase validity including mandatory fields, enforcing unique values, and data schema/structure.
- **Consistency**: This quality concerns syntactic anomalies as well as contradictions. The main challenge concerning data consistency is choosing which data source you trust for reliable agreement among data across different sources.
    - **Schema conformance:** This is especially true for the relational database systems where the adherence of domain formats relies on the user.
    - **Uniformity:** is directly related to irregularities.
- **Density:** This criterion concerns the quotient of missing values in the data. There still can be non-existent values or properties that have to be represented by null values having the exact meaning of not being known.

The above three criteria of Integrity, Consistency, and Density collectively represent the accuracy measure.

The other major quality measure that is also crucial to measure data quality is uniqueness. Uniqueness is satisfied when the data do not contain any duplicates.

Timeliness is another criterion that also has been considered for data quality. This criterion refers to the currency of the data that keeps it up to date.

More information about data quality can be found in [4] and [9]

3.2. Data Cleansing:

Where the data contain noise or anomalies it may be desirable to identify and remove outliers and other suspect data points, or take other remedial action. See noise

Data cleansing is defined as the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Data cleansing can also be referred to as data cleaning, data scrubbing, or data reconciliation. More precisely, the process of data cleansing could be explained as a four-stage process:

1. Define and identify errors in data such as incompleteness, incorrectness, inaccuracy or irrelevancy.

2. Clean and rectify these errors by replacing, modifying, or deleting them
3. Document error instances and error types; and finally
4. Measure and verify to see whether the cleansing meets the user's specified tolerance limits in terms of cleanliness.

3.2.1. Data anomalies:

Data are symbolic representations of information, i.e., facts or entities from parts of the world, called a mini-world, depicted by symbolic values. Imperfections in the dataset correspond to differences between an ideal (i.e., error-free) dataset (DI) and the real data (DR). In this context, anomalousness is a property of data that renders an erroneous representation of the mini-world.

The term *data anomaly* describes any distortion of data resulting from the data collection process. From this perspective, anomalies include duplication, inconsistency, missing values, outliers, noisy data or any kind of distortion that can cause data imperfections.

Anomalies can be classified at a high level into three categories:

- **Syntactic Anomalies:** describe characteristics concerning the format and values used for the representation of the entities. Syntactic anomalies include lexical errors, domain format errors, syntactical errors, and irregularities.
- **Semantic Anomalies:** hinder the data collection from being a comprehensive and non-redundant representation of the mini-world. These types of anomalies include integrity constraint violations, contradictions, duplicates and invalid tuples.
- **Coverage Anomalies:** decrease the number of entities and entity properties from the mini-world that is represented in the data collection. Coverage anomalies are categorized as missing values and missing tuples

Therefore, it is clear that data anomalies can take a number of different forms, each with a different range of analytical consequences.

3.2.2. Data cleansing process:

Data cleansing is an iterative process that consists of the four consecutive steps [9], as depicted in Figure 2:

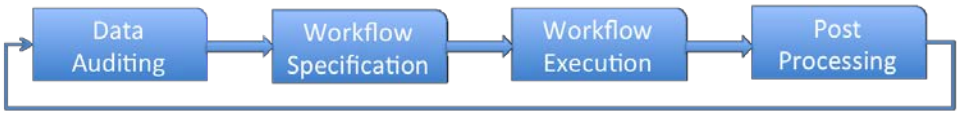

Figure 2: Data cleansing process (adapted from [9])

1. Data Auditing: This first step mainly identifies the types of anomalies that reduce data quality. Data auditing checks the data using validation rules that are pre-specified, and then creates a report of the quality of the data and its problems. We often apply some statistical tests in this step for examining the data.
2. **Workflow specification:** The next step is to detect and eliminate anomalies by a sequence of operations on the data. The information collected from data auditing is then used to create a data-cleaning plan. It identifies the causes of the dirty data and plans steps to resolve them.
3. Workflow execution: The data cleaning plan is executed, applying a variety of methods on the data set.
4. Post-processing and controlling: The post-processing or control step involves examination of the workflow results and performs exception handling for the data mishandled by the workflow.

3.2.3. Dealing with Missing values:

One major task in data cleansing is dealing with missing values. It is important to determine whether the data have missing values and, if so, to ensure that appropriate measures are taken to allow the learning system to handle this situation. See missing attribute values.

Handling data that contain missing values is crucial for the data cleansing process and data wrangling in general. In real-life data, most of existing data sets contain missing values that were not introduced or were lost in the recording process for many reasons.

3.2.4. Handling outliers:

An outlier is another type of data anomaly that requires attention in the cleansing process. Outliers are data that do not conform to the overall data distribution.

Outliers can be seen from two different perspectives; first, they might be seen as glitches in the data. Alternatively, they might be also seen as interesting elements that could potentially represent significant elements in the data. For example, outliers in sales records for a store might reflect a successful marketing campaign. Therefore, to classify data as outliers, we must define what the normal behaviour of the data is and therefore how different or significant the outlier is relative to normal behaviour. There might be different normal behaviours for data and thus different classes of outliers. From the above definition, we can see that as the normality in data differs, various classes of outliers can be detected. To be able to do that, we need to formalize both the normality in the data and inconsistency of the outliers. Read more about handling outliers for data preprocessing in [10].

3.3. Data Enrichment/Integration:

Existing data may be augmented through data enrichment. This commonly involves sourcing of additional information about the data points on which data are already held. For example, customer data might be enriched by obtaining socio-economic data about individual customers. The imported data must be integrated with the other data for a unified view of all data sources.

Data integration is a crucial task in data preparation. Combining data from different sources is not trivial especially when dealing with large amounts of data and heterogeneous sources. Data are typically presented in different forms (structured, semi-structured or unstructured) as well as from different sources (web, database) that could be stored locally or distributed. Moreover, structured data coming from a single source, might have different schemas. The combination of these variations is not an easy task.

Integration of data brings many opportunities, yet it also comes with various challenges. We highlight the most relevant challenges below:

1. **Data are heterogeneous:** Data integration involves a combination of data coming from different sources that have been developed independently of each other and thus vary in data format. Each source will have its own schemas, definition of objects and structure of data (tables, XML, unstructured text, etc.).
2. **The number of sources:** Data integration is already a challenge for a small number of sources, but the challenges are exacerbated when the number of sources grows (such as Web-scale data integration).
3. **Object identity and separate schemas:** Differences exist both on the level of individual objects and the schema level. Every source classifies their data according to taxonomies pertinent to a certain domain.
4. **Time synchronization:** Each source might have a different time window over which data have been captured, different granularities at which events are modelled (daily, weekly, annually), and frequency at which they are updated. Synchronisation of these differences and making time-sensitive data compatible is another challenge.
5. **Dealing with legacy data:** There are still important data stored in a legacy form such as IMS, spreadsheets and ad-hoc structures. Combining legacy data with other modern data structures such as XML is a challenging task.
6. **Abstraction levels:** Different data sources might provide data at incompatible levels of abstraction. When combining data differences in levels of specificity must be resolved.
7. **Data quality:** Data are often erroneous, and combining data often aggravates the problem. Erroneous data has a potentially devastating impact on the overall quality of the integration process.

The integration process can be divided into two main sub-tasks: schema integration and data integration, where each has its own techniques and challenges. Schema integration concerns a holistic view across data sources. It focuses on formats, structures and identification of objects and their level of abstraction. This includes semantic mapping, matching, resolving naming conflicts, entity resolution. The contents of data add another clue to the integration process.

Even with data from different sources that have identical schemas, integration on the data level is still essential. Data integration deals with different types of problems that concern the data itself rather than the overall structure as in schema integration. Common data integration problems are duplication in data and inconsistency. Correlated or duplicated values/attributes may increase both size and complexity of the data. Resolving conflicts at the data level enhances the overall performance of the integration process.

3.4. Data Transformation:

It is frequently necessary to transform data from one representation to another. There are many reasons for changing representations:

- **To generate symmetric distributions instead of the original skewed distributions**:
- **Transformation improves visualisation** of data that might be tightly clustered relative to a few outliers
- Data are transformed to achieve **better interpretability.**
- Transformations are often used to **improve the compatibility of the data with assumptions underlying a modelling process**, for example, to linearize (straighten) the relation between two variables whose relationship is non-linear. Some of the data mining algorithms require the relationship between data to be linear.

In the following we will discuss different types of transformation whereby each data point $\mathbf{x_i}$ is replaced with a **transformed value** $\mathbf{y_i = f(x_i)}$, where **f** is the transformation function. Many techniques are applied for data transformation. Each technique has its own purpose and dependency on the nature of data. Some of the major transformations are discussed below.

3.4.1. Numeric to Numeric Transformation

3.4.1.1. Normalization and rescaling

It is usually the case that raw data are not in a suitable form to be processed by machine learning and data mining techniques. Data normalization is the process of transforming raw data values to another form with properties that are more suitable for modeling and analysis. The normalization process focuses on scaling data in terms of range and distribution. Therefore, it consists of two main processes:

- ***Min-max normalization*** projects the original range of data onto a new range. Very common normalization intervals are [0,1] and [-1,1]. This normalization method is very useful when we apply a machine learning or data mining approach that utilizes distance. For example, in *k*-nearest neighbor methods, using un-normalized values might cause attributes whose values have greater magnitudes to dominate over other attributes. Therefore, normalization aims to standardize magnitudes across variables. A useful application for min-max scaling is image processing where pixel intensities have to be normalized to fit within a certain range (i.e., 0 to 255 for the RGB color range). Also, typical neural network algorithms (ANN) require data that is on a 0-1 scale. Normalization provides the same range of values for each of the inputs to the model.

- **Z-score Normalization (**also referred to as standardization) is a normalization method that transforms not only the data magnitude, but also the dispersion. Some data mining methods are based on the assumption that data follow a certain distribution. For example, methods such as logistic regression, SVM and neural network when using gradient descent/ascent optimization methods assume data follow a Gaussian distribution. Otherwise, the approaches will be ill-conditioned and might not guarantee a stable convergence of weight and biases. Other approaches such as linear discriminant analysis (LDA), principal component analysis (PCA) and kernel principal component analysis require features to be on the same scale to find directions that maximizing the variance (under the constraints that those directions/ eigenvectors/ principal components are orthogonal). Z-score normalization overcomes the problem of variables with different units as it transforms variables so that they are centered on 0 with a standard deviation of 1.

- **Decimal Scaling** is another type of scaling transformation where the decimal place of a numeric value is shifted so the maximum absolute value will be always less than 1.

3.4.1.2. Linear transformation:

Linear transformations preserve linear relationships within data. A function f(.) results in a linear transformation if and only if for all values x and y in the original representation, f(x)+f(y) = f(x+y) and f(x)-f(y) = f(x-y). Examples of a linear transformation are transforming Celsius to Fahrenheit, Miles to Kilometers and Inches to Centimeters. All linear transformations follow the standard linear regression formula to convert variables linearly.

Many other transformations are not linear. A nonlinear transformation changes (increases or decreases) linear relationships between variables and, thus, changes the correlation between variables. Examples of nonlinear transformations are: square root, raising to a power, logarithm, and any of the trigonometric functions. In the following, we discuss some nonlinear transformation methods.

3.4.1.3. Power transformation (Tukey's Ladder of Powers):

Tukey (1977) describes a way of re-expressing variables using a power transformation [11]. The aim of this transformation is to improve the linearity between variables. When we consider two variables (x and y), transformation can be applied to one variable or both of them depending on the relationship between the two variables. This kind of transformation fits when the relationship between the two variables is monotonic and has a single bend. When the data are represented as pairs of (x,y), Tukey has expressed data transformation as:

$$y^a = \beta_0 + \beta_1 + x^b$$

The choice of ***a*** and ***b*** decides on the transformation type in the relationship between x and y. Figure 3 shows a visual rule of thumb that has been proposed by John Tukey. The

following diagram gives us an insight to understand which transformations are likely to work with different types of data.

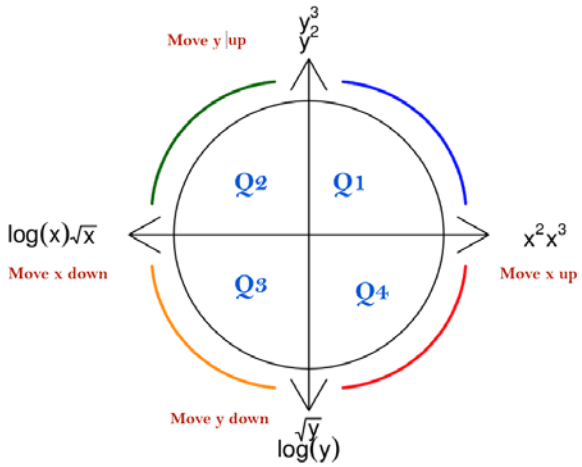

Figure 3: Tukey's ladder rule

We explain Tukey's ladder rule as follows: Suppose the data patterns follow a similar curve as the blue line in Q1, thus the data could be transformed by going up the ladder for x, y or both. If the data pattern is shaped similar to that shown in Q2, then we should try to transform the data by going the down-ladder for x, and/or up-ladder for y. Similar procedures can be applied for the other two quarters. Figure 4 explains the ladder of power for variable y. The transformation is stronger when the power value is away from 1 (the original data) in both directions (up and down).

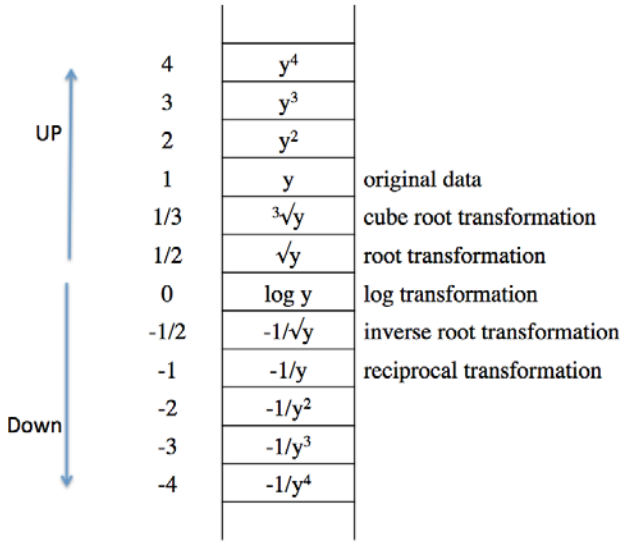

Figure 4: The ladder of power

3.4.1.4. Choosing the right numeric transformation:

There is no definite answer to what is the best transformation method to use for a particular data set. The choice is very data dependent and requires an understanding of the domain as well as the data distribution. Trial and error for the common transformation methods may also be required. Table 1 summarizes the main transformation methods.

Method	Pros	Cons
Standard linear regression	-Preserves the relationship between variables	-No actual transformation occurred.
Reciprocal transformation	- Making small values bigger and big values smaller - Reducing the effect of outliers	-Not applicable for zero
Log transformation	-Good for right skewed data -$\log_{10}(x)$ is especially good at handling higher order powers of 10 (e.g. 1000, 100000)	-Not applicable for zero and negative values (constant can be added to overcome this)
Root transformation	-Simple counts -Good for right skewed data	-Not applicable for negative values(constant can be added to overcome this)
Logit transformation	-Works with proportions and percents.	-Not applicable for 0 and 1 values
Cube root transformation	-Can be applied on negative and 0 values	-Not effective in transformation as log model

Table 1: Summary of key transformation methods

3.4.2. Nominal to numeric transformation:

All the aforementioned methods, transform and re-express numerical variables. However, the transformation of nominal variables is equally important, especially for machine learning and data mining methods that only accept numerical values such as SVM and ANN. Assume that we have a nominal variable *x* with *N* different nominal values. There are two approaches to transform *x* into a numeric variable:

1. The first and simplest approach is to map nominal values to the integers 1 to *N*. Although simple, this method has two major drawbacks:
   - Integer substitution may impose an ordering that does not actually exist in the original data.
   - The integer value might be used as part of the calculation in the mining algorithm giving an incorrect meaning of weights based on the assigned values.
2. The other main approach is to first binarize the variable (see 3.7 Binarization) and then map each of the *N* new binary attributes to the integer values 0 and 1. This approach is generally viewed as safer than the first and hence is more widely used.

3.5. **Propositionalisation.**

Some data sets contain information expressed in a relational format, describing relationships between objects in the world. While some learning systems can accept relations directly, most operate only on attribute-value representations. Therefore, a relational representation must be re-expressed in attribute-value form. In other words, a representation equivalent to first-order logic must be converted to a representation equivalent only to propositional logic.

3.6. Discretization:

Discretization transforms continuous data into a discrete form. This is useful in many cases for: better data representation, data volume reduction, better data visualization and representing data at a various level of granularity for data analysis. Data discretization approaches are categorized as supervised, unsupervised, bottom-up or top down. Approaches for data discretization include Binning, Entropy based, Nominal to numeric, 3-4-5 rule and Concept hierarchy. See Data Discretization

3.7. Binarization:

Some systems cannot process multi-valued categorical variables. This limitation can be circumvented by binarization, a process that converts a multi-valued categorical variable into multiple binary variables, one new variable to represent the presence or absence of each value of the original variable.

Conversely, multiple mutually exclusive binary variables might be converted into a single multi-valued categorical variable.

3.8. Granularity:

It is important to select appropriate levels of granularity for analysis. For example, when distinguishing products, should a gallon of low fat milk be described as a diary product, and hence not distinguished from any other type of dairy product, be described as low fat milk, and hence not distinguished from other brands and quantities, or uniquely distinguished from all other products?

Analysis at the lowest level of granularity makes possible identification of potentially valuable fine-detail regularities in the data, but may make it more difficult to identify high-level relationships.

4. Dimensionality reduction

As many learning systems have difficulty with high dimension data, it may be desirable to project the data onto a lower dimensional space. Popular approaches to doing so include Principal Components Analysis and Kernel Methods.

5. Feature engineering:

It is often desirable to create derived values. For example, the available data might contain fields for purchase price, costs and sale price. The relevant quantity for analysis might be profit, which must be computed from the raw data.

Feature engineering can be considered as means for dimensionality reduction also, by replacing the original features by a smaller number of derived features.

See Feature Selection and Feature Construction.

6. Sampling

Much of the theory on which learning systems are based assumes that the training data are randomly sampled from the population about which the user wishes to learn a model. However, much historical data contains sampling biases, for example, data that were easy to collect or were considered interesting for some other purpose. It is important to consider whether the available data are sufficiently representative of the future data to which a learned model is to be applied.

In all sampling methods, the aim is to select a sample *S* containing *N* instances from the entire data *D*. Each method models the relationship between a population and a sample with an underlying mathematical process. We discuss in the following some of these methods:

6.1. Random sampling:

In this method, *S* is selected randomly from *D* with a probability of 1/*N* for any instance in *D* to be selected. *Simple random sampling* may have very poor performance in the presence of skew in data. There are two main variants of simple random sampling, *with replacement* (SRSWR) and *without replacement* (SRSWOR). For sampling with replacement the instance that is drawn from the population is replaced and therefore it might be chosen again. For sampling without replacement, each instance that is drawn from *D* is removed and hence *S* must contain *N* distinct instances.

Simple random sampling is usually easy to implement and to understand. However, it might cause loss of accuracy if applied to skewed data by failing to include sufficient data to accurately represent the tail of the distribution. The simple random sample might also result in substantial variance across samples.

6.2. Cluster sampling:

This method approximates the percentage of each class (or subpopulation of interest) in the overall dataset then it draws a simple random sample from each cluster. In this method, we might not have a complete list of population members (i.e., not all data available). However, a list of groups or 'clusters' of this population is available and complete. That means the clusters could be incomplete but a list of them is complete. Therefore, cluster sampling is a cost efficient sampling method, as it doesn't require data to be complete. A drawback for cluster sampling is the possible poor representation of the diversity in clusters.

6.3. Stratified sampling:

If *D* is divided into mutually disjoint parts called strata, obtaining a simple random

sample from each stratum generates a stratified sample.

Stratified sampling has a number of advantages. First, inferences can be made about specific subgroups for more efficient statistical estimates. Since each stratum is treated as an independent population, different sampling approaches can be applied to different strata. Second, this method will never result in lower efficiency than the simple random sample, provided that each stratum is proportional to the group's size in the population. Finally, it increases data readability as it represents individual pre-existing strata within a population rather than the overall population.

Stratified sampling is complex to implement and estimate. It also can be sensitive to parameters such as selection criteria and minimum group size. Finally, stratified sampling techniques are generally used when the population is heterogeneous, or dissimilar, where certain homogeneous, or similar, subpopulations can be isolated (strata). Thus, this method will not useful when there are no homogeneous subgroups. Read more about sampling techniques in [7].

Balanced sampling is a special case of stratified sampling where the strata correspond to the classes and the sample drawn from each strata is proportional to the class's size in the population.

See Also:

Data Set
Discretization,
Evolutionary Feature Selection and Construction
Feature Construction
Feature Selection in Text Mining
Feature Selection: An Overview
Kernel Methods
Measurement Scales
Missing Values
Noise
Principal Component Analysis
Propositionalisation
Binning
Dimensionality Reduction
Record Linkage

References and Recommended Reading

[1] D. Pyle, Data Preparation for Data Mining. Morgan Kaufmann, 1999.
[2] Ian H. Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques (Second Edition) Morgan Kaufmann, 2005.
[3] "For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

(http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=0&module=ArrowsNav&contentCollection=Technology&action=keypress&region=FixedLeft&pgtype=article)" (the NYT article by Steve Lohr)
[4] Dasu, Tamraparni, and Theodore Johnson. Exploratory data mining and data cleaning. Vol. 479. John Wiley & Sons, 2003.
[5] Barnett, Vic., and Lewis, Toby. Outliers in Statistical Data, 3rd ed. Chichester; New York: Wiley, 1994. Print. Wiley Ser. in Probability and Mathematical Statistics. Applied Probability and Statistics.
[6] Doan, AnHai, Alon Halevy, and Zachary Ives. Principles of data integration. Elsevier, 2012
[7] García, Salvador, Julián Luengo, and Francisco Herrera. Data preprocessing in data mining. Switzerland: Springer, 2015.
[8] Data science report (http://visit.crowdflower.com/2015-data-scientist-report.html)
[9] Müller, Heiko, and Johann-Christph Freytag. Problems, methods, and challenges in comprehensive data cleansing. Professoren des Inst. Für Informatik, 2005.
[10] Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.
[11] Tukey, John W. Exploratory data analysis. (1977): 2-3.