ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Special Issue (April)

Online - 2455-3891
Print - 0974-2441

Full Proceeding Paper

DATA PREPARATION ON LARGE DATASETS FOR DATA SCIENCE

DARSHAN BARAPATRE, VIJAYALAKSHMI A*

School of Computing Sciences and Engineering, VIT University, Chennai, India. Email: vijayalakshmi.av@vit.ac.in

Received: 23 January 2017, Revised and Accepted: 03 March 2017

ABSTRACT

According to interviews and experts, data scientists spend 50-80% of the valuable time in the mundane task of collecting and preparing structured or unstructured data, before it can be explored for useful analysis. It is very valuable for a data scientist to restructure and refine the data into more meaningful datasets, which can be used further for analytics. Hence, the idea is to build a tool which will contain all the required data preparation techniques to make data well-structured by providing greater flexibility and easy to use UI. Tool will contain different data preparation techniques which will include the process of data cleaning, data structuring, transforming data, data compression, and data profiling and implementation of related machine learning algorithms.

Keywords: Data preparation, Data mining, Machine learning, Map reduce, SPARK, Apache pig, Apache oozie.

 DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.20526

INTRODUCTION

Data preparation is known as the most important and time-consuming phase of data analytics. The real world data are unstructured and extremely complicated to interpret without data preprocessing [1]. After collecting the data from multiple sources, we need to prepare it for different types of analysis, model building and to infer knowledge from the same [2]. This might include integrating multiple data sets together, matching record attributes, eliminating duplication, checking data type validation, correcting errors, and transforming fields. Again, this can be a time-consuming process, and it usually needs to be done each time you perform an analysis [3] - so if we are generating a daily, weekly, or monthly report, we may need to go through the entire data cleansing process for each report generation. As an inference, it is estimated that up to 80% of analysts' time is spent on gathering and preparing data. Spending too much time on preparation diverts the analysts from more valuable knowledge discovery task, and in some cases, this might even discourage them from attempting it [4].

Focusing on impact of data preparation before applying machine learning algorithms to build a model, has following advantages:

- We can easily fit and apply data mining algorithms.
- Effectively improves the performance of the mining algorithms
- Prepared data can be in understandable format to both machines and humans
- Data retrieval from databases can be faster
- Data available will be well suitable for a specific analysis and model building.

On the business side, these include far more insights, sooner, which lead to positive, and significant business outcomes.

Approach is to build a product that removes all the complexity from analyzing data using differe nt data preparation techniques using machine learning algorithms. It provides fast processing paradigm, such as MapReduce and Apache Spark for processing large amount of data [5].

PROBLEM STATEMENT

To make an easy to use and flexible this tool will contain all the possible data preparation techniques in order to make more meaningful and well-structured data to get the best results. To understand different problems, new methods, and effectiveness of each technique this processes helps for data preparation. Many data preparation tools are available out there, but they are unable to process big data, which are the main feature of this tool. Furthermore, some of these tools required complex coding which will be not required for using this tool. As an inference, we will be making a tool which can easily handle large amount of data using MapReduce, Spark, Pig, etc., and will provide easy to use, drag-and-drop user interface containing more machine learning algorithm for accurate faster results to the queries.

LITERATURE SURVEY

In this paper, our main focus will be to improve the quality of data and data preprocessing results [1]. To do that, raw data have to be preprocessed to improve efficiency and ease of model building process. To deal with data preparation, we categorized different techniques as:

- Data cleaning
- Data transformation
- Data profiling
- Data reduction.

Data cleaning is the process of finding and correcting (or removing) data which is incomplete, missing, not in proper format, or repeated [2]. To reduce the execution time, complexity of data mining process is very important, and to increase the quality of data, data cleaning is very important. Different techniques in data cleaning include handling missing values, removing NA, data redundancy, data type or format checking, data binning, regression, and clustering.

Data transformation is the technique of converting a set of data values collected from a data system into the data format of required by the other data system [3]. It converts data from one format (e.g. a database file or Excel sheet) to another. Many times, data reside at different locations and in different formats among various data centers or data sources. In such cases, data transformation is necessary to ensure that dataset or format of data from one application or database should be intelligible to other applications and databases. In scenarios where we want to share the information, data have to be extracted from the data source such as data warehouse and then transform it into another format, and finale needs to load at the target location. Discretization and filling missing data are the mostly used forms of data transformation. Other forms of data transformation include data normalization, smoothing, and regular expression transformations [6].

Data profiling refers to examining data to learn about important characteristics and importance of relationship and correlation between its attributes [7]. In profiling data, we have to model the structure, content, distribution, and uniqueness. Various phases in data profiling include data preview and selection, attribute or subset selection, value distribution analysis, range analysis, and pattern analysis. Data profiling highlights the strength, weakness, and uniqueness of your data.

Data reduction, also known as data compression, is a technique in which we will get a reduced representation of existing data set which is smaller and sometimes dimension but still can produce the same (or almost the same) analytical results. In this technique, we will eliminate that features or attributes which have very low or no impact on model accuracy and other important parameters. Possible data reduction techniques include dimensionality reduction, feature selection, feature creation, and feature subset selection.

Text data can also be preprocessed using this tool. Text mining includes extracting meaningful insight from piece of text, which require some preprocessing before applying any classification or clustering model on text data. Text preprocessing includes stemming, stop word removal, spelling correction, feature extraction, word weightage, special symbols, numbers, punctuation, and removal.

We are looking forward to deploy machine learning algorithms [8] to cover up following factors:

- Clustering algorithms
  - Outlier detection
  - Smoothing
  - Data discretization.
- Decision tree ensembles
  - Concept hierarchy generation.
- Feature selection
- Feature weighting
- Data compression.
  - PCA
- Correlation analysis
  - Chi-square test
  - SVD.

METHODS

As enterprises and internet are creating more diverse and product-oriented data and information, and need for machine learning is very high, using which enterprises can perform and derive recommendations of products or services, generalized insights, trends, etc. Data analyst used to tackle such problems using tools such as R and Python which are easy to use, having support of large open source community and so very popular [9]. However, as the world is generating large volume and variety of data at high velocity, analyst is forced to spend their majority of time using the same infrastructure and environment, instead of deriving results from the machine learning models to solve their main problems [8].

To tackle such problem, there are techniques and framework which can handle vast amount of data. One such framework we can adopt is Hadoop shown in Fig. 1. It is a software framework which supports distributed storage and processing of huge data bundles across the clusters of machines. Hadoop is basically based on 3 things: A file system called Hadoop distributed file system (HDFS) which is a distributed file system and a computation or processing framework (MapReduce), and YARN [5]. MapReduce is the main processing module of Hadoop, it has following features: Scalability, cost-effective solution, flexibility, fast, and parallel processing. In addition, we will be using YARN, yet another resource negotiator is a job scheduling, and Hadoop resource manager uses to manage the resources of each job submitted to Hadoop.

Sometimes, writing java MapReduce code is very lengthy and hectic process to follow. Furthermore, it involves more development effort and time. Here, Apache Pig comes to rescue. We can write 100 of lines of Java code written for MapReduce to 10 lines in pig Latin. Pig is a dataflow language. It is consists of two parts: Pig interpreter and the language, pig Latin. We have to write Pig script in pig Latin and pig interpreter processes them.

When it comes to machine learning algorithms, we have Apache Spark's machine learning library (MLlib), which is use for performing machine learning and associated tasks on large data sets. It designed for scalability and its integration [10]. It provides all its library codes in Scala, Java, and Python with which data scientists can solve and can get insight through their data problems much faster. We will be using built-in to MLlib are algorithms in our tool for:

- Handling data types in forms of vectors and matrices
- Calculating basic statistics such as summary, correlations, conducting simple hypothesis testing
- Use of classification and regression modeling for categorizing and prediction
- Clustering such as K-means and DB scan for outlier detection
- Performing dimensionality reduction
- Feature extraction and transformation
- Correlation analysis and feature weighting
- Decision tree ensembles for concept hierarchy generation.

The Spark MLlib is still under development and expected to add new algorithms to the existing library.

Another important tool we will be using is Apache Oozie. Main purpose of using Apache Oozie is to manage different jobs that are being processed in Hadoop system. In Oozie, dependencies between the jobs are need to be specified by the user in the form of directed acyclic graphs. Oozie uses this information and takes care of the execution process in the order as specified in a workflow. This way user's time to manage and complete workflow is been saved. It has a provision to specify frequency of execution of a particular job. Its other features include: Its Web Service APIs with which we can control jobs from anywhere, it is a provision to run jobs which are scheduled to execute periodically and we can get email notifications on completion of jobs.

RELATED WORK

Building a flexible and resistance tool for big data requires above-mentioned framework stack. Another important goal is the use of machine learning algorithms to help the process become more accurate and faster. To deploy more and more algorithms, we will be categorizing the data preparation techniques and will try to find suitable algorithm for each. We already described the possible techniques in Fig. 2, now we will look at how to adapt machine learning algorithms related to each category.

One such example is that we may not just look at a box plot or scatter plot to look for outliers, as there could be multivariate outliers which are not visible if you plot a single variable. To find multivariate outliers, there are different approaches, such as Gaussian mixture models and

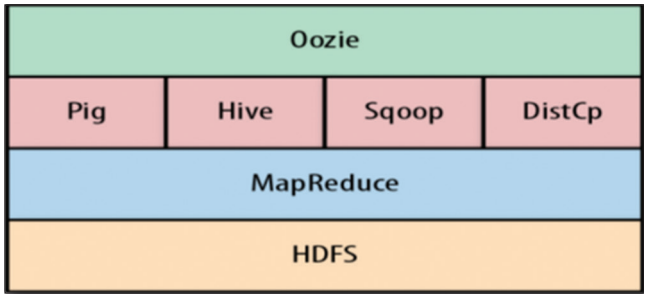

Fig. 1: Used tools and framework stack

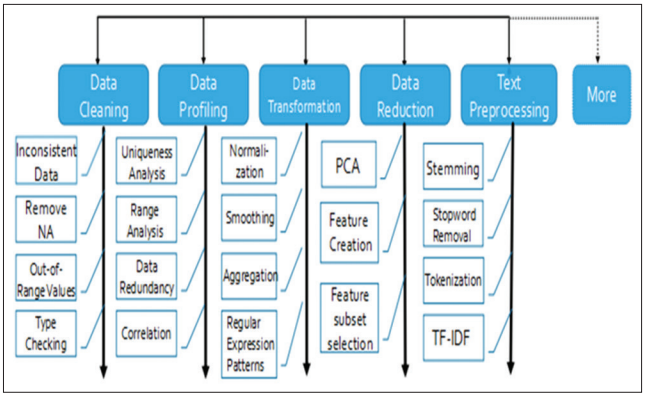

Fig. 2: Data preparation techniques

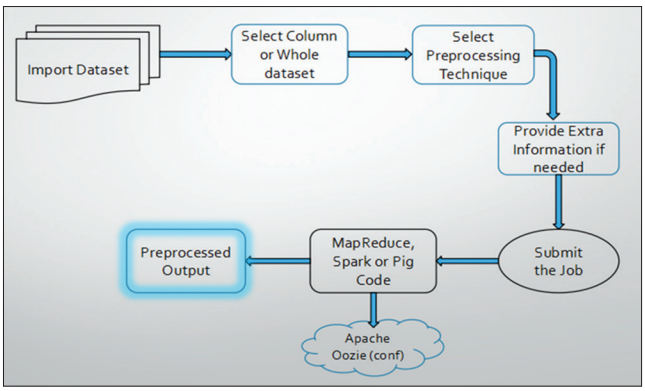

Fig. 3: Proposed system with Spark and MapReduce framework

expectation-maximization algorithms that use "Mahalanobis distance" to detect multivariate outliers.

The process of feature extraction can be done with the Spark structured query language [11], while we can also use Spark's MLlib which has some special functions for such task, such as Term frequency-inverse document frequency and Word2Vec.

For dimensionality reduction, principle component analysis (PCA) is a very mature and commonly used method which is often used to find a small set of variables that counts for most of the variance. Mathematically, PCA determines the low-dimensional subspace that captures as much of the variance of a dataset as possible.

Feature selection is used to remove redundant or irrelevant features but it can be used for the following reasons also:

- Creating fewer chances for overfitting
- Making models easier to understand
- Saving time and space for model estimation.

In MLlib, we can use the "ChiSqSelector" algorithm, which is the sum of squared errors or can be constructed through sample variance. We can use Chi-squared test for variable importance, variable independence, and for distribution of variable.

Moreover, we would like to add general transformation and validation such as data type conversion, search and replace transformation, case transformation, binning, and min-max normalization.

PROPOSED SYSTEM

First phase of proposed system in Fig. 3 includes dataset importation. User can import data from local file system or from HDFS. We are also planning to add data extraction module to collect data from social networking site such as Twitter and Facebook, and also, provision to connect to other machine or cluster to grab required data.

After getting data into current working environment, user interface will show the imported data and will be provided with the options to preprocess data. Before selecting data preparation technique, user will be ask to either process whole data, or if it is structured, user can apply listed preprocessing technique on any column or record if applicable.

Some of the techniques might require such as column name, some k value in case of k-means clustering, and KNN classification and it can also be an input file, JSON schema, and properties file, which will be processed through Oozie.

After selecting appropriate parameters, input files with appropriate class information, and output directory, job is get submitted to the Oozie workflow which will manage the different jobs and schedule it accordingly in the workflow. These jobs can be a MapReduce job or Spark job, depending on preprocessing technique. Apache Oozie will also be managing configuration files and other metadata.

For all the preprocessing techniques, which will be available in tool, most possibly it will be an algorithm written as MapReduce code or Pig job. Moreover, for most of the machine learning algorithms, we will be using Spark MLlib, code written in Scala or Java.

After the successful execution of jobs, output will be stored on HDFS path, a path provided by the user, which will contain transformed data, invalid records, data anomalies, generated features, or reduced dataset, depending on the selected technique.

The preprocessing technique module in the proposed system will contain following algorithms:

- DBSCAN clustering

  Let X = {x_1, x_2... x_n} be the set of data points. DBSCAN needs 2 parameters: ε (eps) and a minimum number of points required to form a cluster (minPts).

  1. First start with an arbitrary point that has not been visited.
  2. Collect neighborhood of this point using ε (All points which are in ε distance of current point selected are neighborhood).
  3. If neighborhood point is sufficient, then we will make a cluster and mark it as visited else this point is labeled as noise.
  4. If a point is found to be a part of the cluster, then its ε neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all ε neighborhood points. This is repeated until all points in the cluster are determined.
  5. A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.
  6. After discovering every point, the point which is marked as not visited is noise and we will declare them as outliers.

Advantages

- It does not require any priory information on number of clusters in the data, as opposed to k-means.
- It is able to find out arbitrarily size and arbitrarily shaped clusters.
- Random forest

  The random forest algorithm is an ensemble classifier algorithm based on the decision tree model. It generates k different training data subsets from an original dataset using a bootstrap sampling approach, and then, k decision trees are built by training these subsets.

  1. Constructing each decision tree model. Decision tree is created by a C4.5 or CART algorithm from each training subset Si.
  2. In each tree node's splitting process, the gain ratio of each feature variable is calculated, and the best one is chosen as the splitting node. This splitting process is repeated until a leaf node is generated.
  3. Collecting *k* trees into an RF model. Each sample of the

testing dataset is predicted by all decision trees, and the final classification result is returned depending on the votes of these trees.

- PCA

  PCA is a linear transformation method which yields the principle components (direction) which will maximize the variance of the data. It projects the dataset into a feature subspace selection.

  1. Standardize the data.
  2. Obtain the eigenvectors and eigenvalues from the covariance matrix or correlation matrix, or perform singular vector decomposition.
  3. Sort eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalues where k is the number of dimensions of the new feature subspace (k≤d).
  4. Construct the projection matrix W from the selected k eigenvectors.
  5. Transform the original dataset X through WW to obtain a k-dimensional feature subspace Y.

DATASET

Data can be in any form such as, it can comma-separated values, tab separated, unstructured (text), log data, etc. This tool will accept data in any form and it will give you functionality to make it in structured form or if it is already structured, process it for data preparation.

REFERENCES

1. Han J, Kamber M. Data Mining: Concepts and Techniques. 2nd ed. San Francisco: Morgan Kaufmann; 2006.
2. Pyle D. Data Preparation for Data Mining. San Francisco: The Morgan Kaufmann Series in Data Management Systems; 1999.
3. Wu X, Kumar V. Survey Paper on Top 10 Algorithms in Data Mining. London: Springer-Verlag Limited; 2007.
4. Ghosh PK. Big Data ETL and Utilities for Hadoop Map Reduce. Available from: https://www.github.com/pranab/chombo.
5. Extract, Transform, and Load Big Data with Apache Hadoop-By Intel, White Paper, Big data Analytics; 2013.
6. Abiteboul S, Clue S, Milo T, Mogilevsky P, Simeon J. Tools for data translation and integration. IEEE Data Eng Bull 1999;26:3-8.
7. Rana S, Negi GP, Kapoor K. A study over data cleansing and its tools. International Journal of Advanced Research in Computer Science and Software Engineering Research Paper; 2016.
8. Meng X, Bosagh-Zadeh R, Ulanov A, Yavuz B, Pu L, Venkataraman S, et al. MLlib: Machine learning in apache spark. J Mach Learn Res 2016;17(34):1–7.
9. LaValle S, Lesser E, Shockley R, Hopkins MS, Kruschwitz N. MIT Sloan study; results published in MIT Sloan Management Review. Big Data, Analytics and the Path from Insights to Value, December, 21; 2010.
10. Liu A. Apache Spark Machine Learning Blueprints. IBM's Leading Experts in Big Data Analytics; 2016.
11. Bandugula N. The-5-Minute-Guide-Understanding-Significance-Apache-Spark by Senior Product Manager. MAPR; 2015.