HORIZON 2020 - ICT-14-2016-1

AEGIS

Advanced Big Data Value Chains for Public Safety and Personal Security

WP2 - Core Data Value Chain Transformation and Handling Methods

D2.2 - AEGIS Data Value Chain Bus Definition and Data Analysis Methods

Version 1.0

Due date: 31.08.2017 **Delivery Date: 20.09.2017**

Author(s): Maurizio Megliola, Elisa Rossi, Cinzia Rubattino (GFT), Evmorfia Biliri, Michael Petychakis, Spiros Mouzakitis, Christos Botsikas, Giannis Tsapelas (NTUA), Yury Glikman, Fabian Kirstein, Andreas Schramm (Fraunhofer), Gianluigi Viscusi (EPFL), Spyridon Kousouris, Fenareti Lampathaki, Sotiris Koussouris (SUITE5)

Editor: Yury Glikman (Fraunhofer)

Lead Beneficiary of Deliverable: Fraunhofer

Dissemination level: Public **Nature of the Deliverable:** Report

Internal Reviewers: Dimosthenis Tsagkrasoulis (Hypertech)

EXPLANATIONS FOR FRONTPAGE

Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author - if he has generated the Foreground - as well as an Editor - if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the AEGIS DOW. The lead beneficiary partner should be listed on the frontpage as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: These should be a minimum of two persons. They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically this review includes: Identifying typos, Identifying syntax & other grammatical errors, Altering content, Adding or deleting content.

AEGIS KEY FACTS

Topic:	ICT-14-2016 - Big Data PPP: cross-sectorial and cross-lingual data integration and experimentation
Type of Action:	Innovation Action
Project start:	1 January 2017
Duration:	30 months from **01.01.2017** to **30.06.2019** (Article 3 GA)
Project Coordinator:	Fraunhofer
Consortium:	10 organizations from 8 EU member states

AEGIS PARTNERS

Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
GFT	GFT Italia SRL
KTH	Kungliga Tekniska högskolan
UBITECH	UBITECH Limited
VIF	Kompetenzzentrum - Das virtuelle Fahrzeug , Forschungsgesellschaft-GmbH
NTUA	National Technical University of Athens - NTUA
EPFL	École polytechnique fédérale de Lausanne
SUITE5	SUITE5 Limited
HYPERTECH	HYPERTECH (CHAIPERTEK) ANONYMOS VIOMICHANIKI EMPORIKI ETAIREIA PLIROFORIKIS KAI NEON TECHNOLOGION
HDIA	HDI Assicurazioni S.P.A

Disclaimer: AEGIS is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2016) under Grant Agreement No. 732189 and is contributing to the BDV-PPP of the European Commission.

EXECUTIVE SUMMARY

The document presents the AEGIS data harmonisation approach providing the means for collecting, harmonising and processing data and metadata and the libraries and algorithms for knowledge extraction, business intelligence and usage analytics as well as visualisations. The proposed solution is based on the re-use of advanced and proven tools and methodologies, ensuring interoperability with existing approaches.

Table of Contents

EXPLANATIONS FOR FRONTPAGE **2**
AEGIS KEY FACTS **3**
AEGIS PARTNERS **3**
EXECUTIVE SUMMARY **4**
LIST OF FIGURES **6**
LIST OF TABLES **6**
CODE LISTINGS **7**
ABBREVIATIONS **7**
1. INTRODUCTION **8**
1.1. STRUCTURE 8
2. AEGIS DATA VALUE CHAIN BUS CONCEPT **9**
3. HARVESTING **9**
3.1. GOALS 10
3.2. REQUIREMENTS 12
3.3. HARVESTER DESIGN 13
4. DATA AND METADATA HARMONISATION **15**
4.1. GOAL 15
4.2. REQUIREMENTS 15
4.3. DATA HARMONISATION 15
4.4. METADATA HARMONISATION 19
5. KNOWLEDGE EXTRACTION & BUSINESS INTELLIGENCE **20**
5.1. GOALS 20
5.2. REQUIREMENTS 22
5.3. ALGORITHM IMPLEMENTATIONS 22
5.3.1. Basic Statistics Algorithms *23*
5.3.2. Knowledge Extraction Algorithms *27*
5.4. ALGORITHMS RELEVANCE TO AEGIS DEMONSTRATORS 53
6. VISUALISATION **55**
6.1. GOALS 55
6.2. REQUIREMENTS FOR VISUALISATION 56
6.3. VISUALISATION TECHNIQUES 57
6.4. VISUALISATION TOOLS 66
6.4.1. Kibana *66*
6.4.2. Banana *66*
6.4.3. Grafana *66*
6.4.4. HighCharts *67*
6.4.5. D3.js *67*
6.5. AEGIS SPECIFIC REQUIREMENTS 67
6.6. AEGIS REQUIREMENTS MET 68
7. USAGE ANALYTICS **70**
8. CONCLUSION **77**

LIST OF FIGURES

Figure 2-1: AEGIS Data Value Chain Bus 9

Figure 3-1: High-Level Harvesting Architecture 14

Figure 6-1: Example of a line chart 58

Figure 6-2: Example of an area chart 58

Figure 6-3: Example of a streamgraph 59

Figure 6-4: Example of a scatter plot 60

Figure 6-5: Example of a bar chart 61

Figure 6-6: Example of a bullet chart 62

Figure 6-7: Example of a bubble chart 62

Figure 6-8: Example of a hive plot 63

Figure 6-9: Example of a tag cloud 63

Figure 6-10: Example of a tree map 64

Figure 6-11: Example of a chord diagram 65

Figure 6-12: Example of plotting on a map 65

LIST OF TABLES

Table 3-1: Harvesting Requirements 12

Table 4-1: Data and Metadata Harmonisation Requirements 15

Table 5-1: Data Algorithms Requirements 22

Table 5-2: List of proposed methods 28

Table 5-3: Mapping Demonstrators and algorithms (for relevance to algorithms) 53

Table 7-1: List of Usage Analytics Requirements, ID, Description and Previous Requirement Reference 72

Table 7-2: Usage Analytics Tools and their main features 76

CODE LISTINGS

Cose Listing 1: JSON format for 'attribute' 11

ABBREVIATIONS

CO	Confidential, only for members of the Consortium (including the Commission Services)
D	Deliverable
DoW	Description of Work
H2020	Horizon 2020 Programme
FLOSS	Free/Libre Open Source Software
GUI	Graphical User Interface
IPR	Intellectual Property Rights
MGT	Management
MS	Milestone
OS	Open Source
OSS	Open Source Software
O	Other
P	Prototype
PU	Public
PM	Person Month
R	Report
RTD	Research and Development
WP	Work Package
Y1	Year 1

1. INTRODUCTION

The ability to collect and process heterogeneous data and metadata from different sources is probably the most important functionality of the AEGIS platform. It includes the necessity to harmonise data and metadata, so they could be efficiently processed, visualised and could be a good source for the knowledge extraction and business intelligence. The following sections present these topics on the level of the current understanding in the project, which will continue evolving and will be updated in the deliverable D2.3.

1.1. Structure

The section 2 presents the concept of the AEGIS Data Value Chain Bus, which individual elements are explained in the following sections. The section 3 presents the AEGIS approach to harvesting/collecting of data and metadata, the section 4 presents what is planned for data and metadata harmonisation, the section 5 is dedicated to the knowledge extraction and business intelligence, the section 6 on visualisation techniques and tools considered for the project, while the AEGIS approach to usage analytics is presented in the section 7.

2. AEGIS DATA VALUE CHAIN BUS CONCEPT

The high-level AEGIS architecture, presented in D3.1 "Technical and User Requirements and Architecture v1.00", provides an outline of the AEGIS data value chain implementation, from the initial data input to the visualisation of the business intelligence outputs. AEGIS aspires to foster data-driven innovation in the PSPS domains, leveraging the available variety of data sources (open data, proprietary data, sensor data etc.), however this heterogeneity of data in the PSPS domains, both in terms of format and content, imposes strong requirements regarding the semantic enrichment and interlinking of the data that need to be performed so as to render the analysis processes possible. The data processing functionalities that address this need for connecting data to the AEGIS system and subsequently harmonising and semantifying them, making them available to more advanced business intelligence and visualisation tasks, is performed inside the AEGIS data value chain bus. The data value chain bus is in essence a conceptual component that includes (Figure 4-1 in D3.1) the data harvester, the data annotator, as well as a set of services performing very specific rapid transformations on the input datasets. The annotator, which is powered by the AEGIS vocabularies and ontologies, together with the data transformation services (e.g. measurement unit transformations, value replacement, data reduction etc.), act as a data harmonization layer, responsible for ensuring that common analysis and visualization processes can be performed across all available data, by making them compliant with the unified AEGIS data and metadata model.

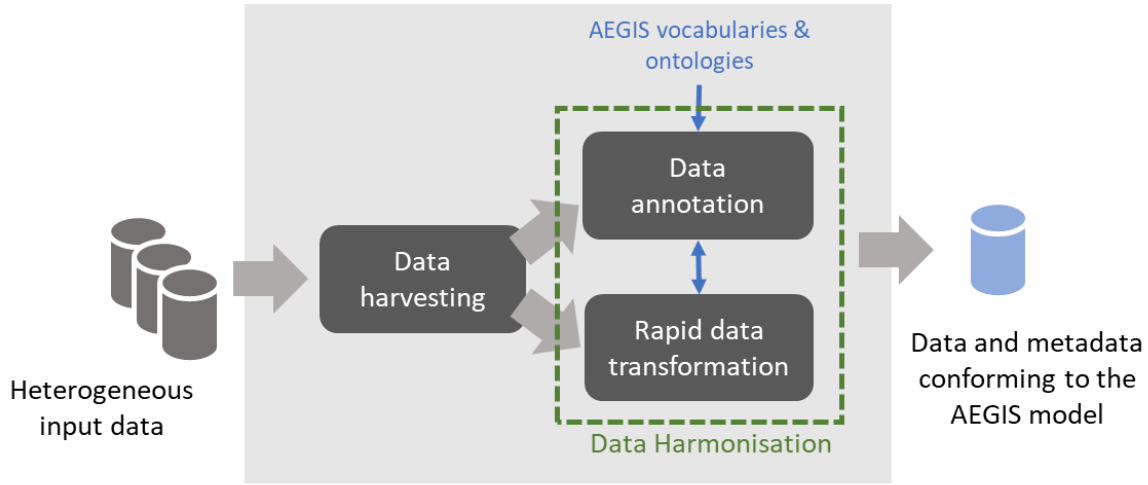

Figure 2-1: AEGIS Data Value Chain Bus

The role of the data harvester is presented in detail in section 3 of the current deliverable, whereas data harmonization is discussed in section 4, however its significance in ensuring actionable insights extraction is highlighted also in other parts of the current document.

3. HARVESTING

Harvesting is the process of extracting or fetching data from a data source. This process includes all necessary means to make data from a specific source available for further processing. Within the AEGIS Data Value Chain this process is part of the Data Acquisition step, where it is the very first sub-step. This definition applies for data and metadata as well.

3.1. Goals

The accessibility to a wide range of data and metadata is one of the key features of the AEGIS platform. Even beyond the initial requirements of the pilot scenarios, the value and sustainability of the platform correlates with the availability of useful data. Hence, the main objective of harvesting in the AEGIS platform is to provide a methodology and mechanisms to access a large variety of data sources.

In order to design the harvesting concept, it is paramount to analyse the characteristics of the data sources. A detailed analysis of data sources was conducted in deliverable D1.1 (Domain Landscape Review and Data Value Chain Definition).

Following the characteristics are defined in detail, which lay the foundation for designing the methodology for the harvesting process. They apply to both metadata and data. These characteristics may affect each other and the agreement on one characteristic may dictate another one. This fact will be respected in the design.

- **Protocol**
  This characteristic aligns with the application layer of the OSI[1] model and describes the methods and interfaces, which are used by the participants in a network. A vast variety of such protocols have established. Yet, only a small subset will be relevant for the scope of AEGIS, since most communication in the Internet, the World Wide Web and intranets rely on widely established protocols. The following basic protocols were identified as the most relevant for the AEGIS platform: HTTP, HTTPS, FTP and SFTP. Furthermore the underlying protocols of popular relational database management systems are relevant, like the PostgreSQL or MySQL client/server protocols. This list may be not complete at this stage and the final implementation should be generic, allowing the extension to further protocols. Especially streaming protocols should be considered, most significant WebSocket (WS).
- **Serialisation format**
  Where the protocol defines on which way the data is transported, the serialisation format states how the data is actually structured. This characteristic can also be referred to as the data format and dictates how to read the data. Basically, the serialisation format can be decoupled from the used protocol – e.g. the same data can be retrieved via HTTP or FTP. This does not necessarily apply for database systems, where the protocol and data formats are tied together in supported clients. Apart from that, AEGIS has to support and deal with a variety of serialisation formats. Based on the findings of D1.1 and the defined scenarios of the client applications the most relevant formats are: tabular formats (XLSX, XLS, CSV, TSV), textual formats (TXT, RTF), XML, Web Map Service (WMS), JSON and relational database formats. As with the protocol, the selection should not be limited and a generic approach will be applied.

- **Semantic**
  The various serialisation formats mainly describe the syntax of the data. In addition some formats do support derivatives, allowing to further describe the meaning of the

[1] Open Systems Interconnection

data, hence the semantic. This may be relevant when the data gets interpreted and harmonised in the next step of the Data Acquisition. Some serialisation formats do not offers such options, like CSV or TXT. Other formats are often used to harness this possibility. Especially the representation of Linked Data is widely established in RDF/XML or JSON-LD, which are based on XML and JSON respectively. Other serialisation formats include to a certain degree some built-in semantic. For example Excel files may define precise data formats for columns, which can be parsed as semantic information, e.g. currency or time. WMS also includes semantic information about the geographical data encoded it. Any semantic information should be considered when harvesting the data.

- **Offline or real-time**

  This characteristic describes how the actual creation time of a dataset relates to its time of availability. Offline describes datasets which are not available at the creation time, but a significant amount of time later. The precise amount depends on the domain and use case. In some applications, several seconds after creation can be seen as offline. This usually happens due to the publication process, used technologies and domain-specific requirements. Real-time datasets refer to availability at the very instance of creation or seconds after it. Although in some applications a delay of hours may be considered real-time too. In the scope of AEGIS only datasets with an availability in the range of single-digit seconds will be understood as real-time data. Which type of data (offline or real-time) has to be harvested, has an impact on the harvesting process, used protocols and other characteristics. AEGIS will deal with both types and therefore both processes have to be considered.

- **Update interval**

  Closely connected to the offline and real-time data characteristic is the update interval of a dataset. Assuming an offline dataset it affects directly the frequency of the harvesting process. Since the update interval differs from source to source the design should consider this a customisable setting to adjust the process to the circumstances. This will avoid unnecessary harvesting or outdated data in the AEGIS repository. As for real-time data, an appropriate adjustment to the interval may save resources.

- **Pull or push**

  A fundamental difference exists in the manner of retrieving the data – either by a pull or by a push mechanism. In the first case, the harvester actively calls for a resource to retrieve the data. The data source provides an address for the data and the control of invocation lays with the harvester. It needs to provide a scheduling mechanism for frequently performing the pull process. This approach has the advantage that the harvesters has the full sovereignty of the process, but needs to define the update interval and control the execution of the process. The push mechanism shifts the control to the provider of the data. The provider gains to a certain degree control over the execution of the harvesting process. The harvester needs to provide an endpoint, which is invoked, whenever the publisher sees fit. This happens usually when new or updated data is available. This demands authentication and authorisation mechanism on the side of the harvester to avoid misuse. The advantage is that the harvester does not need to take care of scheduling. Typically pull mechanisms are used for offline data with a small update frequency, whereas push mechanisms are applied for real-time data, especially in the

context of the Internet of Things. The AEGIS platform should support both paradigms, since both have relevant use cases in the context of the project.

- **Size**
  This characteristic describes the actual size of the data, which needs to be harvested. Since the harvester and the following transformation have to process the data the size depicts an important property. It influences the design, mechanisms and tools which have to be applied. The AEGIS platform will be designed under the premise of working with Big Data. Hence, the harvester design has to consider a very big size, although an individual dataset to be harvested may be not exceeding the megabyte range.

- **Security**
  The interface of a data source may be completely open or require authentication and/or authorisation by the user. The latter needs to be considered in the design of the harvester, since the retrieving mechanism needs to support the applied security methods. The implementation is part of the protocol level and usually tight to it. Several options exist and should be covered by the harvester design. For example, FTP uses a clear-text authentication schema and the database management systems usually support various methods, like user/password or peer authentication. For HTTP, the directory-level authentication htaccess is used, but more common is the transmission of an authentication token (API-key) in the header of the request. Recently the JSON Web Token (JWT) specification is also applied for that purpose. JWTs are small, securely signed data packages, which can be used to share data between different users of a network.

- **Language**
  Datasets may be available in multiple language. How this is presented and implemented by the data source needs to be taken into account for the harvester, since this data should not get lost during the process. How different languages are presented may differ largely, mainly depending on the serialisation format and the semantic. Some formats have multi-lingual capacities by default and well specified, like Linked Data formats. In other formats this is possible as well, but documentation from the data provider is required. This documentation has to state in which way the different translations are provided. For example, in an Excel sheet specific rows or columns are used for representing translations. In addition, it is also possible to distinctly provide a dataset in different languages by distributing it via different URLs. The harvester should consider the different approaches and consider all languages into one dataset when feasible.

3.2. Requirements

The goals and characteristics defined in the previous section leads to requirements for the harvester component, summarized in the following:

Table 3-1: Harvesting Requirements

ID	HV Requirements	Previous Requirements of Reference

HV1	The harvesting process should be able to retrieve data via the following application protocols: HTTP, HTTPS, FTP, SFTP, WSS, WS.	TR1-9
HV2	The harvesting process should support at least the access to the following database management systems: PostgreSQL, MySQL	TR1-9
HV3	The harvesting process has to parse the following tabular file formats: XLS, XLSX, CSV and TSV.	TR1-9
HV4	The harvesting process has to read the following text formats: TXT, RTF, DOC and DOCX.	TR1-9
HV5	The harvesting process has to read the structured formats XML and JSON.	TR1-9
HV6	The harvesting process has to deal with the following Linked Data formats: JSON-LD and RDF/XML.	TR1-9
HV7	The harvesting process should support the processing of the relational data of the databases PostgreSQL and MySQL.	TR1-9
HV8	The harvesting process has to support the guided extraction of semantic information from the data sources.	TR1-9
HV9	The harvesting process has to support the data retrieval from offline and real-time data sources.	TR1-9
HV10	The harvesting process has to be adjustable to the given update interval of a data source.	TR1-9
HV11	The harvesting process should support both, pull and push mechanisms for collecting data.	TR1-9
HV12	The harvesting process should facilitate the fetching of data in the size of several megabyte.	TR1-9
HV13	The harvesting process has to support the data source and protocol specific authentication and authorisation schemas.	TR1-9
HV14	The harvesting process should support the HTTP authentication by API-key and htaccess.	TR1-9
HV15	The harvesting process should be able to process multi-lingual data sources, either encoded in the data itself or by distinct addresses.	TR1-9

3.3. Harvester Design

The previous chapters have shown that the actual retrieval of the data is highly dependent on various characteristics of the precise data source. A high level of abstraction and generalisation is not possible. In many cases the characteristics overlap and are combined within one specific data source. For example, the interface of a database management system combines protocol, serialisation, authentication and other properties. Yet, in many cases a strong separation between the used protocol and the serialisation format can be found – e.g. a CSV file can be distributed via HTTP or FTP. Therefore, in AEGIS the harvesting component will follow the following abstract design:

- Each data source will be associated with a custom component for retrieving the specific data. This component will take into account all precise characteristics of the data source and called "**connector**".
- A connector can be developed completely from scratch in order to allow the most possible flexibility with regards the heterogeneity of the data sources.
- A connector can be composed of existing **artefacts**. An artefact is a library, which already includes logic and functionalities for the previously defined characteristics. For

example, there will be an artefact for dealing with HTTP endpoints or a library for parsing the content of a CSV file.

- For the most common use cases and properties artefacts will be pre-defined and available in the AEGIS project – e.g. for the protocols, the serialisation formats etc.
- A connector needs to implement a certain interface, which defines how the metadata and data needs to be defined and passed to the transformation and harmonisation process.
- Each connector will be registered within the harvesting run-time environment to be employed in a harvesting process.

Hence, in AEGIS a programmatic approach will be followed to support the harvesting of various and multiple data source. Figure 3-1 illustrates the high-level architecture of the harvesting design.

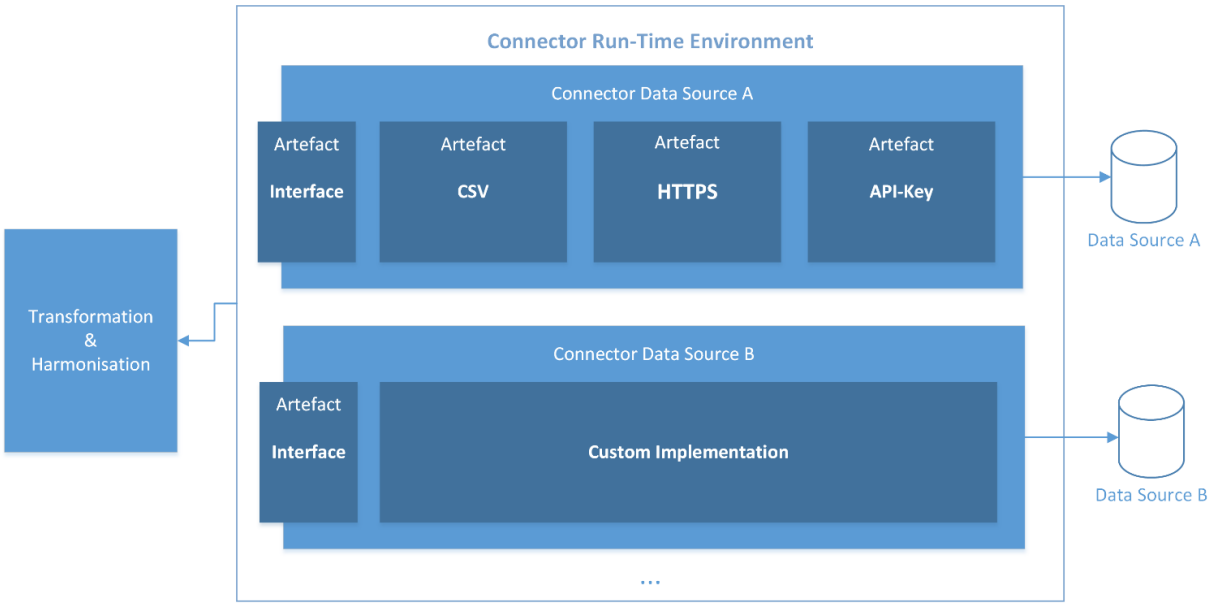

Figure 3-1: High-Level Harvesting Architecture

Most of the defined characteristics can be covered by providing pre-developed artefacts, where some need to be covered by the run-time environment itself. In detail, these are:

- The run-time environment will allow the scheduling or immediate execution of a concrete connector with regards to the sources update interval. This will allow the adoption to the sources update interval. This setting can be defined within the connector and the environment executes with regards to this setting.
- Each connector will run in an isolated environment as a process, which has its own memory and computing power allocated. This will ensure the reliability of the entire harvesting process, even if one connector fails due to a run-time error. In addition, it supports the handling of large files and data. As the update interval the anticipated need of memory can be configured for re-harvesting of static files and data.
- The run-time environment will be publicly accessible via an URL and each connector will be able to allocate sub-routes of that URL and map arbitrary actions to it. Which effectively implies that a connector can expose an endpoint, allowing to interact with data source, using a push mechanism.

To summarize, the introduced design of the harvesting process and related components defines a flexible and sustainable approach to gather data from heterogeneous data sources and allows the AEGIS platform to aggregate a wide variety of data.

4. DATA AND METADATA HARMONISATION

4.1. Goal

The goal of the harmonisation is to decrease the heterogeneity of metadata and data to be managed in the AEGIS platform. It will be done by transforming and refining the collected (harvested) raw data and metadata to the adopted AEGIS data format(s), data structure(s) and use of the selected AEGIS vocabularies. This simplifies further processing and management of data and metadata in AEGIS.

4.2. Requirements

This section presents an initial list of requirements to the data and metadata harmonisation. The requirements are stemming from the technical requirements described in D3.1 "Technical and User Requirements and Architecture v1.00".

Table 4-1: Data and Metadata Harmonisation Requirements

ID	Data and Metadata Harmonisation Requirements	Previous Requirements of Reference
DMH1	AEGIS should be able to harmonise metadata describing data coming from sensor nodes, including wearable devices, smartphones, smart home devices, car sensors and other IoT devices to linked data format.	TR1-9
DMH2	AEGIS has to be able to harmonise tabular data from file formats: XLS, XLSX, CSV and TSV to a selected format in which data will be stored in AEGIS.	TR1-9
DMH3	AEGIS should include a tool for refining and interlinking of metadata.	TR1-9
DMH4	AEGIS should be able to harmonise temporal information, both low-level (e.g. UNIX timestamps) and high-level (e.g. year)	TR1-9
DMH5	The transformation of harvested (meta-)data has to be a scriptable automated process able to work synchronised with the harvester without introducing significant delays.	TR1-9

4.3. Data harmonisation

AEGIS aims to support numerous data sources and heterogeneous data formats. This makes data analysis and visualisation complex tasks. Especially so, when data from different sources and in different arbitrary formats have to be visualised in one chart or somehow analysed together on the request of the end user.

A pragmatic approach to this issue is to perform transformation of raw data to bring it in the format(s) acceptable by the AEGIS data analytics and visualisation tools. The data transformation can be seen as part of the data harvesting process. In case of a lossless transformation, there is no need to keep the original raw data in the AEGIS platform. In other cases, AEGIS should save the original raw data in own data store to enable its processing in future.

The Big Data Cluster provides the means to store and process big data files within the context of the AEGIS project. The Big Data Cluster will be running the software stack provided by Hops Hadoop. The core component of Hops Hadoop is the distributed file system (HopsFS) which is a reliable highly scalable distributed file system that stores massive volumes of data across thousands of machines. On top of HopsFS, multiple processing frameworks such as Spark and Flink can be easily used. Due to the nature of the file system, a user can store any type of data as is without any constraints being placed on how the data is processed. A file in HopsFS can vary in size from kilobytes growing potentially to terabytes of data. Under the hood, HopsFS divides the files into multiple blocks and reliably replicates these blocks across the machines in the cluster.

In the Hadoop/Hops world, users simply store raw data into the system, and then impose the structure at the processing time based on the application requirements. This approach is called Schema-on-Read, an alternative to a well-known approach, Schema-on-Write, which is widely used in traditional data management systems. In the Schema-on-Write approach, the data structure is imposed beforehand at the time of writing the data, that makes it not as agile and as flexible as the Schema-on-Read approach.

Although a user can potentially store any types of data with any kind of format on Hops, there are some considerations need to be taken into account such as how big are the files, what kind of processing and query tool will be used, and what are the performance requirements for read and write.

The following standard file types and specific file formats can be used in Hops/Hadoop:

1. **Standard File Types**. As noted before, a user can store any kind of data on Hops regardless of format. A file could contain text data (such as comma separated files, emails, or log files), structured text data (such as XML files), and binary data (such as image, and videos).
    a) **Text Data** comes in many forms for example comma separated files (CSV), or unstructured data such as emails or server logs. It can very quickly consume considerable amount of storage space on the cluster. That is, storing data as text is not always efficient for example storing integer as text require more space and conversion tools are needed for converting from string to integer and vice verse. Usually, compression is a good idea with such formats but the user must also take into account the usage pattern of the data and the performance of the compression algorithm. In most of the cases, transforming these data into SequenceFile or Avro format is preferable since these format provide better compression support and are splittable.
    b) **Stuctured Text Data** is a more specialized form of text data such as XML and JSON. These formats impose a tricky challenge while storing since they are not splittable by nature, meaning that you cannot split the file into disjoint blocks to

exploit the parallel processing nature of frameworks such as MapReduce, Spark, and Flink. Luckily, transforming such data into Avro format for example provide a more compact and efficient way to process data.

c) **Binary data** such as images, videos or more generally speaking any sequence of bytes can be stored also in Hops. These data can be stored as is, or in a container format such as Avro.

2. **Hadoop File Types.** MapReduce and most of the parallel processing frameworks leverage the idea of data decomposability. That is, decomposing the data into smaller chunks, and then work in parallel on each chunk. There are several Hadoop-specific file formats that were specifically created to work well with such frameworks. These formats support common compression formats and are splittable. They include file-based data structures such as Sequence Files, serialization formats such as Avro, and columnar formats such as Parquet.
    a) **File-based data structures.** These formats include SequenceFiles, MapFiles, SetFiles, ArrayFiles, and BloomMapFiles. The SequenceFile format is the most commonly used format in Hadoop. These formats are well supported within the Hadoop eco-system. The SequenceFile is a flat file consisting of binary key/value pairs. There are three available formats for records stored in SequenceFile; uncompressed key/value records, record compressed key/value records, and block compressed key/value records. The SequenceFiles are well supported within the Hadoop ecosystem, however outside of the ecosystem their support is limited. Also, they are only supported in Java.
    b) **Serialization Frameworks.** Data Serialization is the process of translating data into a byte stream that can be stored or transferred over the network. Similarly, deserialization is the counter process of turning the data back into the original format. The main serialization format used by Hadoop is Writable, but it suffers from a lot of limitations for example it is not easily extendable. There are different serialization frameworks such as Thrift, Protocol Buffers, and Avro that were developed to address the limitations of Hadoop writables.
        - **Thrift** is an interface definition language that was developed by Facebook for scalable cross-language service development. It provides a cross-language serialization that can be used to translate a single interface between different languages. It is used sometimes as a data serialization framework within Hadoop, however, it lacks support for internal record compression, is not splittable, and lacks native support in MapReduce.
        - **Protocol Buffers** was developed by Google, and it is also used for serializing data structures between different languages similar to Thrift. It can be used for data serialization in Hadoop but it lacks support for internal record compression, is not splittable, and lacks native support in MapReduce.
        - **Avro** is a data serialization framework that was developed to address the limitations of Hadoop Writable format. Like Thrift and Protocol Buffers, it uses a language independent format to describe data. However, It has a better native support for MapReduce since Avro data files are compressible and splittable.

    c) Columnar formats. The conventional wisdom was to store the data into a row-oriented fashion. This makes sense for queries reading all the fields from a bunch of rows. But, for queries working on a subset of columns, row-oriented formats won't be that efficient. Also, usually a lot of repetition happen within the values of

a column which impose the need for compression on columns. There are different column-oriented formats such as RCFile, ORC, and Parquet.

- **Record Columnar File (RCFile)** is a column oriented data storage format that was developed to be used by MapReduce applications. It is used in Hive as one of the data storage formats. It writes the data into row splits, and within a split it writes the columns in a column oriented format. It has some limitations in terms of query performance and compression that encouraged the move to a better columnar format such as ORC and Parquet.
- **Optimized Row Columnar (ORC)** is a column oriented data storage format that was developed to overcome the shortcomings of the RCFile format. It provides a lightweight always on-compression, and predicates push down for efficient query processing. It writes files into stripes, where each stripe is independent of all other stripes. Within each stripe, the data is written in a column-oriented fashion. It is also supported by Hive.
- **Parquet** is a column-oriented data storage format that shares the same design goals as ORC, but it is designed to be a more general-purpose storage format for Hadoop. It provides efficient compression that can be specified on per-column level. It also supports complex nested data structures. It is compatible with most of the data processing frameworks in Hadoop eco-system such as Hive, Pig, Impala, and Spark. Also, it can be easily used with Avro and Thrift since they fully support reading/writing Parquet files.

The main input raw data formats, which AEGIS has to support are the following:

- Tabular file formats: XLS, XLSX, CSV and TSV.
- Text formats: TXT, RTF, DOC and DOCX.
- Structured formats XML and JSON
- Linked Data formats: JSON-LD and RDF/XML

The formats for storing data in AEGIS should be well suited for data processing, data visualisation and of cause for storing data. The optimal choice format for storing and processing tabular data in the big data Hadoop infrastructure in AEGIS is Parquet providing good compression and good level of compatibility with data processing frameworks. Some XML and JSON files can be transformed to Parquet as well if they contain tabular data.

Therefore, all harvested in AEGIS tabular, JSON and XML files, containing tabular data, will be transformed and saved in the AEGIS big data store as Parquet files. In case lossless transformation is not possible, the original raw data wiles will be saved in the data store as well.

Linked Data harvested as JSON-LD and RDF/XML files will be stored in AEGIS in a triplestore. Textual files will be saved in the AEGIS platform in their original raw format.

Raw data in XML and JSON format if it cannot be saved a Parquet will be stored in the original format in the AEGIS data store.

Apart, from harmonising data formats with help of transforming raw data to a selected in the AEGIS data format some data elements can be harmonised as well, for example, the date and time writing style. Date and time formats used in data and metadata can vary depending on the

data source or even the concrete dataset. AEGIS will provide a possibility to automatically harmonise date and time data by transforming them to the subset of the ISO 8601[2] format.

In the context of AEGIS this means the format:

- [YYYY]-[MM]-[DD] for calendar dates
- [hh]:[mm]:[ss.sss]±[hh] for time, where "±[hh]" defines the offset from the UTC time zone.

The harmonisation of the date/time data consist from two steps: 1) parsing and recognising the original data/time; 2) transformation of the date/time in the AEGIS format

The functionality will be realised using an existing library for date and time conversion *dateparser*[3]. It is an open source multilingual Python parser for human readable dates published under BSD 3 license).

Another possible harmonisation is the replacement of some data elements by corresponding elements from controlled vocabularies.

4.4. Metadata Harmonisation

All metadata collected by the AEGIS harvester will be transformed to linked data using the DCAT-AP ontology as well as the AEGIS ontology describing the structure and content of data as well as domain specific ontologies as it is explained in the deliverable D2.1[4].

The transformation of metadata as well as transformation of data presented in the previous section will be implemented on the basis of the AEGIS metadata transformation service. The metadata transformation service will work in a pipeline together with the harvester. That means each harvesting pipeline starts with an importer, addressing the specific needs of the source portal, a transformation running a transformation script on each single metadata/data record, and an exporter feeding the AEGIS data store with the transformed (meta-)data record.

The main idea of the current harvesting is that the necessary mapping between two metadata formats between the source and the target repository is outsourced in scripts. The transformer is responsible for the main conversion of the metadata from the source to the target format. Using transformation scripts allows the import of almost all metadata formats, regardless of structure and representation.

To achieve high harmonisation and interlinking of metadata, however, requires manual refinement of metadata. AEGIS will support it with its Data Annotator tool. In order to achieve the namely interlinking the data and connecting it with other data across the web, the AEGIS

[2] http://www.loc.gov/standards/datetime/iso-tc154-wg5_n0039_iso_wd_8601-2_2016-02-16.pdf

[3] https://dateparser.readthedocs.io/en/latest/index.html

[4] D2.1 – Semantic Representations and Data Policy and Business Mediator Conventions

Data Annotator will be using the Named Entity Recognition (NER) service developed in the LinDA project can be of great necessity. It will help users to replace the literals in metadata by appropriate URIs (e.g., the string "Athens" is replaced by an URI that unambiguously refers to the Greek capital, e.g. http://dbpedia.org/resource/Athens).

5. KNOWLEDGE EXTRACTION & BUSINESS INTELLIGENCE

5.1. Goals

The internet provides us with access to vast amounts of information and yet having access to these seas of information has not helped us in finding answers faster to our questions. The amount of information is too vast for humans to process and thus we require an automatic system to process it and provide us with answers to custom questions/queries. However, most of the current Internet information is in plain text, usable by humans, but not by machines. The goal is to provide a system that can ingest unstructured information through advanced natural language processing, semantic analysis, information retrieval, automated reasoning, and machine learning in order to enhance the information in order to be able to answer custom queries.

Although Web Page annotations can facilitate knowledge extraction, these annotations are rare and most often are not complete or do not provide sufficient details. Enhancing this unstructured information can be done manually by domain experts (impractical and un-scalable), by automatic tools (largely undeveloped) and by crowdsourcing (inaccurate, with many possible alternatives). Searching and extracting specific knowledge from unstructured text on the Web can be guided by the use of specific ontologies.

When talking about data analysis, sometimes it can seem tempting to jump straight into the analysis part. Real-world data is typically noisy, enormous in volume, comes from heterogeneous source, each having their own rules of representing data. Before attempting the different analysis tasks, it might make sense to pre-process the data in order to make it more accessible to future tasks. Some of the questions we might ask are: What type of attributes does our data have? What kind of values does each of the attribute have? Are they discrete or continuous values? How are the values distributed? Are there outliers? How similar are the different objects within our data?

According to Han et al.[5] there are various types of attributes, and some of the more prominent are: nominal attributes, binary attributes, ordinal attributes and numerical attributes. Basic statistical descriptions can give us more information about each of the attributes present in our data. If our data happens to contain an attribute temperature, we could for example try to determine its mean (average value), median (middle value) and mode (most common value).

[5] Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.

Knowing such basic statistics regarding each attribute makes it easier to fill in missing values, smooth noisy values, and spot outliers.

We may also want to examine how similar (or dissimilar) data objects are. For example, suppose our data contains patient medical conditions, described by their symptoms. We may want to find the similarity or dissimilarity between them in order to group them together or spot outliers. There are many measures for assessing similarity and dissimilarity. In general, such measures are referred to as proximity measures. Think of the proximity of two objects as a function of the distance between their attribute values, although proximity can also be calculated based on probabilities rather than actual distance.

Web documents use limitless, ever-changing vocabulary and compositional styles to sometimes define approximately the same content and thus can hardly be automatically covered through predefined templates and pattern based extraction rules. Fan et al. [6] propose a two-stage approach to extract the syntactic knowledge and implied semantic. The first step is to extract shallow knowledge from a large collection of documents, followed by a second step where additional semantics are inferred from aggregate statistics on the extracted shallow knowledge. Useful statistics include frequency, conditional probability and normalized pointwise mutual information for specific abstractions.

Besides the text based information, we find on Web Pages, very large datasets, termed as Big Data come with their specific problems. Analysing these datasets require distributed algorithms that can take advantage of multiple machines and algorithm parallelism in order to process the data within acceptable time ranges.

There are many techniques used to reduce the size of datasets and they include sampling, dimensionality reduction techniques, sketching techniques for stream of data, and other transform/dictionary based summarization methods.

One approach would be to summarize the datasets in some fashion so that the resulting dataset is of a more manageable size, but would still retain as much information as possible. The process of summarization might lead to a slightly different data representation. If beforehand a variable might be represented as a single value, after summarizing the dataset, the variable might be represented as lists, intervals, distributions and other similar abstractions.

Consequently, in order to efficiently succeed in all of the aforementioned goals, we propose below a set of algorithms that cover most of the complex aspects of the tasks in hand. These algorithms were selected with the following criteria in mind: a) to adhere to a general approach to cover the AEGIS platform requirements but also a wider variety of problems, b) to have proven their ability and robustness in various domains through the years, and c) to have been implemented in a commonly used software framework or library.

In the following, the requirements of AEGIS platform are mentioned, which constituted the base of the proposed methodology selection. Then, the open-source software libraries are presented where the proposed methodologies are implemented in. Section 6.3 contains a short

[6] Fan, James, et al. "Automatic knowledge extraction from documents." IBM Journal of Research and Development 56.3.4 (2012): 5-1.

description of each selected method, its general applications and its specific applications in the PSPS sector. Moreover, every method's variants and limitations are described.

5.2. Requirements

The current subsection performs a mapping between the requirements of the knowledge extraction and business intelligence components, and the core and demonstrator, functional, non-functional and technical requirements as identified and documented in deliverable D3.1.

Table 5-1: Data Algorithms Requirements

ID	Knowledge Extraction and Business Intelligence requirements	Previous Requirement of Reference
BI_1	Provide simple basic statistics algorithms for data exploration.	CFR4,
BI_2	Provide algorithms for offline analysis of data.	CFR17, CFR4, CFR5
BI_3	Provide algorithms for real time analysis of data.	CFR45, CFR36
BI_4	Provide analysis algorithms for a variety of data formats: geospatial, time series	CFR43, CFR10, CFR28
BI_5	Provide algorithms specialized for the domain of weather and climate data.	CFR23, CFR46
BI_6	Provide algorithms specialized for the IOT domain - wearables and smart-home.	CFR22, CFR41, CFR42, CFR43, CFR15, CFR29-34, CFR49-53
BI_7	Provide algorithms specialized for traffic monitoring and analysis	CFR26, CFR27, CFR39
BI_8	Provide algorithms specialized for healthcare and monitoring.	CFR24, CFR43, CFR28, CFR47,
BI_9	Include tools to compare analysis results among different runs on the same dataset or similar datasets.	CFR18
BI_10	Include methods to combine algorithms (for example sequentially)	CRF40, CRF16
BI_11	Include methods to automatically export analysis results in files	CRF40, CRF16
BI_12	Include methods to automatically trigger online visualisations of analysis results	CRF40, CRF16

5.3. Algorithm implementations

For the next section, the consortium has focused on the following software implementation and libraries:

- Spark MLlib (https://spark.apache.org/mllib/)
- Spark-sklearn (https://github.com/databricks/spark-sklearn)
- Java-ML (https://java-ml.sourceforge.net/)
- Apache Mahout (https://mahout.apache.org/)
- Tensorflow (https://www.tensorflow.org/)

- Keras (https://keras.io/)
- H2O (https://h2o.ai/)
- Torch (https://torch.ch/)
- Deeplearning4j (https://deeplearning4j.org/)
- Custom implementations

5.3.1. *Basic Statistics Algorithms*

5.3.1.1. Basic statistics: Measuring central tendencies

Brief Description

A measure of central tendency is a single value that attempts to describe a set of data by identifying the central position within that set of data. These measures are classed as summary statistics.
The mean (or average) is the most popular and well-known measure of central tendency. The mean is equal to the sum of all the values in the data set divided by the number of values in the data set. Even though it is the value that produces the lowest amount of error from all other values in the data set, in most cases, the mean is actually not a value from the data set. The mean has one main disadvantage: it is particularly susceptible to the influence of outliers.
The median is the middle value for a set of data that has been arranged in order of magnitude. The median is less affected by outliers and skewed data and is a value present in the dataset.
The mode is the most frequent value in our data set. On a histogram it represents the highest bar in a bar chart or histogram. The mode can thus be seen as the most popular option (the most frequent), but as we can easily see, it might also be the case it is not unique and we have a couple of values that conform to this rule.

Purpose

These values can be used to get a feeling of the data and also as part of more complex algorithms.

Software Implementations / Libraries

Any statistics library - Spark MLlib

5.3.1.2. Basic statistics: Measuring dispersion of data

Brief Description

We now look at measures to assess the dispersion or spread of numeric data. The measures include range, quantiles, quartiles, percentiles. Variance and standard deviation also indicate the spread of a data distribution.
Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equal size consecutive sets. There are k−1 k-quantiles. The 4-quantiles are referred to as quartiles and the 100-quantiles are referred to as percentiles.

Variance and standard deviation indicate how spread out a data distribution is. A low standard deviation means that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data are spread out over a large range of values.

Purpose

These values can be used to get a feeling of the data and also as part of more complex algorithms.

Software Implementations / Libraries

Any statistics library - Spark MLlib

5.3.1.3. Basic statistics: Correlation (Pearson's and Spearman's Correlation)

Brief Description

A correlation coefficient measures the extent to which two variables tend to change together. The coefficient describes both the strength and the direction of the relationship.

Pearson product moment correlation evaluates the linear relationship between two continuous variables. A relationship is linear when a change in one variable is associated with a proportional change in the other variable.
Spearman rank-order correlation evaluates the monotonic relationship between two continuous or ordinal variables. In a monotonic relationship, the variables tend to change together, but not necessarily at a constant rate. The Spearman correlation coefficient is based on the ranked values for each variable rather than the raw data.

Spearman correlation is often used to evaluate relationships involving ordinal variables. For example, you might use a Spearman correlation to evaluate whether the order in which employees complete a test exercise is related to the number of months they have been employed.

Purpose

To determine whether there is a statistically significant relation between two variables and how strong the relation is.

Application Examples (from the PSPS domain)

Provided a dataset with measurements on the quality of driving for multiple drivers as well as road quality, weather, length of trip, accident hotspots. Are there any correlation between the quality of data with any of the other features.

Established Variations

- Kendall rank correlation
- Point-Biserial correlation

Software Implementations / Libraries

Spark MLlib

References

- Kendall, M. G., & Gibbons, J. D. (1990). *Rank Correlation Methods* (5th ed.). London: Edward Arnold.

5.3.1.4. Sampling: Stratified sampling

Brief Description

Stratified random sampling is a method of sampling that involves the division of a population into smaller groups known as strata. In stratified random sampling, the strata are formed based on members' shared attributes or characteristics. A random sample from each stratum is taken in a number proportional to the stratum's size when compared to the population. These subsets of the strata are then pooled to form a random sample.

Purpose

Sampling technique with the possibility of providing a desired sample distribution. It also ensures adequate representation of all subgroups.

Application Examples (from the PSPS domain)

Suppose we have a dataset that contains objects with a number of features and one of these features is a k-class feature. This feature could be, for example, types of roads with three classes: highway, national roads, regional roads. We now require a sample with a particular structure, for example we want our sample to contain 35% highway datapoints, 40% national roads datapoints and 25% regional roads datapoints. For this we could apply the stratified sampling technique.

Limitations

This sampling technique cannot be used when the population cannot be split into strata. This technique also requires prior knowledge of strata membership.

Software Implementations / Libraries

Spark MLlib

References

- Thompson, S. K. (2012) Stratified Sampling, in Sampling, Third Edition, John Wiley & Sons, Inc., Hoboken, NJ, USA. doi: 10.1002/9781118162934.ch11

5.3.1.5. Hypothesis testing: Pearson's chi-squared tests for goodness of fit

Brief Description

The chi-square goodness of fit is applied to binned data (data put into classes), but any non-binned data can be transformed into binned data with the use of a histogram or frequency table. The chi-square test statistic are dependent on how the data is binned and test requires a sufficient sample size in order for the chi-square approximation to be valid.

$$\chi^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i$$

After computing the chi value and comparing it with a table value we can decide whether to accept or reject a hypothesis.

Purpose

Hypothesis testing is a powerful tool in statistics to determine whether a result is statistically significant, whether this result occurred by chance or not. The chi-square test is used to test if a sample of data came from a population with a specific distribution.

Typical Examples

Provided a hypothesis and result from an experiment, does the result suggest we should accept or reject initial hypothesis?

Application Examples (from the PSPS domain)

Say we are working on a problem that involves a hospital and the number of patients for each day of the week. We are provided with the distribution of patients for each of the week days(this is our hypothesis). From our measurements, however, we experience 200 patients within a particular week.

	M	T	W	T	F	S	S
Distribution (Hypothesis)	0.1	0.2	0.15	0.15	0.2	0.1	0.1
Experienced patients	20	40	40	30	20	30	20
Expected Patients (Hypothesis)	20	40	30	30	40	20	20

$$\chi^2 = \frac{(20-20)^2}{20} + \frac{(40-40)^2}{40} + \frac{(40-30)^2}{30} + \frac{(30-30)^2}{30} + \frac{(20-40)^2}{40} + \frac{(30-20)^2}{20} + \frac{(20-20)^2}{20}$$

$$\chi^2 = 18.(3)$$

Based on the computed chi value and a predetermined table, this goodness of fit test will tell us whether we should accept or reject our hypothesis based on the experienced data.

Established Variations

- Anderson-Darling Goodness of Fit
- Kolmogorov-Smirnov Test
- Shapiro-Wilk Normality Test
- Probability Plots
- Probability Plot Correlation Coefficient Plot

Software Implementations / Libraries

R, Dataplot, Spark MLLib

References

- Snedecor, George W. and Cochran, William G. (1989), Statistical Methods, Eighth Edition, Iowa State University Press.

5.3.2. *Knowledge Extraction Algorithms*

The algorithms selected for this section belong to the following abstract categories:

A. **Feature extraction, dimensionality reduction and natural language processing**

In machine learning, feature extraction and dimensionality reduction are concepts closely related since both aim to reduce the input features (set of input variables) with the minimum possible loss of information.

Feature extraction methods construct combinations of input variables and practically offer a new -reduced- set of features by transforming the original ones. The desired task can be performed with sufficient accuracy by employing this reduced representation instead of the complete initial dataset. Especially in classification and regression algorithms, a carefully extracted feature set has been proven to increase the learning rates and overall performance in many cases. It is worth mentioning there are simpler algorithms for dimensionality reduction, belonging to the 'feature selection' category, whose purpose is to find a subset of the initial set of features by removing the most redundant or irrelevant ones.

Natural Language Processing (NLP) refers to the application of various computational or machine learning techniques on the analysis and synthesis of human language and speech. Today, the field of NLP have seen a renewed favor mainly due to the rise of speech recognition, voice activated sensors and Chabot technologies.

B. Clustering methods

Clustering is the process of grouping the data patterns based on their feature characteristics, aggregating them according to their similarities. Clustering is performed in an unsupervised manner and allows an input pattern to either strictly belong to one cluster (hard partitioning) or to belong to many clusters, each in a determined degree (soft partitioning).

C. **Classification and regression methods**

Classification and regression are learning techniques to create models of prediction from gathered data. Regression and classification can work on some common problems where the response variable is either continuous or discrete. Classifiers predict the class where belongs an input pattern, while regressors estimates the future value of a continuous variable.

D. Recommendation systems

A recommendation system is an algorithm used to predict what a user may or may not like among a list of given items. Recommendation systems are a pretty interesting alternative to search fields, as they help users discover products or content that they may not come across otherwise. This makes recommendation systems a great part of web sites and services such as Facebook, YouTube, Amazon, and more.

E. Expert Systems

In computational intelligence, an expert system is a knowledge-based system that attempts to emulate the reasoning of a human expert and deduce logical decisions. An expert system's architecture consists essentially of two main components: the knowledge-base, where all the expert logic is aggregated in the form of IF-THEN rules, and the inference engine, which evaluates input data, applies the rules and results into reasonable conclusions. The clear benefit of expert systems is the explicit knowledge representation and the transparency of the inference process which allows for tracing back over the firing of rules that resulted in the assertion.

Table 5-2: List of proposed methods

ALGORITHM		CATEGORIES
FEATURE EXTRACTION-DIMENSIONALITY REDUCTION-NATURAL LANGUAGE PROCESSING		
(1)	**Principal Component Analysis (PCA)**	Feature extraction
(2)	**Recursive feature elimination (RFE)**	Feature selection
(3)	**T-Distributed Stochastic Neighbor Embedding (t-SNE)**	Dimensionality reduction, Clustering
(4)	**Self-Organizing Map (SOM)**	Dimensionality Reduction, Clustering
(5)	**Term frequency-inverse document frequency (TF-IDF)**	Natural language processing

(6) Word2vec	Natural language processing
CLUSTERING METHODS	
(7) K-Means	Clustering
(8) Gaussian Mixtures	Clustering
(9) Hidden Markov Models (HMM)	Clustering, Classification
CLASSIFICATION-REGRESSION METHODS	
(10) Ordinary Least Squares (OLS)	Regression, Feature Selection
(11) Generalized linear models	Classification, Regression
(12) Naive Bayes (NB)	Classification
(13) K-Nearest Neighbors (k-NN)	Classification, Regression
(14) Support vector machines (SVM)	Classification, Regression
(15) Decision Trees (DT)	Classification, Regression
(16) Random Forest (RF)	Classification, Regression
(17) Multi-layer Perceptron (MLP)	Classification, Regression
(18) Recurrent Neural Networks (RNN)	Classification, Regression
RECOMMENDATION SYSTEMS	
(19) Collaborative filtering (CF)	Recommendation systems
(20) Content-based filtering (CBF)	Recommendation systems
EXPERT SYSTEMS	
(21) Fuzzy Systems	Expert systems
(22) ANFIS	Hybrid expert system, Classification, Regression

FEATURE EXTRACTION - DIMENSIONALITY REDUCTION - NATURAL LANGUAGE PROCESSING ALGORITHMS

5.3.2.1. Principal Component Analysis

Brief description

Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It finds a rotation such that the first coordinate has the largest variance possible, and each succeeding coordinate in turn has the largest variance possible.

Purpose

Dimensionality reduction

Typical Examples

Neuroscience, medicine, atmospheric science etc.

Application Examples (from the PSPS domain)

- In large datasets, it is very likely that subsets of variables are highly correlated with each other. The accuracy and reliability of a classification or prediction model will suffer if we include highly correlated variables or variables that are unrelated to the outcome of interest because of over fitting. To overcome this problem, PCA is used selecting a subset of variables together with a low complexity method for classification or regression. Because the subset size is predefined, the optimal number of selected variables is unknown. It can be fixed applying PCA many times to whole dataset and checking the accuracy of a classification or a regression method when employed with the resulted subset of variables (e.g. Demonstrator Scenario 2.1).

- We suppose that the K-means algorithm described in a following paragraph is doing to implemented to cluster a data set obtained by smart home sensors. As known, K-means uses as similarity measure the Euclidean distance. However, in high-dimensional spaces, Euclidean distances tend to become inflated and the data points essentially become uniformly distant from each other. Due to this limitation, PCA is a useful relaxation of k-means clustering

Limitations

PCA is a linear algorithm. It will not be able to interpret complex polynomial relationship between features.

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, H2O, Apache Mahout

References

- Fukunaga, Keinosuke. 'Introduction to Statistical Pattern Recognition'. Elsevier. ISBN 0-12-269851-7 (1990)

5.3.2.2. Recursive feature elimination

Brief description

In recursive feature elimination method, first, the estimator is trained on the initial set of features and weights are assigned to each one of them. The features whose absolute weights are the smallest are pruned from the dataset. The procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

Purpose

Feature selection

Typical Examples

Classification and regression problems.

Application Examples (from the PSPS domain)

- Assumed that we need to estimate the condition of a road using a set created by data gathered during a driver's trip. However, the data set has many features and we don't know which of them we should use to solve the problem. The recursive feature elimination method can be used to drop out the features with useless information. So, without to need to know about what kind is the available data, we can solve a problem applying the recursive feature elimination method. (Demonstrator Scenario 1.1)
- Now assumed that the room temperature should be predicted in a smart home using only the temperature timeseries recorded by a sensor. In a timeseries prediction problem, the lags should be defined. A *n* lag is the temperature value recorded *n* timestamps from the current time. Namely, it should be defined the which temperature values recorded in the past would be used as the inputs of the problem. Firstly, a large lag is selected and all the values recorded later than this lag form the input vector. This is done for every point of the timeseries creating the initial training set. Applying the recursive feature elimination method, the best features of the problem can be estimated. (Demonstrator Scenario 2.2)
- The recursive feature elimination method is commonly used together with SVM

Software Implementations / Libraries

Spark-sklearn

References

- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V., "Gene selection for cancer classification using support vector machines", Mach. Learn., 46(1-3), 389–422, 2002.

5.3.2.3. T-Distributed Stochastic Neighbour Embedding

Brief description

T-Distributed Stochastic Neighbour Embedding (t-SNE) is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets.

Purpose

Dimensionality reduction, Data Visualization

Typical Examples

Text data visualization, image recognition.

Application Examples (from the PSPS domain)

- T-SNE maps multi-dimensional data to two or more dimensions suitable for human observation. For example, word representations capture many linguistic properties such as gender, tense, plurality and even semantic concepts like “capital city of”. Using T-SNE, a 2D map can be computed where semantically similar words are close to each other. This combination of techniques can be used to provide a bird’s-eye view of different text sources, including text summaries and their source material. This enables users to explore a text source like a geographical map.
- The algorithm can be applied to any scenario that pertains vast amounts of social or text data and a rapid visualization of similarity groupings is required.

Limitations

T-SNE is extremely efficient only when reducing dimensionality to 2-D or 3-D, so it is mostly used for visualization purposes.

Software Implementations / Libraries

Spark-sklearn, custom implementations in Java, Python, R

References

- L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245, 2014

5.3.2.4. Self-Organizing Map

Brief Description

A self-organizing map (SOM), also known as Kohonen network, is a popular neural network typically utilized for clustering purposes and dimensionality reduction. SOMs apply competitive (unsupervised) learning in order to produce a low-dimensional (usually 2-D) map of the input space, while trying to preserve its topological properties.

Purpose

Clustering, Dimensionality Reduction, Data Visualization

Typical Examples

High-dimensional data exploration, text clustering, weather prediction, etc.

Application Examples (from the PSPS domain)

- Even if a driving expert would be able to recognize certain driving styles, he could never describe all driving styles worldwide and categorize them in a specific number of classes. It would be much more reliable to extract this knowledge directly from driving data, using a competitive learning technique that clusters the input space depending on similarities.

Therefore, in Demonstrator Scenario 1.3 a self-organizing map could be employed in order to form groups of similar characteristics based on driving habits and behaviors, regional and demographic data, thus creating a geographical representation of driving styles.

- A common application of SOM networks is the grouping of consumer transactions, which results in a mapping of similar consumer behavior that can be easily visualized and function as a compass for future marketing strategy. The same logic could be applied in the Demonstrator Scenario 3.3 where customer habits are to be utilized. This kind of data could form clusters of similar customer mentality and behavior, which would then operate as predictive models for personalized promotional offers.

Established Variations

Growing self-organizing map (GSOM) is a growing variant of SOM in the sense that it starts with a minimal number of nodes and grows new ones based on a heuristic method.

Neural Gas is an algorithm inspired by SOM used for cluster analysis. The name derives from the dynamics of the input vectors during adaptation process, whose distribution resembles a gas flowing within the input space. Usually Neural Gas is applied in speech recognition and image processing where data compression is important.

Limitations

Data should be represented as vectors.

Software Implementations / Libraries

Custom implementations in Java-ML, Tensorflow and Python (SomPy)

References

- Kohonen, Teuvo (1982). "Self-Organized Formation of Topologically Correct Feature Maps". Biological Cybernetics. 43 (1): 59–69.
- T. Kohonen, Self-Organization and Associative Memory. Springer, Berlin, 1984

5.3.2.5. Term frequency-inverse document frequency

Brief Description

Term frequency-inverse document frequency numerical statistic is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Purpose

Text mining, natural language processing

Typical Examples

Search engines

Application Examples (from the PSPS domain)

- Scoring and ranking a documents' relevance. Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 * 4 = 0.12. (Demonstrator Scenario 3)
- Assumed the problem of event detection from social media and we want to find specific events happened in one day. We can apply Tf-idf to several sites and compute the measure of a word for example ‘fire’ in several sites or in social media. (Demonstrator Scenario 2 and 3)

Software Implementations / Libraries

Spark MLlib

References

- Rajaraman, A.; Ullman, J. D. "Data Mining". pp. 1–17. ISBN 978-1-139-05845-2 (2011)

5.3.2.6. Word2vec

Brief Description

Word2vec is used to produce word embeddings. It is a two-layer neural networks that is trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Purpose

Text classification, natural language processing

Typical Examples

Natural language processing

Application Examples (from the PSPS domain)

- Let say we want to identify a word which doesn't belong in a list with several other words. As an example, in the list: ride, come, drive, run, crash, all the words except crash are verbs of transportation, so the answer would be crash (Demonstrator Scenario 3.1)
- Creating a model with word2vec using documents specified by insurance sector and million tweets, it can be improved the language used in advertisements or it can be created new offering strategies.
- Language modeling and feature learning techniques (Demonstrator Scenario 3)

Software Implementations / Libraries

Spark MLlib, H2O

References

- Mikolov, Tomas; Yih, Wen-tau; Zweig, Geoffrey. "Linguistic Regularities in Continuous Space Word Representations.". HLT-NAACL: pp. 746–751 (2013)

CLUSTERING ALGORITHMS

5.3.2.7. K-Means

Brief Description

K-means is a popular algorithm for cluster analysis. It clusters the data points into predefined number of clusters, namely it constructs *k* clusters from *N* observations. The algorithm is commonly employed via an iterative refinement procedure and converge quickly to a local optimum.

Purpose

Clustering

Typical Examples

Entities recognition, grouping events etc.

Application Examples (from the PSPS domain)

Can be applied to any scenario requires typical clustering results. For instance, Demonstrator Scenario 1.3 describes the grouping of data into a number of driving styles, whereas Demonstrator Scenario 3.3 requires a grouping and categorization of customer habits, which would eventually lead to personalized offers.

Established Variations

Fuzzy C-means is a popular variation where each data point belongs to more than one cluster to a fuzzy degree.

Hierarchical variations of k-means attempt to determine the optimum number of clusters automatically starting with a small number and then adding or splitting clusters according to the method's logic.

Limitations

- *k-means converges when the clusters has comparable spatial extent.*
- *k-means does not perform well in high dimensional datasets*

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, Java-ML, Weka, Tensorflow, R, H2O, Apache Mahout

References

- *E.W. Forgy (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics.* ***21****: 768–769.*
- MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297.

5.3.2.8. Gaussian Mixtures

Brief Description

A Gaussian mixture is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

Purpose

Clustering

Typical Examples

Image segmentation, text processing, handwriting recognition, etc.

Application Examples (from the PSPS domain)

- Same as k-means

Limitations

Insufficient input points lead to algorithm divergence.

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, Tensorflow

References

- McLachlan, G.J. (1988), "Mixture Models: inference and applications to clustering", Statistics: Textbooks and Monographs, Dekker

5.3.2.9. Hidden Markov Models

Brief Description

A Markov model is a stochastic model used to describe randomly changing systems where it is assumed that the next state is solely chosen based on the current state and not on the events before it (Markov property). A hidden Markov model (HDD) is a Markov model in which the system being modeled has hidden (unobserved) states, thus making it ideal for modeling complex and unclear behaviors.

Purpose

System modeling

Typical Examples

Pattern recognition in bioinformatics (DNA), speech, handwriting, human behavior, Time series analysis

Application Examples (from the PSPS domain)

In order to recognize activities being performed by smart home residents and wearable devices (Demonstrator 2, scenario 2.1, 2.2, 2.3), mechanisms such as Markov models could be employed to mathematically combine and classify sensor data streams. In these real-world situations, the tasks are usually interleaved, activities are commonly incomplete, and learned models must be adapted for new individuals. Hence, HMMs could offer an ideal solution for activity modelling and recognition.

Established Variations

Poisson HMM is a special case of a hidden Markov model where a Poisson process has a rate which varies in association with alterations among the different states of a Markov model.

Software Implementations / Libraries

Apache Spark MLLIB, Spark-sklearn, Weka, Tensorflow

References

- Baum, L. E.; Petrie, T. (1966). "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". The Annals of Mathematical Statistics. 37 (6): 1554–1563.

CLASSIFICATION/REGRESSION ALGORITHMS

5.3.2.10. Ordinary least Squares

Brief Description

Ordinary least squares is the most common formulation for regression problems. It minimizes the residual sum of squares between the observations in a dataset, and the model responses established the linear combinations between the input data and their corresponding outputs. Mathematically it solves a problem of the form:

$$\min_{w} ||Xw - y||_2{}^2$$

The coefficients *w* are estimated with the singular value decomposition method or with the stochastic gradient descent algorithm that is more suitable for a large scale problem.

The performance of ordinary least squares relies on the independence of the input variables. When the inputs are correlated and have an approximate linear dependence, the least-squares estimate becomes highly sensitive to random noise in the observed response, producing a large variance. To address some of the problems of ordinary least squares generalization methods can be applied.

Purpose

Feature Selection, Regression

Typical Examples

Analysis of variance, measuring accuracy, financial forecasting, etc

Application Examples (from the PSPS domain)

- Assumed that someone need to predict the CO2 and the only available data are the temperature timeseries. He would like to know if this problem could be solved using only temperature as the only input. He should choose a simple method running in short time. He could apply linear regression together with a regularization method to figure out if the predictability of this problem is acceptable. (Demonstrator Scenario 2.2)
- An insurance company tries to predict loss amounts based on the variables obtained by its policyholders' profiles. Because the dataset is high dimensional, it should be applied a method to describe how correlate each variable to the loss amounts. The Lasso regression method can give this information providing a set of weights which indicate the importance of each variable. (Demonstrator Scenario 3)

Established Variations

Ridge regression addresses some of the problems of ordinary least squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares.

Lasso reduces the number of variables upon which the given solution is dependent. For this reason, the Lasso and its variants are fundamental to the field of feature selection. Under certain conditions, it can recover the exact set of non-zero weights. Since reducing parameters to zero removes them from the model.

ElasticNet is a linear regression model combining ridge regression with lasso. This combination allows for learning a sparse model where few of the weights are non-zero like Lasso, while still maintaining the regularization properties of Ridge. *ElasticNet* is useful when there are multiple features which are correlated with one another. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.

Limitations

It performs well for feature selection, but it is not recommended for estimation. The input data should be normalized.

Software Implementations / Libraries

Apache Spark MLLIB, Spark-sklearn

References

- Björck, Åke . Numerical methods for least squares problems. Philadelphia: SIAM, 1996. ISBN 0-89871-360-9.

5.3.2.11. Generalized linear models

Brief Description

Generalized linear models (GLMs) are an extension of linear regression models that allows to model response variables with error distribution other than a normal distribution. The estimation of the model is obtained by maximizing the log-likelihood over a parameter vector. In GLMs, the error variance varies as a function of the mean. In addition, the response distribution is assumed to belong to the exponential family, which includes the Gaussian, Poisson, binomial, multinomial and gamma distributions.

The main components of a GLM are:

- The density function $f(y; \theta, \varphi)$ has a probability distribution from the exponential family parametrized by θ and φ.
- The systematic component η: $\eta = X\beta$, where X is the input matrix.
- The link function g: $E(y) = \mu = g^{-1}(\eta)$ which relates the expected value of the response μ to the linear component η. The link function can be any monotonic differentiable function.

The GLM models are trained using the generalisers (Ridge regression, Lasso and Elasticnet) described in 6.2.3.1

Purpose

Classification, Regression

Typical Examples

Software cost prediction, epidemiology, agriculture, etc.

Application Examples (from the PSPS domain)

- An insurance company needs to estimate the crime rate in a geographical area using data from social media, e-media and demographic data. This problem can be solved using logistic regression. The model learns the data corresponding to the area with known crime rate and then it can predict the crime to another area using data from web (Demonstrator Scenario 3.1)
- Using data from social media it can be obtained information about the mental conditions. Gathering the posts of a user and correlating them with the user medical profile, it can be predicted whether a person is depressed or not. (Demonstrator Scenario 2.1)

Established Variations

Linear regression corresponds to the Gaussian family model: the link function g is the identity and the density of corresponds to a normal distribution. It is the simplest example of a GLM, but has many uses and several advantages over other families. For instance, it is faster and requires more stable computations.

Logistic regression is used for binary classification problems where the response is a categorical variable with two levels. It models the probability of an observation belonging to an output category given the data. The canonical link for the binomial family is the logit function. Its inverse is the logistic function, which takes any real number and projects it onto the [0, 1] range as desired to model the probability of belonging to a class.

Poisson regression is typically used for datasets where the response represents counts and the errors are assumed to have a Poisson distribution. In general, it can be applied to any data where the response is non-negative.

Limitations

Responses must be independent

Software Implementations / Libraries

Apache Spark MLLIB, Spark-sklearn, H20

References

- Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., ”Numerical recipes: The art of scientific computing”, 3rd edn., Cambridge University Press, Cambridge, 2007

5.3.2.12. Naive Bayes

Brief Description

Naïve Bayes (NB) is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes' theorem to compute the conditional probability distribution of label given an observation and use it for prediction.

Purpose

Classification

Typical Examples

Spam filtering, Image recognition etc.

Application Examples (from the PSPS domain)

- Obtained numerical weather prediction by a local meteorological office, the day-ahead weather conditions can be described. Usually numerical weather predictions consist of many variables. Naïve Bayes can be employed to classify the numerical weather predictions and to contribute to form recommendations about the outdoor conditions of the next day (Scenario 2.1)
- Naïve Bayes can be used for traffic incident detection such as accidents, disabled vehicles, spilled loads, temporary maintenance and construction activities, signal and detector malfunctions, and other special and unusual events that disrupt the normal flow of traffic and cause motorist delay. Using social media, smart phones devices and vehicles black boxes GPS data naïve Bayes can estimate various incidents that have recorded. (Demonstrator Scenario 3.2)
- Using several features recorded in real time by wearables, naïve Bayes can be implemented for emotional recognition. The user can record his emotions once per day for one-month period. The data gathered by its wearable is harmonized with the recorded emotions. Then naïve Bayes learn this data set and inform the user about his emotions. (Demonstrator Scenario 2.1)
- Using positioning information obtained from mobile phones or wearable devices and recorded behavioral routines, naïve Bayes can identify irregularity patterns. A large dataset with all above information can be created. Due to every person acts differently, a dimensionality reduction or a feature selection method needs to be applied. In follows, naïve Bayes algorithm can find the conditions where a patient acts irregular. (Demonstrator Scenario 2.3)

Established Variations

- *Gaussian Naive Bayes* is employed when the likelihood of the features is assumed to be Gaussian.
- *Multinomial Naive Bayes* implements the naive Bayes algorithm for multinomially distributed data.
- *Bernoulli Naive Bayes* is applied to data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued variable. Therefore, this class requires samples to be represented as binary-valued feature vectors.
- *Compliment Naive Bayes* estimates the parameters from all classes except the one which it is going to be evaluated.

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, H2O, Apache Mahout, Java-ML, Weka, R

References

- Hastie, Trevor, Robert Tibshirani, and J Jerome H Friedman. The Elements of Statistical Learning. Vol.1. N.p., Springer New York, 2001

5.3.2.13. K-Nearest Neighbors (k-NN)

Brief Description

The ***k*-nearest neighbors algorithm** (***k*-NN**) is a non-parametric method and one of the simplest of all machine learning algorithms, employed mainly for classification and regression problems. The input consists of the *k* closest training examples in the feature space, while the output denotes either a class membership (for classification task) or a property value (for regression task).

Purpose

Classification, Regression

Typical Examples

Face recognition, anomaly detection, etc.

Application Examples (from the PSPS domain)

In data mining, Anomaly Detection is the identification of patterns which repeatedly do not conform to the general logic of a specific dataset (outliers). This methodology is applicable in various fields such as fraud detection, health monitoring, event detection and generally speaking, in cases where a (desired) target class is represented by very few patterns. Outliers can generally be detected by algorithms used for predictions, such as k-NN. In our scenarios, Anomaly Detection could be performed in Demonstrator 2, as well as in Demonstrator 3, as a

means to detect undesired conditions related to health conditions, crime-related data, natural disasters or any other potential threat or risk which is relatively uncommon in given datasets.

Established Variations

Fuzzy k-NN is a variant of the original k-NN algorithm where labeled samples are assigned with fuzzy membership for each class.

Limitations

Performance relies heavily on data, each of the labeled data is given equal weight deciding the class memberships.

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, Java ML, Tensorflow, R

References

- Cover TM, Hart PE (1967). "Nearest neighbor pattern classification". IEEE Transactions on Information Theory. 13 (1): 21–27.
- Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. **46** (3): 175–185.

5.3.2.14. Support vector machines

Brief Description

Support vector machines (SVM) are based on Vapnik's research about learning theory. Support vector machines are based on two functionalities. First, a principle of maximum margin, which is the distance maximizing the separation hyperplane and nearest data points, is performed. A hyperplane is one that separates between a set of objects having different class memberships and is called support vector. Secondly, the input dimension space is transformed into a higher dimensional space, thanks to a kernel function, where a maximal separating hyperplane is constructed. The goal is to transform a complex (non-linear) low dimension problem into a simple (linear) high dimensional problem. The learning step is the optimization of this hyperplane and can be presented like a quadratic optimization problem.

Purpose

Classification, Regression

Typical Examples

Diabetes prediction, Financial timeseries forecasting, computational biology etc.

Application Examples (from the PSPS domain)

- Using vehicle sensor data like X-Y-Z acceleration, X-Y-Z rotation, speed etc, it can be detected the driving behaviour with quite good accuracy. These data set can be enhanced by historic traffic data, weather data or news data to increase the accuracy. Support vector classifiers have good performance to detect such an event. For generalization purposes and for robust fitting, support vector machines should be trained with the cross-validation technique. Cross-validation is an iterative procedure; to each iteration a different partition of the training set is used to measure the accuracy of the model. (Demonstrator Scenario 1.2)
- Support vector machines and generally machine learning can contribute to support individuals belonging in vulnerable groups. Using data from sensors at home it can be detect a distress situation (fall for instance) or a significant change in the habits or behavior of the person which could indicate a loss of autonomy. Remarkable property of support vector machines is that their ability to learn can be independent of dimensionality of feature space (Demonstrator Scenario 2.2)
- An insurance company needs to classify customers based on their habits. Gathering information from smart phones, social media & web, sentiment analysis can be performed showing the customer preferences. Sentiment analysis is a subfield of NLP concerned with the determination of opinion and subjectivity in a text. First step in sentiment analysis is transforming text into format suitable for learning algorithm. Each word will correspond to one dimension and identical words to same dimension. SVM learns the word combinations that correspond to an emotion (Demonstrator Scenario 3.3)

Established Variations

NuSVC has a new parameter ν which controls the number of support vectors and training errors. The parameter $\nu \in (0, 1]$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

Limitations

High algorithmic complexity and extensive memory requirements

Software Implementations / Libraries

Spark-sklearn, spark-svn, Apache Mahout, R, Weka, Java-ML

References

- Cortes, C., Vapnik, V. (1995) "Support Vector Networks", Machine Learning 20 (3): 273

5.3.2.15. Decision Trees

Brief description

Decision tree is built incrementally breaking down a dataset into smaller and smaller subsets that contain instances with similar values (homogenous). The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. A leaf node represents the

decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree. ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one. The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

Purpose

Classification, Regression

Typical Examples

Agriculture, manufacturing and production, astronomy etc.

Application Examples (from the PSPS domain)

- Due to their rule-based architecture, decision trees can detect efficiently events from data without noise. Using streaming data from a vehicle such as X-Y-Z acceleration, speed, and location, a decision tree can detect road damages. (Demonstrator Scenario 1.1)
- Similarly, with the above example, decision trees could be implemented to create alerts for an event. After processing messages from social media and news using a natural language processing method like word2vec, decision trees can established the rules for an event detection (Demonstrator Scenario 2.1)

Limitations

Low generalization ability

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, R, Weka

References

- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, Belmont, CA, 1984

5.3.2.16. Random Forest

Brief description

Random forests (RF) are ensembles of decision trees. Random forests are one of the most successful machine learning models for classification and regression. They combine many

decision trees in order to reduce the risk of overfitting. Like decision trees, random forests handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. Random forests train a set of decision trees separately, so the training can be done in parallel. Each decision tree is constructed subsampling the original dataset (bootstrapping) and considering different random subsets of features to split. Each feature is analyzed using the most discriminative thresholds that increase the information gain. Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data.

Purpose

Classification, Regression

Typical Examples

Medical applications, computer vision, forecasting, fault detection etc.

Application Examples (from the PSPS domain)

- The decision tree drawbacks related to the generalization ability is solved using random forests. Random forests can analyse noisy data and perform classification with great accuracy. Using vehicle data, random forests can estimate damages on a road. Random forests create trees selecting features based on the variable importance, a metric that indicates the potential of each variable (Demonstrator Scenario 1.2)
- Random forests can effectively balance the indoor ambient conditions in a smart house. It can manipulate sensor data, device measurements (CO2, VOC) and energy pricing data to ensure optimal comfort levels and compliance with health requirements. (Demonstrator Scenario 2.2)
- As support vector machines, random forests can detect accurately a distress situation, a cognitive deterioration and frailty status in a smart home environment. Applying data from sensors, smart phone and wearables to random forests, daily living activities can be supervised and emerging incidents can be identified.

Established Variations

Extremely Randomized Trees uses random thresholds for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule.

Gradient Tree Boosting follows a gradient descent like procedure to minimize a differential loss function by adding new weak learners (decision trees).

Limitations

- Gradient Tree Boosting can train one tree at a time, so they can take longer to train than random forests. Random Forests can train multiple trees in parallel.
- Training more trees in a Random Forest reduces the likelihood of overfitting while training more trees with Gradient Tree Boosting increases the likelihood of overfitting.

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, H2O, Java-ML, Weka, R, Apache Mahout

References

- Hastie, T.; Tibshirani, R.; Friedman, J. H. (2009). "The Elements of Statistical Learning", New York: Springer. pp. 337–384. ISBN 0-387-84857-6.

5.3.2.17. Multi-layer Perceptron

Brief Description

The multi-layer Perceptron (MLP) is a popular category of feedforward neural networks able to solve non-linear problems. It consists of more than one layer of nodes, (in contrast to the single layer perceptron) and utilizes a supervised learning technique called backpropagation for training.

MLPs were very popular during the '80s especially in the fields of speech processing and image recognition, but this interest degraded gradually due to the appearance of faster and simpler algorithms (e.g. SVNs). Today, the enthusiasm towards multi-layer networks has returned as an outcome of the success of *deep learning.*

Purpose

Classification, Regression, Deep learning

Typical Examples

Function approximation, image recognition, speech processing, etc.

Application Examples (from the PSPS domain)

- MLPs and their variations should be assigned with tasks suitable for deep learning, such as the road damage classification task in Demonstrator Scenario 1.1, the discrimination of driving styles in *safe* or *unsafe* manner in Demonstrator Scenario 1.2, or the recognition of the alerting conditions in Demonstrator Scenario 2.3.

Established Variations

Convolutional Neural Networks (CNNs) is a class of deep, feed-forward neural networks, whose design emulates the vision processing in living organisms. They consist of an input and an output layer, with multiple hidden layers which are either convolutional, pooling or fully connected. They have wide applications in speech, image and video processing, as well as in recommenders and NLP.

Limitations

Problem of overfitting or underfitting the data

Software Implementations / Libraries

Spark MLlib, Spark-sklearn, Weka, Tensorflow, Theano

References

- Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961
- Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986

5.3.2.18. Recurrent Neural Networks

Brief description

A recurrent neural network (RNN) is any network whose neurons send feedback signals to each other, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences. The decision made by a RNN is based on its previous decision and the current input data. So recurrent networks have two sources of input, the present and the recent past, which combine to determine how they respond to new data, much as we do in life. The RNNs are trained using the backpropagation-through-time algorithm, a variation of the classical backpropagation algorithm which summarizes the standard error function through the time.

Purpose

Classification, Regression

Typical Examples

Language translation, text recognition, timeseries forecasting etc.

Application Examples (from the PSPS domain)

- For sentiment analysis, remembering old or future information and to understand the context is very important. Recurrent neural networks address this issue. They are networks with loops in them, which allows information to persist in memory. But, it can be difficult to train standard RNNs to solve problems that require learning long-term dependencies. This is because the gradient of the loss function decays exponentially with time. However, LSTMs networks are a special kind of RNN, capable of learning long-term dependencies using LSTM units called a 'memory cell'. These cells can maintain information in memory for long periods of time. (Demonstrator Scenario 2.1, Demonstrator Scenario 3.1, Demonstrator Scenario 3.2)

- Despite the binary classification made for sentiment analysis, RNN can solve the opinion mining problem which consists of the emotion, the intensity, and the sentiment obtained by a text. RNN employ a temporal hierarchy with multiple layers operating at different time scales: lower levels capture short term interactions among words; higher layers reflect interpretations aggregated over longer spans of text. (Demonstrator Scenario 2.1, Demonstrator Scenario 3.1, Demonstrator Scenario 3.2)
- Recurrent neural networks can estimate a harsh event using vehicle data come from a smart phone. Positioning data such as X-Y-Z acceleration, X-Y-Z rotation, speed usually is not synchronized when are recorded by smart phone. RNNs are able to mine the information from measurements before and after of a harsh event. (Demonstrator Scenario 1.2)

Established Variations

Restricted Boltzmann machine is a generative stochastic artificial neural network that can learn a probability distribution over its set of inputs. It has two-layer that constitute the building blocks of deep-belief networks.

Long short-term memory RNN (LSTM) can percept the error that is back-propagated through time. It continues to learn over many time steps and create channels to link causes and effects remotely.

Software Implementations / Libraries

H2O, Tensorflow, Torch, Keras, deeplearning4j

References

- J. L. Elman. Finding structure in time. Cognitive Science, 14:179–211, 1990
- S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 1997
- C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

RECOMMENDATION SYSTEMS

5.3.2.19. Collaborative filtering

Brief description

Collaborative filtering (CF) is a technique used by recommender systems It filters information by using techniques involving collaboration among multiple agents, viewpoints, data sources or people. Most collaborative filtering systems apply the so-called neighborhood-based technique. In this approach, a number of individuals is selected based on their similarity to an active individual. A recommendation for the active individual is made by calculating a weighted average of the decision of the selected individuals.

Purpose

Recommendation

Typical Examples

E-service personalization, e-commerce, e-learning, e-government etc

Application Examples (from the PSPS domain)

- Collaborative filtering can provide real time recommendations to drivers with the safest and faster routes. Using historical driving data with information come from social media and breaking news, the optimal route with controlled traffic in the safest area can be estimated. (Demonstrator Scenario 1.1, Demonstrator Scenario 1.3)
- Insurance Product Recommender helps underwriters and brokers identify industry-specific client risks; pinpoint cross-selling and up-selling opportunities by offering access to collateral insurance products, marketing materials, and educational materials that support a complete sales cycle. As more products are sold to an ever-increasing customer base, the recommendations become more reliable, resulting in an exponential increase revenue realization. (Demonstrator Scenario 3.3)

Established Variations

Item-to-item approach is simply an inversion of the neighborhood-based approach using the correlation the individual decisions.

Classification approach. The individuals are grouped using a classification method.

Software Implementations / Libraries

Spark MLlib, Apache Mahout

References

- F. Ricci, L. Rokach and Bracha Shapira, 'Introduction to Recommender Systems Handbook', Recommender Systems Handbook, Springer, 2011, pp. 1-35

5.3.2.20. Content-based filtering

Brief introduction

Content-based filtering (CBF) methods are based on a description of the item and a profile of the user's preferences. In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. A user profile might be seen as a set of assigned keywords (terms, features) collected by algorithm from items found relevant (or interesting) by the user. An item profile is a set of assigned keywords (terms, features) of the item itself. Actual profiles building process is handled by various information retrieval or machine learning techniques. For instance, the most frequent terms in the document describing an item can represent the item's profile. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past. In particular, various

candidate items are compared with items previously rated by the user and the best-matching items are recommended.

Purpose

Recommendation

Typical Examples

E-service personalization, e-commerce, e-learning, e-government etc

Application Examples (from the PSPS domain)

- A smart home recommender system can continuously interpret the user's situation and recommend services that fit the user's needs or habits, i.e. automate some action that the user would want to perform anyway. After a period of learning process, the recommender system would be able to predict what users might want to do next. (Demonstrator Scenario 2.1, Demonstrator Scenario 2.2)

Software Implementations / Libraries

Spark MLlib, Apache Mahout

References

- Aggarwal, Charu C. (2016). Recommender Systems: The Textbook. Springer. ISBN 9783319296579.

EXPERT SYSTEMS

5.3.2.21. Fuzzy Systems

Brief Description

A fuzzy system is a form of artificial intelligence that uses a collection of membership functions (fuzzy logic) and rules (instead of Boolean logic) to reason about data. The rules in a fuzzy system are usually created by experts in a form similar to this: "If x is low and y is high, then z = medium", where x and y are input variables, z is an output variable, low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and medium is a membership function defined on z.

The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. Most tools for working with fuzzy expert systems allow more than one conclusion per rule. The set of rules in a fuzzy expert system is known as the "rulebase" (or knowledge base).

Purpose

Expert Logic implementation

Typical Examples

Robotics, Machine control and automation, “Smart” devices

Application Examples (from the PSPS domain)

- Can be utilized at a higher level, especially in cases where multiple and heterogeneous inputs require different approaches that ultimately need to be combined using expert logic (e.g. Demonstrator scenario 2.1)
- Fuzzy systems can function in parallel to other methods, in order to apply expert knowledge in addition to data driven reasoning.

Limitations

The most common limitation of fuzzy systems, and expert systems in general, is the acquisition and maintenance of real expert knowledge.

Software Implementations / Libraries

Weka(furia), custom implementations in Java (jFuzzyLogic, FuzzyLite), Python (skfuzzy, fuzzython), R(frbs)

References

- *Zadeh, Lotfi A. (1965). "Fuzzy sets". Information and Control.* ***8*** *(3): 338–353*
- *Zadeh, Lotfi A. (1973). "Outline of a new approach to the analysis of complex systems and decision processes". IEEE Transactions on Systems, Man and Cybernetics.* ***1****: 28–44*

5.3.2.22. ANFIS

Brief Description

An **adaptive neuro-fuzzy inference system** (**ANFIS**) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. This technique integrates both neural networks and fuzzy logic principles, so that it captures the benefits of both in a single framework. The learning procedure aims to create a set of fuzzy IF–THEN rules as its inference system capable to approximate nonlinear functions.

Purpose

Classification, Regression, Knowledge extraction

Typical Examples

Engineering applications, weather forecasting, image processing, etc.

Application Examples (from the PSPS domain)

- ANFIS could be employed as a decision-making system that combines results/data from various heterogeneous subsystems in order to provide a useful recommendation such as a health-related direction or an explicit alert, as described in Demonstrator Scenario 2.1, 2.2 and 2.3. The same scenarios could use ANFIS for prediction, trained by historic medical information and expecting it to foresee the future and create respective alarms, which in turn lead to proactive actions.
- In addition, ANFIS could be used as an inference machine that learns from data and builds up a representative set of knowledge rules. A set of explicit rules derived directly from data would definitely offer an added value to the overall system.

Established Variations

SuPFuNIS is a subsethood-product fuzzy neural inference system which has the flexibility to handle both numeric and linguistic input variables simultaneously.

Limitations

Requires sufficient data for training. Requires many experiments for parameter adjustment and finetuning

Software Implementations / Libraries

Custom implementations in Java, Python, R

References

- *Jang, J.-S.R. (1993). "ANFIS: adaptive-network-based fuzzy inference system". IEEE Transactions on Systems, Man and Cybernetics.* ***23*** *(3)*
- Jang, Sun, Mizutani (1997) – Neuro-Fuzzy and Soft Computing – Prentice Hall, pp 335–368, ISBN 0-13-261066-3

5.4. Algorithms Relevance to AEGIS Demonstrators

The following table presents a first approach of a Demonstrators and Algorithms mapping, given the descriptions of scenarios in Deliverable 7.1. Each description includes a set of input data and a target goal, the combination of which suggests one or more data analysis tasks. A set of algorithms is then proposed, according to each task, that cope with it in the most efficient way.

Table 5-3: Mapping Demonstrators and algorithms (for relevance to algorithms)

	Input Data	Purpose	Possible Task(s)	Suggested Algorithms
Demonstrator_1: Automotive and Road Safety Data				

Scenario 1.1: Broken Road Indicator	Vehicle sensors, traffic data, map data	Route Safety	Route Classification	k-NN, SVN, MLP, CF, CBF, GLM
Scenario 1.2: Safe Driving Indicator	Vehicle sensors, traffic data, weather data, news data, accident types & frequency data	Driver Safety index	Pattern Recognition/ Driving Classification	k-NN, SVN, MLP, RF, NB
Scenario 1.3: Regional Driving Style Risk Estimator	Regional Driving styles, events from media	Map city safety challenges	Mapping / clustering	SOM, CF, CBF, NB, k-means
Demonstrator_2: Smart Home and Assisted Living Demonstrator				
Scenario 2.1: Personalised Recommendations App for High-Risk Population Groups	alerts and messages from media, events, health records, crime per location, road accidents, weather data, public health statistics, health regulations, wearable sensors, personal records	health care directions	Feature selection/ Recommender/ Decision making	HMM, ANFIS, GLM, NB, RNN, CF, CBF
Scenario 2.2: Smart Home Automation App for Security and Well-being enhancement	Interior sensor data, exterior conditions, energy consumption and pricing, VOC & CO2 data, crime & occupancy data	Personalized Decision making / machine adjustment	Feature selection/ Recommender/ Decision making/ Classification	K-Means, Fuzzy Systems, HMM, Naive Bayes (NB), OLS, GLM, CBF
Scenario 2.3: Monitoring and	interior sensor data, exterior	Personalized, monitoring	Feature selection/	k-NN, NB, SVN, RF,

Alert Services for social care services providers	conditions, GPS, wearable sensors, Routine profile	and alert services	Decision making/ Anomaly Detection / Classification	MLP, ANFIS
Demonstrator_3: Insurance Demonstrator				
Scenario 3.1: Risk impact analysis	social media, web trends, weather data, web news, natural disasters	Identify potential risk or threat in specific regions	Feature selection/ NLP based on rules	Word2vec, RF, GLM, Fuzzy Systems, RNN
Scenario 3.2: Personalized Early Warning System for Asset Protection	social media, web trends, weather data, web news, natural disasters, GPS data,	Personalized exposure to risk or threat	Feature selection/ NLP based on rules / recommender	Word2vec, RNN, CF, CBF
Scenario 3.3: Personalized commercial offering	smart home data, wearables, smart phones, social media, web trends	Personalized offers	Feature selection/ Clustering	SOM, k-means, CF, CBF, NB

6. VISUALISATION

Predictive modelling and other kinds of advanced analytics are done with powerful software built specifically for running complex algorithms on large data sets. Yet, the ability to provide useful information in an easily and quickly perceivable manner also relies heavily on more humble data visualization tools. Far from being a bit player in analytics applications, data visualization fills several crucial roles throughout the process. From initial data exploration to development of predictive models to reporting on the analytical findings the models produce, data visualization techniques and software are key components of the data scientist's toolkit. Visualization helps expose patterns over time as well as patterns between different variables. On its own, the data doesn't mean much. It needs context, and that's what data visualization can provide.

6.1. Goals

Visualisation allows end users to access business intelligence and analytics data in eye-catching and easy-to-understand formats. Visualizing data can simplify the process of analysing big data

sets. Data analysts might pull out some specific variables into a graph to see if there's any correlation between them, or chart basic summary statistics (including for example mean and median averages, data spread and standard deviation metrics) to get a sense of the scope of the data. Advanced data visualization offers new ways to view data, which, when used correctly, they can streamline the visual presentation of large, complex sets of data for better business decision making and can deliver business insights to users faster than they can get it with traditional BI tools. Data visualization tools enable users to view and manipulate data in a more instinctive way than they can with traditional reporting and analytics technologies. Exploring the data visually gives them a better idea of where to focus their attention when building analytical models than they could get by looking at a giant spreadsheet. By displaying data graphically for example, users can see if two or more variables correlate and determine if they are good candidates for further in-depth analysis.

Nevertheless, advanced visualization tools are often too complex to deploy and use, requiring support from outside consultants or the services of internal data scientists. Many smaller businesses, and even some larger ones, may not have the resources to invest in the needed skills. Towards this end, the scope of the visualisation services to be offered by AEGIS, is to simplify the visualisation process especially for users lacking experience and expertise in data science, while unlocking for them the visualisation potential.

6.2. Requirements for Visualisation

The visualisation approach to be followed by the AEGIS consortium, needs to safeguard that at minimum the following main generic requirements should be supported:

1. **Minimum database expertise**. Even though related more to the analytics components of the platform providing their results to the visualisation component of the platform, the AEGIS platform may require IT expertise in order to be set up and to connect to the data sources required, yet, when offered to the end users, the platform and the visualisation tool should provide varying degrees of simplicity when it comes to writing own queries or using natural language syntax.
2. **Support for multiple databases and data types**. Again, even though related more to the analytics components of the platform providing their results to the visualisation component of the platform, the AEGIS platform should be able to act as a unified front end to multiple databases and data types. By providing support for a variety of data, the AEGIS visualisation component being responsible for take care building the visual representation, can provide immediate payback against fast-growing Big Data stores (with a single query spanning multiple databases and data types) as well as new insights and ways to leverage data.
3. **Visualization kinds supported**. Many users have standardized on a certain way to look at key data. The AEGIS platform should safeguard - to the extent possible - that the visualisation component can render data in as many ways as possible, since different visualization methods hold the potential to unlock new insights.
4. **Exporting capabilities**. The AEGIS platform users need to be presented with the option to also export the results of their analysis and their visualisations, so that they can be in turn consumed by external applications and/or entities. Key options should include not just a variety of flat graphic formats (CVS, JPEG, PDF) but also code snippets that can be dropped directly onto webpages, incorporated into other apps via open APIs, and rendered in the best way possible on both desktop and mobile devices. The main advantage of integrating code snippets, is that far from simply rendering a visualization,

they can maintain their connections to the live data sources referenced in the query, allowing them to change on the fly as source data changes.

6.3. Visualisation Techniques

Today's data visualization tools go beyond the standard charts and graphs used in Excel spreadsheets, displaying data in more sophisticated ways such as infographics, dials and gauges, geographic maps, sparklines, heat maps, and detailed bar, pie and fever charts. The images may include interactive capabilities, enabling users to manipulate them or drill into the data for querying and analysis. Indicators designed to alert users when data has been updated or predefined conditions occur can also be included.

Six traits that separate advanced data visualization from static graphs include:

1. Dynamic data: Dynamic data is the ability to update visualizations as data changes in sources such as databases.
2. Visual querying: With visual querying you can change the query by selecting or clicking on a portion of the graph or chart (to drill down, for example).
3. Linked multi-dimensional visualization: With multi-dimensional linking, selections made in one chart are reflected as you navigate into other charts.
4. Animation: Visualization animation is a technique used, for example, to show changes over time, in relationship to other variables.
5. Personalization: With personalization you can give power users an in-depth view and newbies a simpler view, and you can also control access to data based on user- and role-based access privileges.
6. Actionable alerts. Alerting safeguards that the user can set thresholds and parameters that trigger messages whether the user is interacting with reports or not.

A subset of the visualisation techniques that could be employed in the context of AEGIS include:

(Multi-Series) Line Charts: A line chart or line graph is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. It is a basic type of chart common in many fields. It is similar to a scatter plot except that the measurement points are ordered (typically by their x-axis value) and joined with straight line segments. Line Charts, or Fever Charts are used for data that changes continuously, like stock prices. They allow for a clear visual representation of a change in one variable over a set amount of time - thus the line is often drawn chronologically. In these cases they are known as run charts.

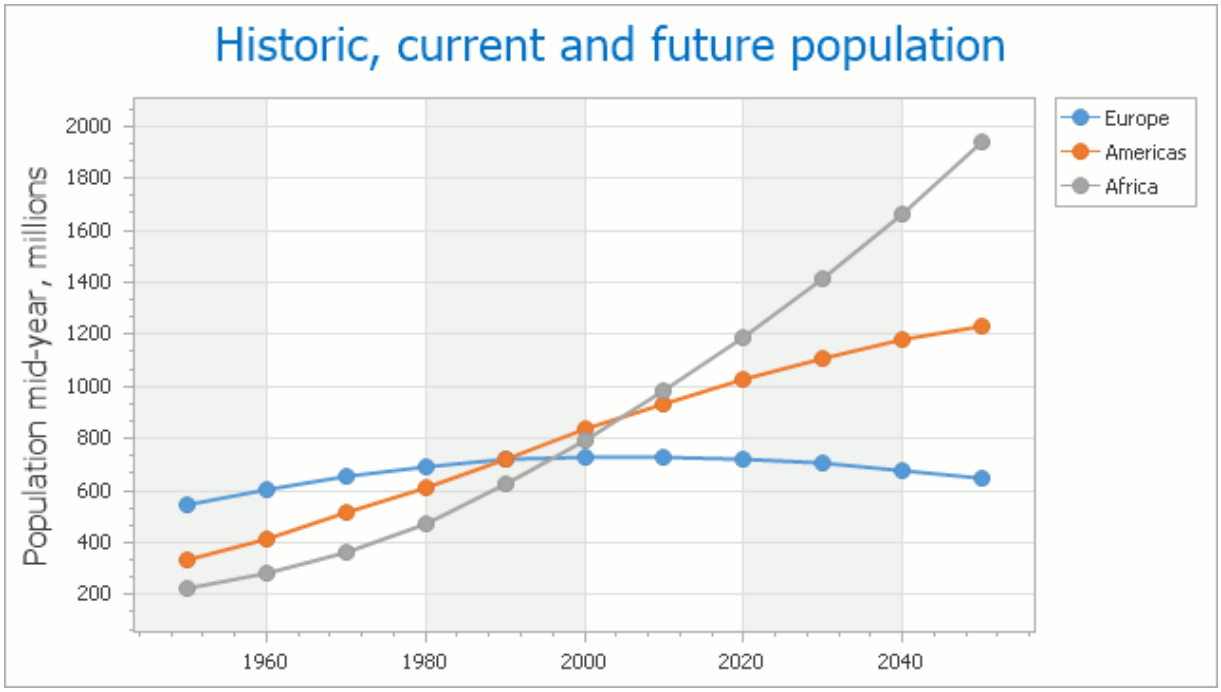

Figure 6-1: Example of a line chart

(Multi-Series / Bivariate / Stacked) Area Charts: An area chart or area graph displays graphically quantitative data. It is based on the line chart. The area between axis and line are commonly emphasized with colors, textures and hatchings. Commonly one compares with an area chart two or more quantities.

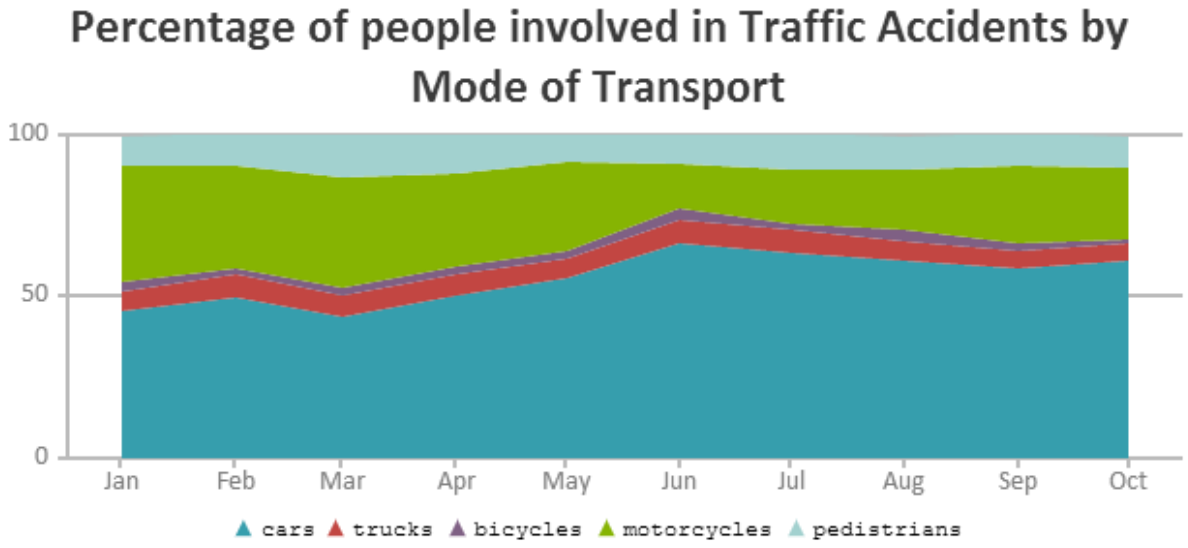

Figure 6-2: Example of an area chart

Steamgraphs: A streamgraph, or stream graph, is a type of stacked area graph which is displaced around a central axis, resulting in a flowing, organic shape.

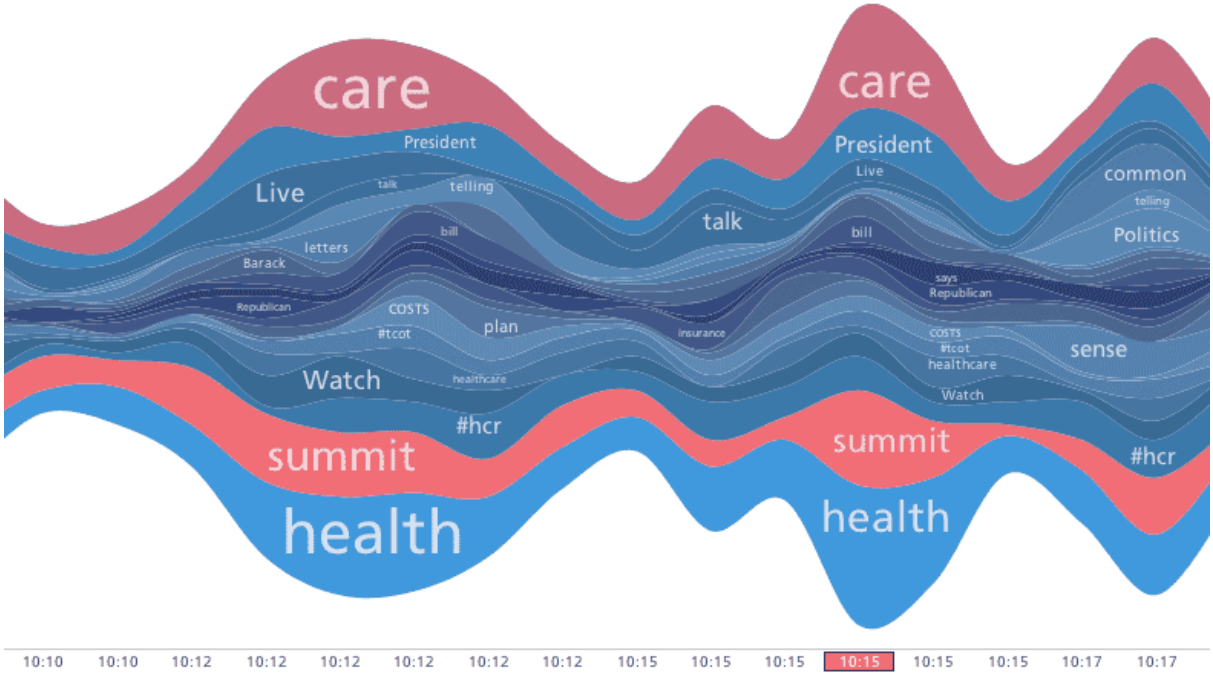

Figure 6-3: Example of a streamgraph

Scatterplots: A scatter plot (also called a scatter graph, scatter chart, scattergram, or scatter diagram) is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are color-coded, one additional variable can be displayed. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

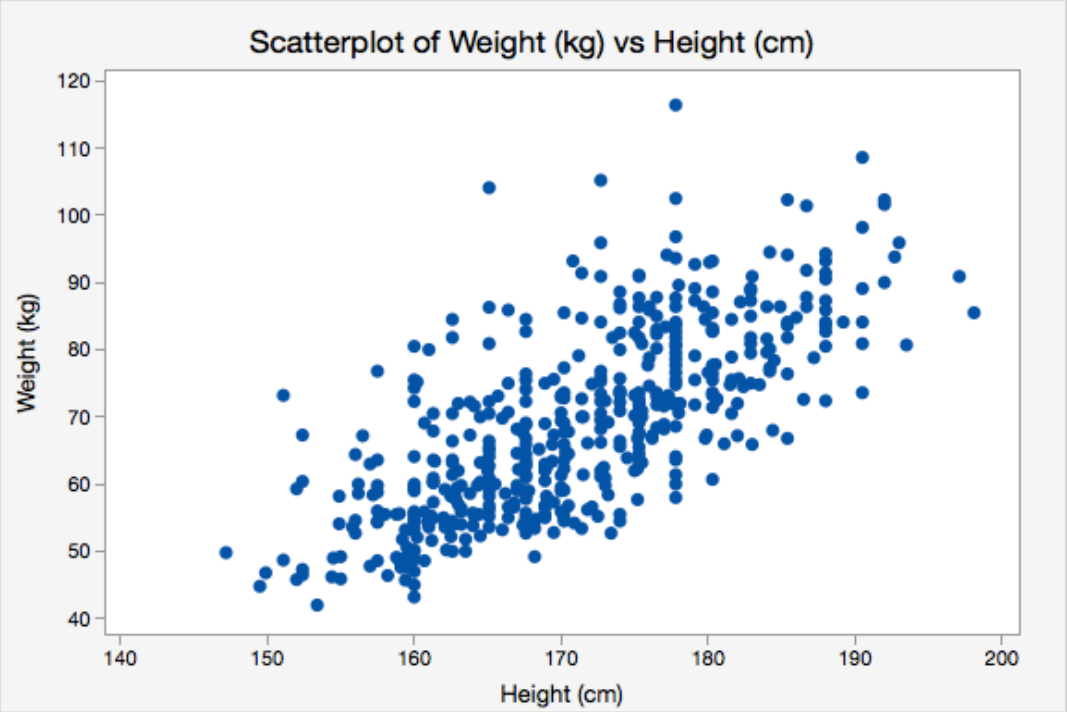

Figure 6-4: Example of a scatter plot

(Grouped / Stacked / Normalised) Bar Graphs: A bar graph is a pictorial rendition of statistical data in which the independent variable can attain only certain discrete values. The dependent variable may be discrete or continuous. The most common form of bar graph is the vertical bar graph, also called a column graph. In a vertical bar graph, values of the independent variable are plotted along a horizontal axis from left to right. Function values are shown as shaded or colored vertical bars of equal thickness extending upward from the horizontal axis to various heights. In a horizontal bar graph, the independent variable is plotted along a vertical axis from the bottom up. Values of the function are shown as shaded or colored horizontal bars of equal thickness extending toward the right, with their left ends vertically aligned.

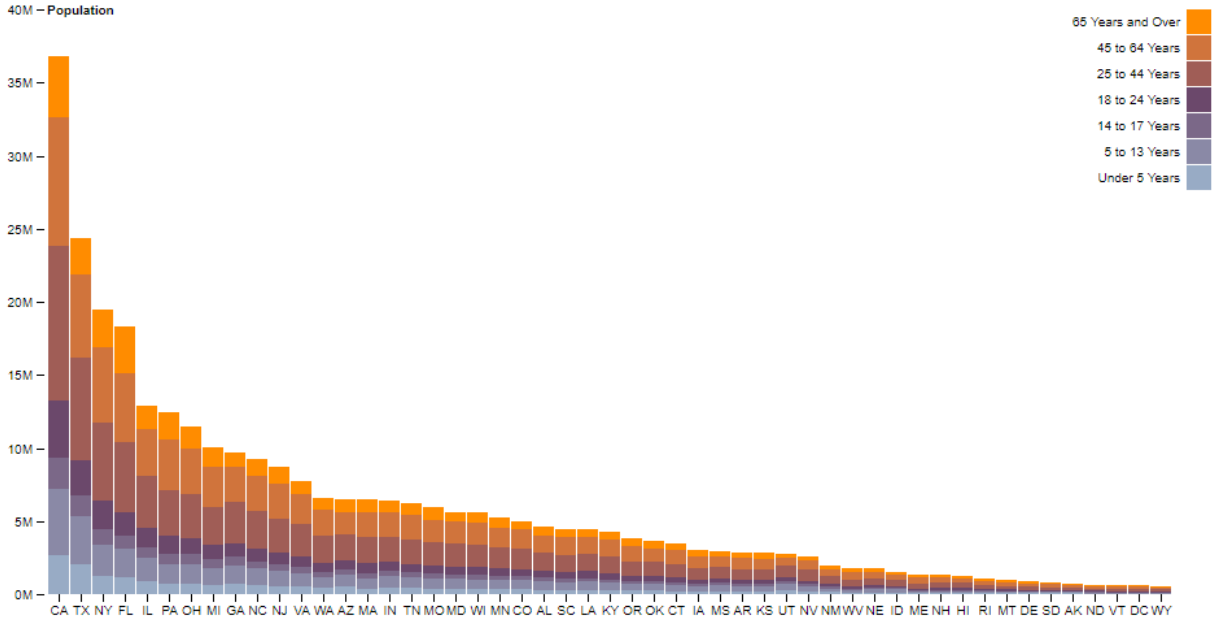

Figure 6-5: Example of a bar chart

Histograms: A histogram is a display of statistical information that uses rectangles to show the frequency of data items in successive numerical intervals of equal size. In the most common form of histogram, the independent variable is plotted along the horizontal axis and the dependent variable is plotted along the vertical axis. The data appears as colored or shaded rectangles of variable area.

Pie Graphs: A pie graph (or pie chart) is a specialized graph used in statistics. The independent variable is plotted around a circle in either a clockwise direction or a counterclockwise direction.The dependent variable (usually a percentage) is rendered as an arc whose measure is proportional to the magnitude of the quantity. Each arc is depicted by constructing radial lines from its ends to the center of the circle, creating a wedge-shaped "slice". The independent variable can attain a finite number of discrete values (for example, five). The dependent variable can attain any value from zero to 100 percent.

Bullet Charts: Seemingly inspired by the traditional thermometer charts and progress bars found in many dashboards, the bullet graph serves as a replacement for dashboard gauges and meters. Bullet graphs were developed to overcome the fundamental issues of gauges and meters: they typically display too little information, require too much space, and are cluttered with useless and distracting decoration. A variation on a bar chart, bullet charts compare a given quantitative measure (such as profit or revenue) against qualitative ranges (e.g., poor, satisfactory, good) and related markers (e.g., the same measure a year ago).

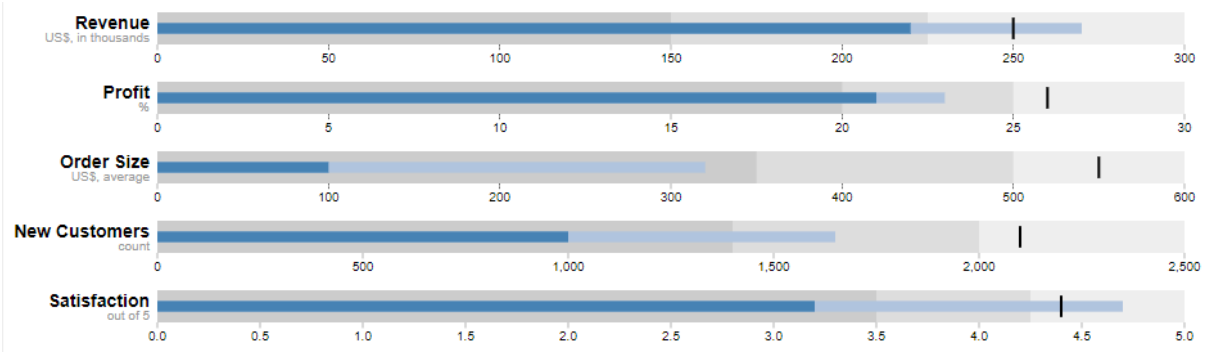

Figure 6-6: Example of a bullet chart

Bubble Graphs: Bubble charts encode data in the area of circles. Although less perceptually-accurate than bar charts, they can pack hundreds of values into a small space. A bubble chart is a type of chart that displays three dimensions of data. Each entity with its triplet (v1, v2, v3) of associated data is plotted as a disk that expresses two of the vi values through the disk's xy location and the third through its size. Bubble charts can facilitate the understanding of social, economic, medical, and other scientific relationships. Bubble charts can be considered a variation of the scatter plot, in which the data points are replaced with bubbles.

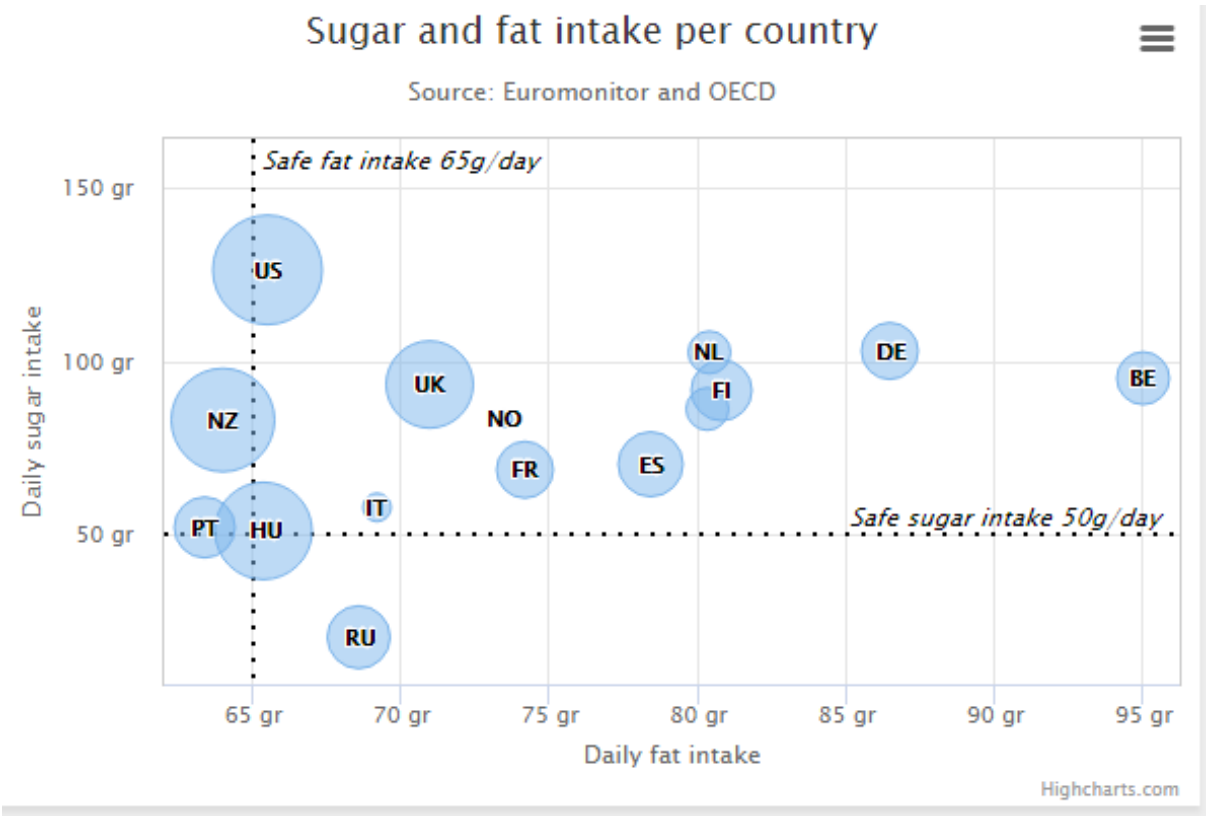

Figure 6-7: Example of a bubble chart

Hive Plots: Many methods of graph drawing, such as force layouts, do not assign intrinsically-meaningful positions to nodes: the position is only approximate, in the hope that related nodes appear nearby. While intuitive, these methods arguably make poor use of the most effective visual channel (that is, position). Hive plots define a linear layout for nodes, grouping nodes by

type and arranging them along radial axes based on some property of data. The explicit position encoding has the potential to better reveal the network structure while communicating additional information. Hive plots can also be extended to show aggregate relationships.

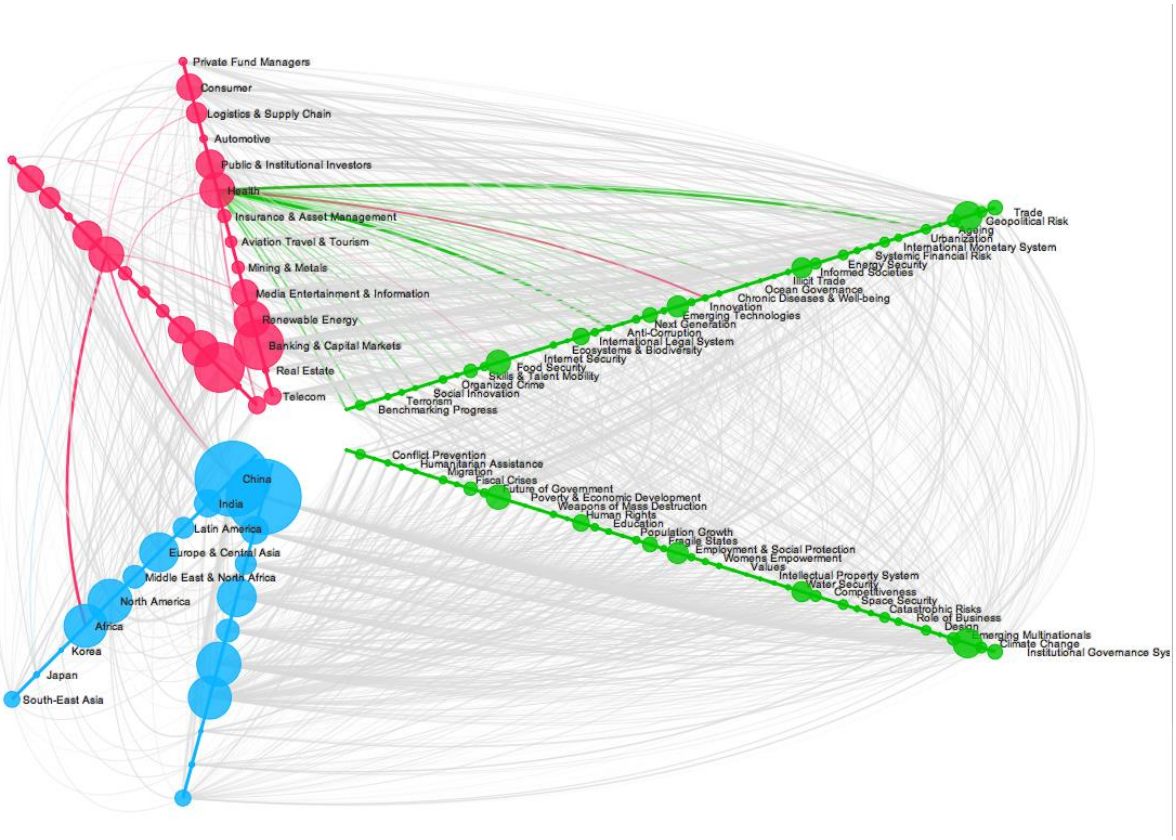

Figure 6-8: Example of a hive plot

Tag Clouds: A tag cloud (word cloud, or weighted list in visual design) is a visual representation of text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color. This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence.

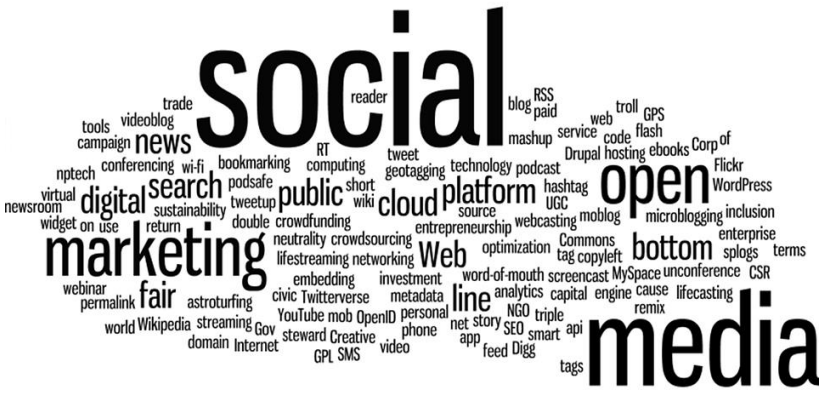

Figure 6-9: Example of a tag cloud

Treemaps: In information visualization and computing, treemapping is a method for displaying hierarchical data using nested figures, usually rectangles. Treemaps display hierarchical (tree-structured) data as a set of nested rectangles. Each branch of the tree is given a rectangle, which is then tiled with smaller rectangles representing sub-branches. A leaf node's rectangle has an area proportional to a specified dimension of the data. Often the leaf nodes are colored to show a separate dimension of the data. When the color and size dimensions are correlated in some way with the tree structure, one can often easily see patterns that would be difficult to spot in other ways, such as if a certain color is particularly relevant. A second advantage of treemaps is that, by construction, they make efficient use of space. As a result, they can legibly display thousands of items on the screen simultaneously.

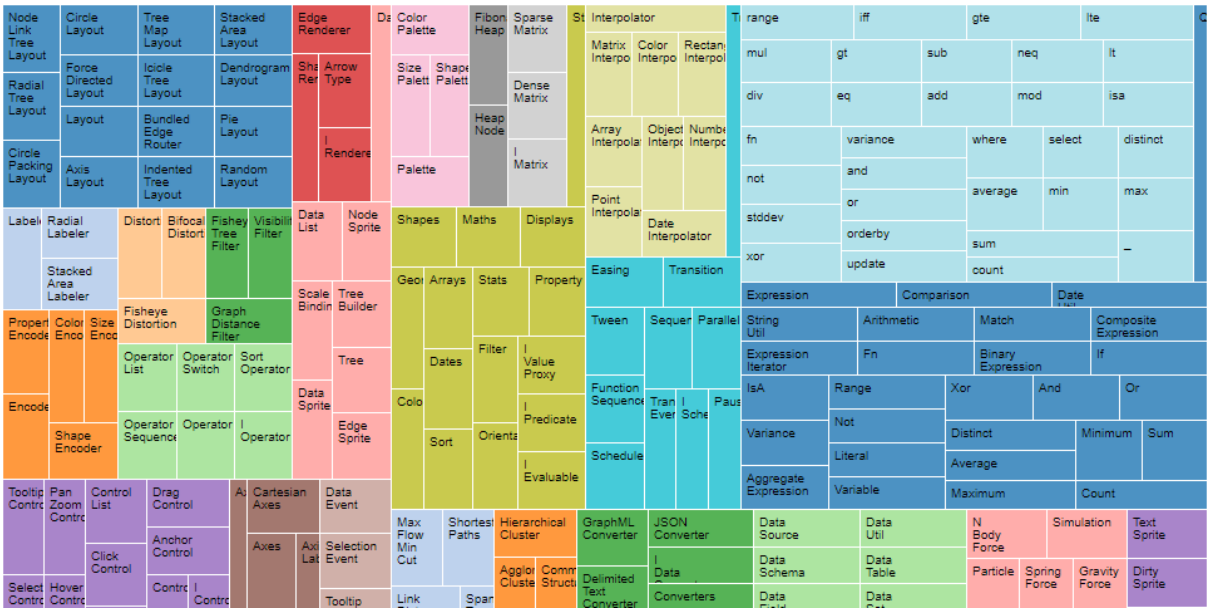

Figure 6-10: Example of a tree map

Chord Diagrams: A chord diagram is a graphical method of displaying the inter-relationships between data in a matrix. The data is arranged radially around a circle with the relationships between the points typically drawn as arcs connecting the data together. The format can be aesthetically pleasing, making it a popular choice in the world of data visualization.

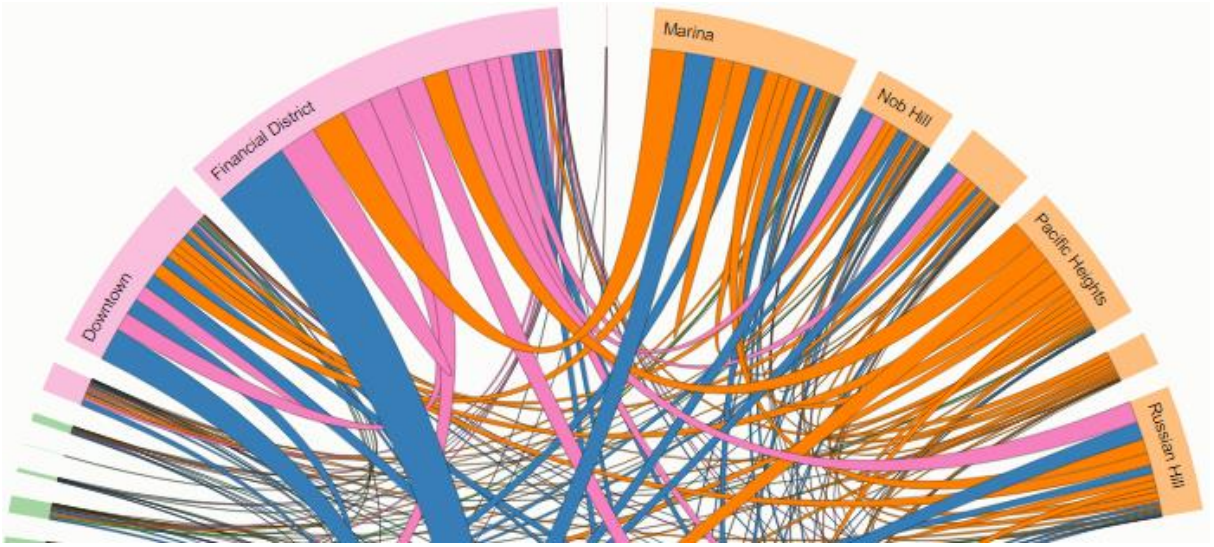

Figure 6-11: Example of a chord diagram

Heat Maps: A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets. There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be would be much harder to understand if presented numerically in a spreadsheet.

Plotting on Maps: A very common advanced visualisation is plotting on maps, where the user has the option to use a map as a bottom layer and add additional layers of visualisation on top, including for example bubble charts or heat maps or tag clouds to illustrate geo-referenced analysed data sets. Mapping data can help users discover geographical trends that may have gone unnoticed in traditionally analyzed data sets. Users can utilize these visualizations to pinpoint geographic strengths and weaknesses and find ways to address them.

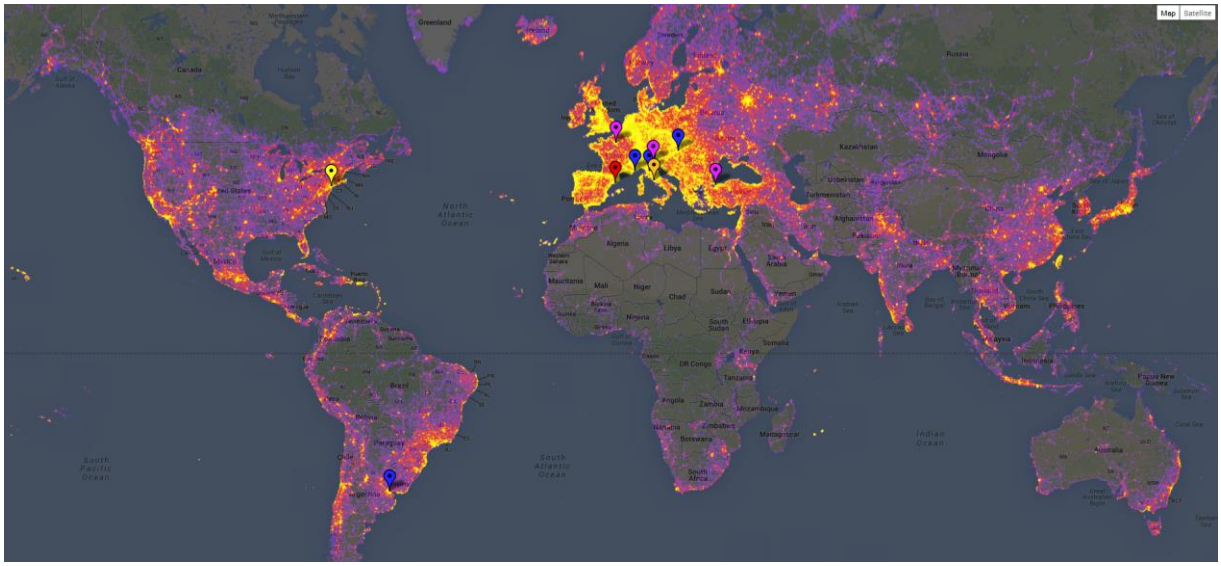

Figure 6-12: Example of plotting on a map

6.4. Visualisation Tools

Within the context of AEGIS, the consortium, based upon the core and the demonstrator requirements analysed, and based upon the conceptual architecture designed, is contemplating upon the integration and utilisation of one of the following platforms / services / libraries, each demonstrating pros and cons.

6.4.1. *Kibana* [7, 8]

Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data. The combination of Elasticsearch, Logstash, and Kibana (also known as ELK stack or Elastic stack) is available as products or service. Logstash provides an input stream to Elastic for storage and search, and Kibana accesses the data for visualizations such as dashboards. Kibana core ships with the classics: histograms, line graphs, pie charts, sunbursts, and more, which leverage the full aggregation capabilities of Elasticsearch.

6.4.2. *Banana*[9, 10]

Banana is a data visualization tool that uses Solr for data analysis and display. Data display in Banana is based on dashboards, which contain rows of panels that implement the analysis required. Banana is open source, and based on a port of Kibana. The Banana dashboard is the central feature of Banana, and is the place where the various visualizations are stored. A dashboard contains one or more controls for search query inputs and one or more displays over the results for that query. These controls and displays are called panels. Dashboards run as a client-side application in a web browser. Solr facets provide the quantifications required for visualizations, which can be charts, graphs, tables, and maps (for geospatial data). Dashboards also have tabular displays for drilling down to the individual documents in a results set.

6.4.3. *Grafana*[11, 12]

Grafana is an open source metric analytics & visualization suite. It is most commonly used for visualizing time series data for infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control. Grafana

[7] https://www.elastic.co/products/kibana

[8] https://github.com/elastic/kibana

[9] https://doc.lucidworks.com/lucidworks-hdpsearch/2.5/Guide-Banana.html

[10] https://github.com/lucidworks/banana

[11] https://grafana.com/grafana

[12] https://github.com/grafana/grafana

has a plethora of visualization options, from heatmaps to histograms, from graphs to geomaps, to help the user understand his data. Grafana allows the seamless definition of alerts where it makes sense, and allows notifications via Slack, PagerDuty and other tools and services, while it also supports dozens of databases, natively and empowers their mix in the same Dashboard.

6.4.4. *HighCharts*[13, 14]

Highcharts is a charting library written in pure JavaScript, offering an easy way of adding interactive charts to a web site or web application. Highcharts currently supports line, spline, area, areaspline, column, bar, pie, scatter, angular gauges, arearange, areasplinerange, columnrange, bubble, box plot, error bars, funnel, waterfall and polar chart types. Highcharts is solely based on native browser technologies and doesn't require client side plugins like Flash or Java. Setting the Highcharts configuration options requires no special programming skills. The options are given in a JavaScript object notation structure, which is basically a set of keys and values connected by colons, separated by commas and grouped by curly brackets. Through a full API the user can add, remove and modify series and points or modify axes at any time after chart creation. Numerous events supply hooks for programming against the chart. This opens for solutions like live charts constantly updating with values from the server, user supplied data and more.

6.4.5. *D3.js*[15, 16]

D3.js is a JavaScript library for manipulating documents based on data. D3 helps the user bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives the user the full capabilities of modern browsers without tying to a proprietary framework, combining powerful visualization components and a data-driven approach to Document Object Model (DOM) manipulation. D3 allows the user to bind arbitrary data to a DOM, and then apply data-driven transformations to the document. For example, D3 can be used to generate an HTML table from an array of numbers, or to create an interactive SVG bar chart with smooth transitions and interaction using the same numbers. D3 allows efficient manipulation of documents based on data, thus avoiding proprietary representation and affording extraordinary flexibility, exposing the full capabilities of web standards. With minimal overhead, D3 is extremely fast, supporting large datasets and dynamic behaviors for interaction and animation. D3's functional style allows code reuse through a diverse collection of official and community-developed modules.

6.5. AEGIS Specific Requirements

Within the context of AEGIS, the main architectural decision that will drive the selection of the most appropriate tool / service / library for integration, is associated with how the queries will

[13] https://www.highcharts.com/products/highcharts

[14] https://github.com/highcharts/highcharts

[15] https://d3js.org/

[16] https://github.com/d3/d3/wiki

be performed on the data. More specifically, the architectural decision to be taken within the context of the project is associated with whether the AEGIS consortium will mainly support:

1) indexed (Elastic Search or Solr based) queries, in which case the visualization efforts will focus on a dashboard as a service approach such as Kibana, Banana or Grafana, with Grafana probably being easier to integrate because it does not depend on an indexing engine, unlike Kibana and Banana which support automation of widget generation but are tightly connected to Elastic Search and Solr respectively.

2) "raw" queries in the AEGIS big data repository (and / or triplestore) in which case we would focus on solutions such as HighCharts or D3.js which are not dashboards provided as services, yet are completely index agnostic.

6.6. AEGIS Requirements met

The current subsection performs a mapping between the requirements of the visualisation component, and the core and demonstrator, functional, non-functional and technical requirements as identified and documented in deliverable D3.1. The first row of the table represents the unique code of the visualisation component requirement, the middle row provides the description of the requirements formulated based upon the description of the requirements documented in D3.1, updated based upon the goals and needs of the component as analysed in the current section, while the third and last row maps the specific requirement to the set of requirements from D3.1.

ID	Visualisation Requirements	Previous Requirement of Reference
VC_R1	Feature a user-friendly interface which provides an overview of supported kind of visualizations	NFR7
VC_R2	Act as a unified front end to multiple databases and (big) data types. The visualisation component should be able to handle (visualise / create advanced graphs) large datasets, queries spanning multiple datasets, and scale horizontally.	TR14 (NFR5), TR41 (CFR9, NFR1), TR55 (CFR5, CFR9, CFR18)
VC_R3	Preview a small selection of the results of the generated query so as to extract some initial insights out of the foreseen analytics	TR53 (CFR18, NFR3)
VC_R4	Update the visualisation by re-running the query contained in it.	FR_RT12
VC_R5	Automatically update the visualisation if a data source is modified, without the need to re-run the query	FR_RT14

VC_R6	Render data in as many ways as possible / support different kinds of visualisations, based on different types of input datasets formats. Provide means for visualizing different data modalities (e. g. special, temporal, statistical) and provide an overview of the supported kinds of visualization	FR_RT1, TR54 (CFR5, CFR9, CFR18)
VC_R7	Save the results of the analysis and visualisations	FR_RT2
VC_R8	Export the results of the analysis and visualisations, so that they can be in turn consumed by external applications and/or entities	FR_RT3

7. USAGE ANALYTICS

In this section, we define the main goal to be reached through the usage analytics tool adopted by AEGIS, giving a detailed description of the requirements that such tool has to support (paragraph 8.1). The main part of these requirements was elucidated from the technical, demonstrator and non-functional requirements collected during the works for WP3 and reported in D3.1. The Usage Analytic Requirements, hereinafter UAR, were identified taking also into account the results of WP1, in particular the results of the stakeholders' questionnaire (D1.1) and the scenarios, hence the needs and the expectations, pointed out by the demonstrators in WP1 (D1.2).

The overall objective of this review is to identify the UAR of the AEGIS Usage Analytic tool in order to adopt the best-suited solution exploiting its functionalities in order to raise the AEGIS stakeholders, to highlight the weak features of the AEGIS solution and to create a platform which fits as much as possible to the market needs.

Within this assessment, the purpose is to collect the UAR, to give an overview of the existent Usage Analytics tools, evidencing the pros and the cons of each (paragraph 8.2) and to make some considerations concerning their features and the goals to be reached through the Usage Analytics in AEGIS (paragraph 8.3).

8.1 USAGE ANALYTICS

To define the UAR which will drive the choice of the AEGIS Usage Analytic tool, the first step involved the study of the goals that the AEGIS developers and the whole consortium want to achieve through the Usage Analytics.
The aforementioned goals identified, considered as high-level Usage Analytic requirements, are the following:

1) Enhance the number of the AEGIS stakeholders: through the Usage Analytics tool we would like to understand the real AEGIS stakeholders' target. As example, the sector of the interested companies could be useful from one side to consolidate the approval of existent users, on the other side to develop and implement a strategy to improve the catchment area.
2) Extend the AEGIS market: through the Usage Analytics tool we would like to comprehend the country of provenance of the AEGIS users, in order to maximize the investments in advertising and evaluate the possibility to translate the AEGIS interface in further languages.
3) Enrich the AEGIS core functionalities: through the Usage Analytics tool we would like to rate the use of each AEGIS functionality, to understand which are the most appreciated by the users, trying to find a correlation between some major features (e.g. sector of the company activity, role of the user, type of data analysed, number of datasets and/or data sources involved in the analysis, type of analysis performed).
4) Enlarge the AEGIS data sources: through the Usage Analytics tool we would like to have information about the number and the type of the datasets involved in an analysis correlate to the type of analysis performed, in order to understand if the users are interested in specific types of datasets. Moreover, to guarantee a proper use of the

AEGIS platform we would like to record for each user the quantity of the data shared with the other users and the data from other source (non in-house data) involved in his queries/analysis. Could be helpful also track the amount of data stored within AEGIS, both temporarily and permanently, against time.

5) Evaluate the AEGIS user experience: through the Usage Analytics tool we would like to track the funnel of the AEGIS users considering both of the step before entering in the AEGIS platform hence the words used for the search on the browser and the type of the browser itself (i.e. Google Chrome, Internet Explorer, Safari...) and the path followed within AEGIS. From this analysis we could evaluate the user needs (e.g. curiosity, general information, direct access to a - defined type of - service...) and if and which are the pages with the major reduction of view compared to the previous one. From such user experience analysis we could increase our visibility in selected browser, and create direct links to functionalities of interest.

The UAR were driven from the list of requirements defined in the WP3 previous deliverable (D3.1), in agreement with the results of the D1.1 survey and the D1.2 scenarios and with the mission to finalize the goals and expectations both of the users, the stakeholders and the AEGIS team.
The following Table 8.1-1 lists the UAR that will pave the way for the choice of the Usage Analytic tool to be utilized by AEGIS. The first column of the Table reports an arbitrary incremental ID, the second column the usage analytics requirement description, while the third shows the previous requirement reference of D3.1 (TR for Technical Requirement, DFR for Demonstrator Functional Requirement and NFR for Non-functional Requirement). Please note that the UAR without references were pointed out from the questionnaire and from the scenarios.

Table 7-1: List of Usage Analytics Requirements, ID, Description and Previous Requirement Reference

ID	UA Requirement	Previous Requirement of Reference
UAR1	The UA could evaluate for a specific stakeholder/type of company which is the most utilized tool/service both analytics and concerning the visualization modality	TR44, TR46, TR47, TR48, TR54, TR55, TR66
UAR2	The UA could evaluate which are the most utilized data source (e.g. wearables, sensors...)	TR1, TR3
UAR3	The UA could compare the real-time data usage with the 'historical' data usage	TR1, TR2, TR4, TR50
UAR4	The UA could evaluate the percentage of analysis of data imported directly from the customer of the AEGIS user (e.g. from wearables/social network…)	TR1, TR2, TR3, TR4, TR16, TR21
UAR5	The UA could correlate the number and the type of datasets involved with the type of analysis and the stakeholder	TR3, TR4, TR14, TR35, TR40
UAR6	The UA could evaluate the utilization of in-house data compared to open data and data bought through the Business Brokerage Framework	TR16, TR21, TR22
UAR7	The UA could compare the proactive and passive behaviour (dataset upload versus dataset analysis utilization) of the AEGIS user	TR38, TR58
UAR8	The UA could evaluate the most utilized type of data (e.g. geospatial data, environmental data, weather data, Public Health Information Data…)	TR1, TR2, TR3, TR4, TR40
UAR9	The UA could evaluate how many stakeholders after reading the policies decide to use or withdraw the platform	TR25
UAR10	The UA could evaluate how many imported data are anonymized by AEGIS	TR29, TR59
UAR11	The UA could evaluate for each stakeholder the ratio between the use of his own uploaded code for the analysis compared to the AEGIS algorithms	TR69
UAR12	The UA could correlate the users' role with the type of AEGIS service utilized	TR17, TR19, TR23, TR24, TR26, TR58
UAR13	The UA could evaluate how many of the AEGIS users use the private cloud area for dataset storage provided by AEGIS	TR11, TR13, TR18, TR20, TR57
UAR14	The UA could evaluate AEGIS query storage utilization rate and the time of response of these queries	TR49, TR53
UAR15	The UA could evaluate how many users (and in which field) exploit the AEGIS notification functionality	DFR5, DFR30

UAR16	The UA could verify the performance of the AEGIS Analytic tool from the time of processing and/or visualizing point of view	TR49,TR68
UAR17	The UA could outline the interaction between the AEGIS platform and other applications	TR15, TR56
UAR18	The UA could outline how many users download/read the user guides	NFR7
UAR19	The UA could outline the setting of each language for the user interface, and track the country of the user to evaluate possible future other languages	NFR8
UAR20	The UA could monitor the access of the AEGIS platform (discerning from the tool) along the day in order to program efficient upgrades	TR15, TR67
UAR21	The UA could correlate the AEGIS failures with the type of analysis, the type of dataset and of the stakeholder	TR36, TR65
UAR22	The UA could evaluate which is the most utilized format of the data imported	TR7, TR8, TR32, TR33, TR42, TR48
UAR23	The UA could evaluate the customer satisfaction through user behaviour and/or specific feedback	TR64
UAR24	The UA could outline the usage of add/import, select and upload functionalities	TR37, TR51
UAR25	The UA could count the access to the AEGIS platform related with the country, the place and the link of provenience (e.g. a certain Google/browser search) in order to evaluate where and how is better to promote better AEGIS	
UAR26	The UA could track the funnel in order to understand the user needs and curiosity	
UAR27	The UA could monitor the natural languages of imported data in order to evaluate the usage of the AEGIS translating tool(s)	
UAR28	The UA could evaluate the number of access to the AEGIS platform and the number of analysis performed through AEGIS during the time	
UAR29	The UA could evaluate the percentages of processed data and of stored data during the time	

8.2 ALGORITHMS FOR USAGE ANALYTICS

In this section, we will present an overview of the available web analytics tools, describing each of them and comparing their performances pointing out their behaviour related to the features that we consider more relevant for their application in the AEGIS platform.

Nowadays a point of interest of each web service owner is the understanding of the performance of his website, the happiness of his customers and gain key context from competitors. Therefore, a good statistics package working in the background is essential to gather that important feedbacks. These statistics packages are properly named as Web Analytics. Web Analytics is the collection, measurement, analysis and documenting of internet data for the purpose of optimizing and understanding usage of web. Web analytics do not only measure the website traffic but may also be used for market and business research providing data about page views, visitors, and more.

The Web Analytics can be distinguished in two categories:

- Off-site Web Analytics: They refer to measurement and analysis of websites; they include the opportunity or the measurement of the potential audience of the website, visibility or the share of voice, and comments regarding what is happening on the Internet.

- On-site Web Analytics: They measure the journey of the visitors on a website. They consist of conversions and drivers (e.g. which landing pages are encouraging the people to make purchase). The data are compared with key performance indicators for the performance and is used to market the audience's response to the campaign and to improve a website.

There are two technological approaches in collecting data: the *logfile analysis* and *page tagging*. In the first case, the logfiles are read wherein the web server records its transactions. In page tagging, a Java Script runs on every page so that a third-party server is notified when a page is executed by a web browser. Both data collection can be processed producing reports for web traffic.

We have selected six of the most recommended by the experts Usage Analytics tools. These tools are listed hereinafter.

1. **Google Analytics** (google.com/analytics) – Free version and payment version

   Google Analytics is the simplest, most robust and most popular web analytics offering; it is a completely free service that generates detailed statistics about the web service visitors. It is used by over 50% of the top 10,000 websites in the world, according to the site's usage statistics; it is possible to find out as example where its visitors are coming from, what they are doing while on the site and how often they come back. It has a highly usability, a well-executed user interface and for an expert in usage analytics it's easy to receive more detailed reports. Moreover, it is possible to integrate Google Analytics with other packages for specific needs (e.g. AdWords for ecommerce…).

2. **AWStats** (https://awstats.sourceforge.io/) - Free

AWStats is an open source Web Analytics reporting tool, suitable for analyze data from Internet services such as web, streaming media, mail, and FTP servers; it parses and analyses server log files, producing HTML reports. AWStats supports most major web server log file formats. It goes deeper into the referring sites' information than most analytics packages. Data is visually presented within reports by tables and bar graphs. Static reports can be created through a command line interface, and on-demand reporting is supported through a Web browser CGI program. AWStats can be deployed on almost any operating system. AWStats can be installed on a workstation, such as Microsoft Windows, for local use in situations where log files can be downloaded from a remote server. Proper web log analysis tool configuration and report interpretation requires a bit of technical and business knowledge. AWStats support resources include documentation and user community forums.

3. **W3Perl** (http://www.w3perl.com/) - Free

W3Perl differs from other analytics packages in that it doesn't just measure web traffic, but also can parse the log files of email and RSS to measure just about anything choose, supporting the major web logfile formats. It is possible to set up the administration interface for web access and gain both real-time and session stats from there. The administration interface (that should be restricted with login/password) allows building configuration files, giving the opportunity to manage configuration files, package updates, run scripts, and see stats output with graphics and a sortable table. It is written in Perl and supported by any operating system. Finally different plug-in are available to enhance some functionalities such as the IP mapping with country code without querying DNS, HTML, PDF and email reports, get cities stats.

4. **Web Analyzer or Webalizer** (http://www.webalizer.org/) - Free

The Webalizer is a web log analysis software, which generates web pages of analysis, from access and usage logs. It is one of the most commonly used web server administration tools. Website traffic analysis is produced by grouping and aggregating various data items captured by the web server in the form of log files while the website visitor is browsing the website. The Webalizer analyzes log files, extracting such items as client's IP addresses, URL paths, processing times, user agents, referrers, etc. and grouping them in order to produce HTML reports. By default, The Webalizer produces two kinds of reports - a yearly summary report and a detailed monthly report, one for each analyzed month.

The yearly summary report provides such information as the number of hits, file and page requests, hosts and visits, as well as daily averages of these counters for each month. The report is accompanied by a yearly summary graph.

5. **ClickTale** (https://www.clicktale.com) - Free to $990 (3 months free on paid plans)

A qualitative customer analysis, ClickTale records every action of the website's visitors from their first click to the last. It uses Meta statistics, creates visual heat maps and reports on customer behaviour, as well as providing traditional conversion analytics. ClickTale is essentially a video recorder for web site visits and provides detail about mouse movement, scrolling, and dozens of other critical visitor behaviours.

6. **Optimizely** (https://www.optimizely.com) - $19-$399/month

   Optimizely is simple to use but its results can be quite powerful. In essence, it's an easy way to measure and improve the website through A/B testing. As a business, it is possible to create experiments with the site's very easy-to-use visual interface. The advantage of this service is that people with zero coding or programming background can use it also.

8.3 THE AEGIS USAGE ANALYTICS TOOL

In the table below (Table 7-1) we show a resume of the main features of four of the tools aforementioned: Google Analytics, AWStats, Webalizer and W3Perl. These tools were considered the most suitable for the AEGIS needs among the six described before. In fact, taking into account that both of ClickTale and Optimizely do not have any particular feature despite of the others and are paid tools we decided to focus, at least for the first project phase, on the others.

Table 7-2: Usage Analytics Tools and their main features

Usage Analytics Tool	Characteristics
Google Analytics	Free and payment version.
	Google Analytics 360, the payment version, allows the collection of higher data amounts, giving more flexibility regarding data management, offering more settings features, and visualization tools. Moreover, it has roll-up functions and customizable reports. GA 360 offers also the support of a data analyst.
	Google Analytics for Mobile Apps allows the measurement and optimize user acquisition and engagement with mobile apps.
	Technological approach: page tagging
	It is possible to monitor the visitors of the website from browsers, referrer sites or advertisements, pay per click networks or email marketing, links from .pdf files
	Integrated with AdWords, users can now review online campaigns by tracking landing page quality and conversions (goals). Goals might include sales, lead generation, viewing a specific page, or downloading a particular file. Through GA the performance of the advertising could be determined, providing information to optimize the ratio between the investments and the income.
	GA provides various types of dashboards, both for expert and non-expert users
	Main Analysis provided:
	- the most visualised page
	- referrer
	- how long the user has stayed in the site
	- geolocalization of the user
	It is possible to track data for free for websites with less than 10 million of visualization/month
AWStats	Open source, suitable for analysing data from web, streaming media, mail, and FTP servers.
	Technological approach: server log files producing HTML reports. Works both with IIS (W3C) and with personalized log format.

	Data presented within reports by tables and bar graphs. Static reports can be created through a command line interface
	Monthly and yearly statistics
	Various types of reports, including also personalized reports for miscellaneous/ marketing purpose.
	Written in Perl
Web Analyzer	Free
	Technological approach: log files, usage reports in HTML format.
	Written in C to be extremely fast and highly portable.
	Needs a patch to work with IIS (W3C) and does not work with personalized log format.
	Monthly and yearly statistics
	Generated reports can be configured from the command line, or more commonly, by the use of one or more configuration files.
	Unlimited log file sizes and partial logs are supported, allowing logs to be rotated as often as needed, and eliminating the need to keep huge monthly files on the system.
W3Perl	Free
	Technological approach: log files, Most major web log file formats are supported. *Page tagging* and counter are also supported if the user do not have log files access. Usage reports in HTML format.
	Generated reports can be configured from a single command line or from a web browser.
	The reports are spread over HTML pages, with graphics and a sortable table.
	Features like hosts, pages, scripts, countries, file type, traffic, referrer, user agent, and error are available along with other specific W3Perl stats like real-time and session stats.
	W3Perl has an administration interface, which allows building configuration files from a web interface. One can also manage configuration files, package updates, run scripts, and see stats output
	Written in Perl.

Considering the usage analytics requirements of Table 7-1, our choice fell on Google Analytics. Google Analytics seems to be the most suitable tool for the AEGIS platform needs; in fact, first, it guarantees robustness and worldwide availability and reliability. Moreover, GA offers user – friendly interfaces, appropriate for both expert and non-expert users, provides the main required functionalities (see Table 7-1) such as the geolocalization of the user and the tracking of the funnel. Last but not least GA is a free tool, and the free package allows to satisfy the main part of the AEGIS usage analytics requirements, nevertheless for further needs such as a potential increase of the data to be analysed, GA could be improved through the higher functionalities of the payment version.

8. CONCLUSION

The document presents what the AEGIS consortium plans regarding data and metadata harvesting, harmonisation, visualisation, knowledge extraction, business and usage analytics. The selection of approaches, techniques and tools presented in this document is not final. It will

further evolve in this and in other work packages. This document is important as an input for the WP3 for defining detailed technical design of the AEGIS platform.