Mining BIM Models: Data Representation and Clustering from Implicit Relationships

Joyce P. M. Delatorre[1], Fabiano R. Correa[2], Eduardo T. Santos[3]

1) M.Sc. Candidate, Department of Construction Engineering, University of Sao Paulo, Sao Paulo, Brazil. Email: joyce.delatorre@usp.br
2) Ph.D., Asst. Prof., Department of Construction Engineering, University of Sao Paulo, Sao Paulo, Brazil. Email: fabiano.correa@usp.br
3) Ph.D., Asst. Prof., Department of Construction Engineering, University of Sao Paulo, Sao Paulo, Brazil. Email: etoledo@usp.br

Abstract:

Building Information Models are sometimes used as databases, but typically, only component properties and parametric data are retrieved and used in business processes. However, BIM models also contain many implicit data, both in the geometry of and in the relations between their components.

Exploitation of this implicit data linked with explicit information may provide useful, new and untapped knowledge hidden in BIM models. Our hypothesis is that BIM models could be used in data mining analytics to add value to the enterprise.

The main difficult to achieve our intent is to transpose object-oriented BIM models to a flat representation so that it could be used with the large majority of data mining algorithms available. In this article, it is discussed how to deal with: different entities that have different types and number of attributes; the geometry of each instance of an entity that is an important source of information and must be translated somehow; the hypothesis, often made by statistical models, that data is independently distributed, against the fact that entities in BIM models contain many types of relationships, including topological ones such as “inside”, “above”, or “touching” consisting in very important information.

Our initial approach it to test if a clustering algorithm can find patterns in BIM models related to spatial relationships not explicitly modeled. In the experiment, a BIM model, inputted with construction quality-control data that could lead to the cause of the flawed production of some components, is used. Latent Class Models were applied for clustering in an implementation using the R statistical software package and its *poLCA* extension, and *IfcOpenShell* to parse the models in the IFC data format.

Preliminary results indicate that it is possible to aggregate components in a cluster based on its spatial distribution.

Keywords: BIM, Data Mining, Clustering.

1. INTRODUCTION

Building Information Modeling (BIM) has gained strength in the construction market. For Nepal et al. (2012), even though the fast development of BIM has opened numerous opportunities for design and construction, and although there is a growing number of developers using BIM to support construction management, there are still numerous challenges related to extracting information from building information models, which limits their usability for construction and subsequent processes.

One difficulty is that the storage and use of data from BIM models are not always trivial tasks. For Wülfing, Windisch and Scherer (2014), the search or retrieval of necessary information in BIM models is complex, as all information will not be necessarily integrated into a single model due to the difference between stakeholders interests, and may be linked to other information BIM resources, such as cost or time models. Mazairac and Beetz (2013) believe that the large amount of information generated by the integration of models from different disciplines in a common virtual model also increase the size and complexity of data repositories.

BIM objects have the ability to bind, receive, impart or export sets of attributes, such as materials, acoustic and energy data, among others, for other applications or models (Eastman et al., 2011). It is possible to store construction information generated throughout the building's life cycle in BIM models, avoiding the use of different media formats and providing a unified comprehensive information resource (Wülfing, Windisch and Scherer, 2014). Currently, BIM models were almost exclusively used for inputting information and exporting of its attributes.

Beyond construction information, BIM works with parametric objects in which the geometry is integrated in a non-redundant way and does not allow inconsistencies. By using parametric rules, for example, the relationship

between elements is explicit: the objects automatically change their associated geometries when they are inserted in a building model or when changes are made to objects associated with them. Parametric BIM objects have geometry, a location in a tridimensional space, topological relationships and carry a variety of properties that can be interpreted, analyzed and acquired by other applications (Eastman et al., 2011).

BIM models contain a lot of implicit data, both in the geometry of their components and in the relations between them. Exploitation of this implicit data linked with explicit information may provide useful, new and untapped knowledge so far hidden in BIM models.

2. BIM AS DATA SOURCE AND INFORMATION REPOSITORY

Advances in technology have allowed companies to generate and store large volumes of data. According to Silberschatz, Korth, Sudarshan (2006), the data used by organizations for decision-making can be originated from different sources and can be stored under different schemes. Often, for reasons of performance and control of the organization, the data sources are not accessible to any part of the company by simple request.

Given the possibility of entering information on BIM models, it is possible to concentrate a large volume of relevant data in the BIM model, which would make it a data repository of design, construction and operation information.

In order to explore the implicit data to extract knowledge hidden in BIM models, it is necessary to go beyond database manipulation. The general process of extracting patterns and discover knowledge from large amounts of data is known as Data Mining (Han et al., 2011).

Knowledge Discovery in Databases (KDD) techniques, including Data Mining, can be used for analysis and processing of large volumes of information. According to Silberschatz, Korth and Sudarshan (2006), KDD techniques aim to automatically discover rules and standard statistics from the data. Data mining combines Knowledge Discovery techniques from artificial intelligence and statistics analysis with techniques for efficient implementation, allowing their use in a database with a large volume of information.

A database comprised of several BIM models, or one BIM model by itself, can be used as a source of information for applying data mining techniques to identify useful patterns in data that could inform companies, such as: the identification of patterns in design and / or execution flaws, causes for variations in building task productivity, etc.

Some other studies have addressed BIM as a source of input data for KDD processes, such as the research of Jiang Zhang and Zhang (2013) that proposed a method for integrating text information and BIM models. Other works aim to provide useful techniques to query BIM models with functionalities beyond a simple database query (Mazairac & Beetz, 2013), including the exploitation of topological relationships in the data (Daum & Borrmann, 2014). However, little has been investigated about the use of specifics of BIM information for data mining, among them, the geometry of the elements.

The hypothesis advocated in this article is that BIM models could be used in data mining analytics to add value to the enterprise. In this article, it is discussed the difficulties in transposing object-oriented BIM models to a flat representation so that it could be used with data mining algorithms: different entities have different types and number of attributes; the geometry of each instance of an entity is an important source of information and also must be translated somehow; statistical models often hypothesize that data is independently distributed, but entities in BIM models contain many types of relationships, including topological ones such as “inside”, “above”, or “touching”.

Our initial approach was to test if data mining techniques could be used to find patterns related with the implicit information in the models, without resorting to a data representation that makes explicit the spatial relationships between BIM components.

3. MINING BIM MODELS

In the scientific literature, there is a wide range of mathematical models to represent and deal with real data. Statistical and Probabilistic models currently are more popular and successful when applicable to real data consisting in huge data sets and uncertain information. Logic models are also used, but results are still inferior if compared with the former discussed models (Russel & Norvig, 2009; Murphy, 2012). Although logic models are more adequate to the inherent structure of real data, statistical (no structure at all) and probabilistic (some structure) methods dominate the landscape.

Besides the discussion of data representation (data rich / relational data or flat representation) and the architecture of the model (statistical, probabilistic or logical models), there is also to be decided which task would be performed in the dataset and if the user will provide some labeled dataset to train the algorithm

(supervised learning) or if it will explore and discover some kind of knowledge or patterns by itself (unsupervised learning). Common tasks are: classification and class probability estimation, regression, similarity matching, clustering, co-occurrence grouping, profiling, link prediction, data reduction, and causal modeling (Provost & Fawcett, 2013).

For the initial purpose of exploring BIM models with Data Mining techniques, it seems that clustering, where the algorithm identifies and groups similar components in a given number of clusters, would be more appropriated to our intent. Clustering is a kind of unsupervised learning.

3.1 Clustering within BIM models

BIM models are composed of entities that, in general, have very clear semantic regarding construction information. Basically, when it is necessary to deal with the model, it is used some sort of database search and filtering.

The main difficulty in using BIM models as input to Data Mining techniques is exactly the data richness of this representation. Those algorithms usually expect data in the form of tables in a database, where each line is characterized by a fixed number of columns, with a defined type in each column.

If that was the case, the use of BIM models for data mining would be trivial. However, BIM models, as many other types of real data, have a complex pattern of relationships between its components and, in general, have to be flattened to allow traditional algorithms to be used.

But the choice on how to flat a model is not without consequences and there are many ways to do it, which impact in the final result from such algorithms. For the discussion presented below, BIM models are considered to be in the Industry Foundation Classes (IFC) format. IFC models are largely adopted as it is the open standard used throughout the industry (buildingSMART, 2013).

As it is not trivial to work with components that are of different types and thus have distinct number of attributes, the solution proposed is to take advantage of the hierarchical nature of inheritance presented in BIM models (considering it in its IFC representation schema), and deal only with entities of a specific abstract supertype. All its subtypes would have the same subset of attributes.

To produce a flat representation of the geometry of each component in the model, the alternatives depend on the architecture of the chosen model. It could be a bounding box specified by two diagonally opposed points (six attributes – coordinates - in a tridimensional space) if the data mining algorithm accommodates continuous variables. Or, as is the case with many BIM models, the geometries that appears in the model consists of a small set of types with many repetitions, and thus could be represented as a discrete or categorical variable.

The very process of flattening a dataset with relationships between its components consists in transforming all components in individuals, independently distributed. So relationships, including topological ones such as “inside”, “above”, or “touching”, would be neglected and would remain implicit in the observed attributes of the model. The spatial relationships could be presented in a flat form as categorical values of grid line position and elevation levels for BIM models, or using continuous variables in a similar approach taken for the geometry of the components.

The next decision to make, regards the architecture of the model, which will determine the allowed types of variables.

3.2 Latent Structure Analysis

For clustering, it was chosen the Latent Class Analysis that works primordially with categorical values. The classical theory of latent structure analysis was founded by Lazarsfeld in 1950, in the context of studies of sociological phenomena (Andersen, 1982). Later on, more efficient estimation methods were introduced, following the work of Anderson (1954). It is particularly suited for sociological sciences as it considers the “latent structure” on its name as discrete variables that cannot be observed, but that appears in the observation of other correlated variables. It resembles factor analysis (Harman, 1976) and other statistical methods which work with continuous variables.

The idea behind it is that latent classes (clusters) exist which do not appear in the observed model with manifest variables (the attributes of the components in the BIM model). Each cluster has its own probability distribution over the values of the manifested variables. In principle, it is like trying to approximate a distribution of probability with a finite mixture model. The number of mixtures, for example Gaussians, which compose the model, is the number of clusters that will be used.

The probability that an individual *i* in class (or cluster) *r* produces a particular set of *J* outcomes on manifest variables, assuming local independence, is the product represented in Eq. (1) (Linzer & Lewis, 2013):

$$f(Y_i;\pi_r) = \prod_{j=1}^{J}\prod_{k=1}^{K_j}(\pi_{jrk})^{Y_{ijk}} \quad (1)$$

where π_{jrk}: class-conditional probability that an observation in class *r* produces the *k* th value on the *j* th variable,

J: number of manifest variables;

K_j: number of possible outcomes for each manifest variable *J;*

Y_{ijk}: observed value of the J manifest variables such that it equals to 1 if individual i presents k value of the j manifest variable and 0 otherwise.

The probability density function across all classes is the weighted sum presented in Eq. 2 (Linzer & Lewis, 2013):

$$P_r(Y_i|\pi,p) = \sum_{r=1}^{R} p_r\, f(Y_i;\pi_r) \quad (2)$$

where p_r: the *R* mixing proportions that provide the weights in the weighted sum of the component tables, with $\sum_r p_r = 1$.

To estimate the parameters of the model, it is necessary to maximize the log-likelihood function given by Eq. 3:

$$\mathbf{ln}\, \boldsymbol{L} = \sum_{i=1}^{N} \ln P_r(Y_i|\pi,p) \quad (3)$$

where *N*: number of individuals in the dataset;

3.3 Using *poLCA* package and *R* software

R is both a language and a very popular free software environment for statistical computing and graphics (R Project, 2016), and currently is in its 3.2.3 version. It runs in a wide variety of platforms including Windows. Many packages were developed in this language, including *poLCA* (Linzer & Lewis, 2011) which, given a number of classes, performs the Latent Class Analysis. In this research, all the clustering was done using *poLCA* running in R software.

It uses expectation-maximization and Newton-Raphson algorithms to find maximum likelihood estimates of parameters of the latent class and latent class regression models (Linzer & Lewis, 2013). *poLCA* only accepts manifest variables with categorical values.

To avoid local maximum, it is necessary to run *poLCA* many times. With the calculated maximum likelihood, it is possible to keep the model that obtained the maximum likelihood.

When estimating the parameters of a model representing observed data, it is important to assess how well it fits the data and not the noise. The number of parameters estimated for the model is one indicative of fitness: estimation of too many parameters tends to fit to noise in the data, causing overfitting. Another metric calculated for this purpose by *poLCA* is the Bayesian Information Criteria (BIC) given by Eq. 4. Lower values of BIC indicate a better fitted model.

$$BIC = -2\Lambda + \Phi \ln N \quad (4)$$

where Λ: Maximum log-likelihood;

Φ: Total number of estimated parameters;

N: Number of individuals.

In the case of clustered data, the number of clusters to be employed in the estimative is unknown beforehand. One recommended procedure is to started running the algorithm with two latent classes, and then increment the number of clusters based on the result of BIC, maximum likelihood value, and number of parameters, among other metrics, that determines which model best fits the data.

4. METHODOLOGY

To run the proposed data mining technique, it is necessary to have a BIM model created or exported in IFC format. The developed software parses this IFC file, extracts X, Y and Z coordinates and quality-control property of each building element present in the BIM model and translates it to an input file for the *poLCA* package running in R environment. Then, an LCA-based cluster analysis is performed, producing results to be examined in the form of clusters.

4.1 BIM Test Model

Initially, the system was tested with simple models composed of columns and slabs were an entire floor or just a single position in the structure grid possessed components marked as not approved by quality-control standards. Our system was able to successfully create a cluster containing those components. The more complex test which is presented in this article is described below.

The BIM model used in this research is the reinforced concrete structure of a 19-story office building with auditorium, restaurant and theater, totaling 5766 elements. The lack of architectural detailing does not affect the overall results for this initial work of exploration of the potential of data mining techniques in BIM models.

Only column (3041) components had an execution quality control parameter, inputted after site inspection, with the following possible values: non-conform or conform.

For testing our setting, two groups of elements were marked as non-conform: one composed of several columns near the west façade (Figure 1) and, the other, a set of columns and foundation blocks vertically aligned in the annex (Figure 2). Our goal is to detect if this two sets of non-conforming columns appear in different clusters in the final result, thus showing correlation between spatial distribution and quality-control status.

Figure 1. Cluster of non-conform components in the structure model

Figure 2. Line of non-conform columns above a non-conform foundation

4.1 Preparation of the IFC Model

To prepare the data for the Latent Class Analysis algorithm, it is necessary to pre-process the IFC model. The pre-processing was done with the help of *IfcOpenShell* library (IfCOpenShell, 2016) to parse the IFC file. A C++ code, integrated with the library, was created to transform IFC models into an input file for the *poLCA* package.

In the pre-processing, all *IfcBuildingElement*-type classes, such as *IfcColumn* and *IfcSlab*, were extracted and stored in memory. The chosen attributes for the representation of BIM models for data mining were mapped in previously established categorical values.

In preparing the model, the following information was used:

- The global coordinates (X, Y) were calculated by the IFC library from multiple local coordinates systems. Also, it was necessary to augment the information because mapped items with identical geometry could be translated or rotated in this mapping process and the used library do not contemplate this fact;
- the *IfcRelContainedInSpatialStructure* class, which gives each element a position in the Z coordinate, related to the different pavements;
- the Property Set (PSet) 'Identity Data', in which we registered quality-control status;

Table 1 lists all the data considered in representing each element of the model. Each component of the model was represented by a vector with four components: Grid Position X (24 possible values); Grid Position Y (24); Elevation (24); Quality-control status (2).

Table 1. Representation of elements from the BIM model used with data mining techniques

Manifest variables	Description	Categorical Values
Grid Position X (B variable in figure 3)	Grouping components based on its position on X coordinates of the reference plane	The numeric values are substituted by a discrete number of grid lines (range: 0 -23);
Grid Position Y (C variable in figure 3)	Grouping components based on its position on Y coordinates of the reference plane	The numeric values are substituted by a discrete number of grid lines (range 0 - 23);
Elevation Position (D variable in figure 3)	Grouping components based on its elevation	1…24 pavements (5 underground) (range 0 - 23)
Quality Control (E variable in figure 3)	Conformity status of the element	IfcPropertySingleValue (related to Comments property added to the model): conform (0) or non-conform (1) (range 0 -1)

The IFC model used in the experiments had a size of about 50 MB, and was parsed in about 15 minutes in a core i5 PC with 8 GB of RAM. The input file for *poLCA* package was produced in 3 to 4 minutes as the properties of elements in IFC models are dispersed throughout the IFC file, and specific properties were used.

5. RESULTS

5.1 Running *poLCA* for clustering

poLCA executes almost instantly in each run, but to avoid reaching a local maximum, it is necessary to configure it to run many times (set to 100 times) and to keep the model which reached the maximum log-likelihood (option *nrep=100* in *poLCA*).

The algorithm deals with individuals (or components in our case) presenting missing values in its manifest variables (as was the case of about half of the components in the test model, which didn't have a grid position value – in fact, only the columns had this information), ignoring them during the estimation procedure.

We tested the data with clustering for 2 to 5 classes. The meaning of each cluster is somewhat undefined. Table 2 lists the result of each run to determine the best architecture for our model. It turns out that as the number of classes increases, so does the number of parameters estimated (which could lead to overfitting of the model); the maximum log-likelihood increases and the BIC decreases, until it reaches a minimum, which seems a better fitted model. Using 5 classes, the results grow worse. Analyzing the distribution of components between the clusters, it seems that 4 clusters best accommodate our data.

Table 2. Results of *poLCA*

Number of clusters	Maximum log-likelihood	Number of estimated parameters	BIC
2	-23,388.54	139	47891.86
3	-22,825.10	209	47326.37
4	**-22,395.68**	**279**	**47028.93**
5	-22,222.56	349	47244.09

Our expectation was that the technique would be able to separate in two different clusters the two groups of non-conform components: the one vertically aligned and the other concentrated in one façade (see figures 1 and 2). Those components would be together with many other elements with the same spatial or topological context. This in fact happened in clusters 2 and 4.

Figure 3 represents the output of the *poLCA* algorithm in two different forms. In the left, it is shown the direct result of clustering from the software, and exhibits the probability distribution of each manifest variable (B, C, D, and E – see Table 1). One should pay particular attention to cluster (Class) 2 on Figure 3: it contains almost 20% of the components of the model (from the total of 3041 elements considered in the estimation procedure), well located in the lower pavements of the building (D variable), including foundation, and mainly located around the faulty components, vertically aligned (all of them are in this cluster). The second group of non-conform components is all inside cluster 4, together with more than 35% of the components of the building.

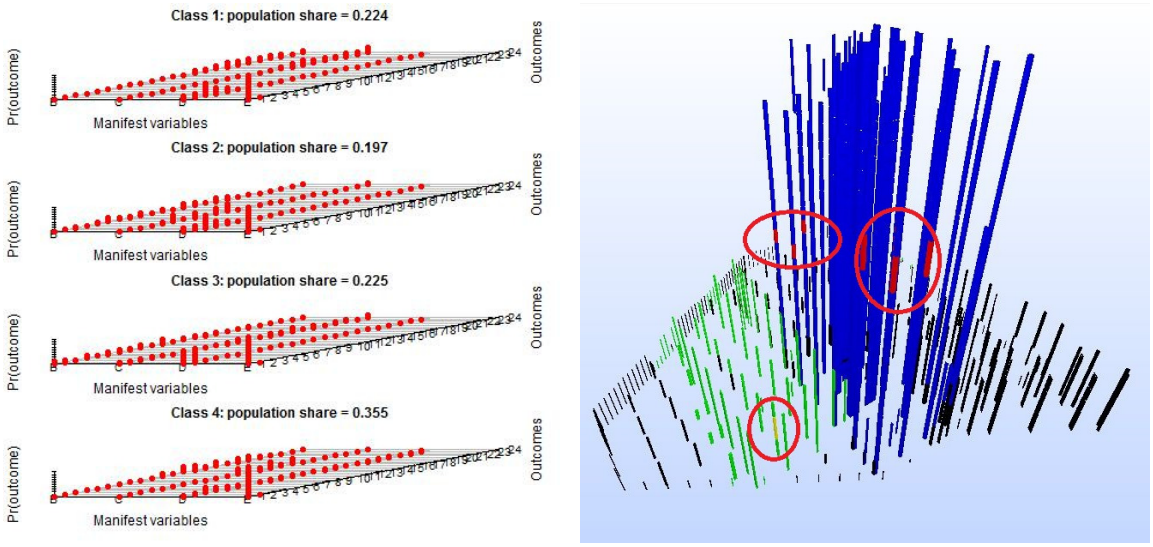

Figure 3. *poLCA* results for 4 latent classes (left) and visualization of clusters and non-conform elements (right)

The right portion of Figure 3 represents the elements of the building that were used in the data mining process, and they are colored based on the cluster in which they were assigned by the clustering process: clusters 1 and 3 (black: no non-conform elements), cluster 2 (green: conform and yellow: non-conform elements), and cluster 4 (blue: conform and red: non-conform elements). The non-conform components are highlighted with red ellipses.

6. DISCUSSION

The expected result of the tests we run on *poLCA* was the existence of one cluster concentrating the non-conform components vertically aligned with other components with the same spatial relationship, such as in the same façade, or aligned, or other form of spatial correlation between the manifest variables grid line / elevation with the quality control parameter. We achieved that result.

Although LCA, as the architecture of the model to be explored with Data Mining techniques, seems not to be specially suited for the task of exploring the implicit geometry and topological features present in BIM models, it is an inexpensive computational resource that could be used in preliminary analysis of patterns.

Other potential features to be explored in the same architecture of LCA would be to combine continuous and categorical variables in the model, but this is only possible with commercial software (e.g., Mplus from UCAL), not available for this research, at this time.

Clearly, a mathematical model which could more directly represent explicit topological relations between elements in the BIM model, such as undirected graphical models, are a possible path to follow in trying to deal with rich structured models. Also, it is important to state that many models that do not represent the dependencies among different components of the data may achieve superior results (e.g. Support Vector Machines), avoiding a more computational expensive model which represents a great deal more of the data set. That hypothesis should be tested.

7. CONCLUSIONS

It was demonstrated that without using explicit spatial or topological relationships between components, it is possible to explore BIM models with data mining techniques. However, the data representation, architecture and task defined must be used with other BIM models, and for other purposes related to the spatial positioning of its components, to entirely validate our approach. Still, there is a great amount of work to do in exploring BIM models with data mining and correlated software tools and a lack of studies in the subject.

ACKNOWLEDGMENTS

The last author would like to thank CNPq - National Council for Scientific and Technological Development for the partial support received.

REFERENCES

Andersen, E. (1982). Latent Structure Analysis: a survey, *Scandinavian Journal of Statistics*, 9 (1): 1-12.

Anderson, E. (1954). On estimation of parameters in latent structure analysis. *Psychometrika*, 19 (1), 1-10.

buildingSMART. (2013). *IFC4 Release Candidate 4*. Retrieved from buildingSMART website: http://www.buildingsmart-tech.org/ifc/IFC2x4/rc4/html/index.htm, accessed on September 24, 2015.

Daum, S. & Borrmann, A. (2014). Processing of topological BIM queries using boundary representation based methods. *Advanced Engineering Informatics*, 28 (4), 272-286.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors* (2nd ed.). John Wiley & Sons.

IfcOpenShell (2016). IfcOpenShell: the open source ifc toolkit and geometry engine. Retrieved from http://ifcopenshell.org/, accessed on January 9, 2016.

Han, J; Kamber, M.; Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.), Morgan Kaufmann.

Harman, H. (1976). *Modern Factor Analysis* (3rd edition), University of Chicago Press.

Jiang, S., Zhang, H., Zhang, J. (2013). Research on BIM-based Construction Domain Text Information Management, *Journal of Networks*, 8 (6), 1455–1465.

Linzer, D. and Lewis, J. (2011). poLCA: an R Package for Polytomous Variable Latent Class Analysis, *Journal of Statistical Software*, 42(10): 1-29.

Linzer, D. and Lewis, J. (2013). poLCA: Polytomous Variable Latent Class Analysis. R package version 1.4.

Mazairac, W. and Beetz, J. (2013). BIMQL – An open query language for building information models, *Advanced Engineering Informatics*, 27 (4), 444–456.

Murphy, K. (2012). *Machine Learning: a probabilistic perspective* (1st edition), The MIT Press.

Nepal, M. P. et al. (2012). Querying a building information model for construction-specific spatial information, *Advanced Engineering Informatics*, 26 (4), 904-923.

Provost, F & Fawcett, T. (2013). *Data Science for business: what you need to know about data mining and data-analytic thinking* (1st edition), O'Reilly Media.

R Project (2016). The R Project for Statistical Computing. Retrieved from The R Project website: https://www.r-project.org, accessed on January 9, 2016.

Russell, S. & Norvig, P. (2009). *Artificial Intelligence: a modern approach* (3rd edition), Pearson.

Silberschatz, A., Korth, H.F., Sudarshan, S. *Sistema de banco de dados* (5 nd ed.). Elsevier.

Wülfing, A., Windisch, R., Scherer, R. J. (2014). A visual BIM query language. *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014*, 157-164.