ISSN 2395-8618

Editorial

STARTING from this issue, the Editorial Board and the owner of the journal, the Centro de Innovación y Desarrollo Tecnológico en Cómputo of the Instituto Politécnico Nacional has accepted my suggestion to change the complete title of the journal to *Polytechnic Open Library International Bulletin of Information Technology and Science*. The new title, with clear meaning in English but with the familiar acronym POLIBITS, will help positioning of our journal as an international open access publication in computer science and computer engineering, will open it to wider readership, and will help attracting better authors.

This issue of POLIBITS includes ten papers by authors from seven different countries: Chile, USA, India, Mexico, Colombia, Cuba, and Russia. The papers included in this issue are devoted to such topics as logical algorithms, data visualization, sentiment analysis, word sense disambiguation, business process modeling, ontology learning, software engineering, computer vision, recommender systems, information extraction, and question answering.

Adrián Jaramillo et al. from **Chile** in their paper "Comparing the Black Hole and the Soccer League Competition Algorithms Solving the Set Covering Problem" compare implementations of the Black Hole and Soccer League Competition algorithms in a statistical way, involving the use of non-parametric tests and supported by R statistical computing environment, considering regularity and consistency of their results. Both implementations are tested on the same benchmark sets.

Chaman Lal Sabharwal from **USA** in his paper "Exploration, Exploitation Phenomena and Regression Analysis: Propensity Metric, Anomaly Reduction, Dimensionality Reduction" applies visualization for comparison of the Ordinary linear Least Square approximation model has for the best fit regression for linear trend data with other existing methods. He has found that this technique is reliable and preferable to explain to the expert as well as non-expert. The empirical tests show the accuracy improvements over conventional methods.

Nachiappan Chockalingam from **India** in the paper "Simple and Effective Feature Based Sentiment Analysis on Product Reviews using Domain Specific Sentiment Scores" outlines a novel technique to extract features from a product's reviews along with the corresponding sentiment expressed, using POS tagging and Dependency Parsing in conjunction. The use of these of these allows both the context and the parts of speech of a word to be employed in feature and corresponding opinion word detection. The opinion word is given a sentiment polarity determined from a training set of positive and negative reviews. The method described in this paper is for large data sets, and requires no domain specific data for feature extraction.

Grigori Sidorov and Francisco Viveros-Jiménez from **Mexico** in their paper "One Sense per Discourse Heuristic for Improving Precision of WSD Methods based on Lexical Intersections with the Context" show how to increase precision of word sense disambiguation for some word classes of these simple methods to the level comparable with that of the most sophisticated ones. They observe that these methods usually disambiguate correctly those words that conform to the One Sense per Discourse heuristic (OSD words). They use Semcor and Wikipedia to find the OSD words and left non-OSD words without disambiguation, thus improving precision at the expense of recall. Their motivation for this situation (more precision, less recall) is, as follows. First, if one needs high-quality disambiguation and uses human evaluators, then one can reduce the cost by asking them to disambiguate only words that are really difficult for the algorithms; (2) in an automatic system, one can apply this method for disambiguation of the corresponding words, and use other more sophisticated method for disambiguation of other words, i.e., use different methods for disambiguation (meta-disambiguation). The authors experimented with the complete and simplified Lesk algorithms, the graph based algorithm, and the first sense heuristic. The precision of all algorithms increased and some algorithms reach the level of the inter annotator agreement.

Hugo Ordoñez et al. from **Colombia** in their paper "TrazasBP: A Framework for Business Process Models Discovery Based on Execution Cases" present TrazasBP, a framework for Business Process (BP) indexing and searching based on execution cases. It indexes BPs based on execution cases (traces) retrieved from log files. TrazasBP not only takes into account the textual information of BP elements, but also the causal dependence between these elements. Furthermore, due to its low computational cost, TrazasBP can be used as indexing mechanism in order to reduce the search space. Experimental evaluation shows promising values of graded precision, recall and F-measure when compared with results obtained from human search.

V. Sree Harissh et al. from **India** in their paper "Unsupervised Domain Ontology Learning from Text" present an iterative focused web crawler for building corpus and an unsupervised framework for construction of Domain Ontology. The proposed framework consists of five phases: (1) Corpus Collection using Iterative Focused crawling with novel weighting measure; (2) Term Extraction using HITS algorithm;

ISSN 2395-8618

(3) Taxonomic Relation Extraction using Hearst and Morpho-Syntactic Patterns; (4) Non Taxonomic relation extraction using association rule mining; and (5) Domain Ontology Building. Evaluation results show that proposed crawler outweighs traditional crawling techniques, domain terms showed higher precision when compared to statistical techniques and learnt ontology has rich knowledge representation.

D. Larrosa et al. from **Cuba** in their paper "GeCaP: Unit Testing Case Generation from Java Source Code" propose a tool that allows developers to automatically generate test cases for unit testing from Java source code. In this proposal, the basis path testing technique is used for the design of the test cases. The control flow graph is automatically generated from the source code being tested, in order to subsequently generate the independent paths. Finally, the combinations of test values that satisfy every linearly independent paths are generated. In the process of implementing this new tool, a case study was designed for the purpose of validation; metaheuristic algorithms were applied to generate test values and value combinations for each path. These combinations were compared against the ones obtained by other state-of-the-art algorithms. Since in this case study a 100% coverage of the independent paths is reached, the proposed tool exhibits competitive results with respect to the ones reported by tools proposed by other authors.

Antonio Alarcón-Paredes et al. from **Mexico** in their paper "Naïve Screw Nut Classifier Based on Hu's Moment Invariants and Minimum Distance" present an algorithm for classification of screw nuts by means of digital image processing. This work is part of a project where a production line was built, and is focused on the quality assessment section. The algorithm presented classifies among good and poor quality screw nuts passing by a conveyor belt, by computing Hu's moment invariants of its picture. Those moment invariants are the input of a minimum distance classifier, obtaining very competitive results compared with some other classification algorithms of the WEKA platform.

Liliya Volkova et al. from **Russia** in their paper "Recommender System for Tourist Itineraries Based on Aspects Extraction from Reviews Corpora" describe a recommender system that takes a set of venue categories of user's interest into ac- count to form a tourist itinerary throughout a city. The system is focused on user preferences in venue aspects. Techniques of such aspects extraction are developed in this paper, in particular from reviews corpora. User preferences are used to weigh aspects associated with particular sights and restaurants. These filtered venues along with time restrictions are subject to submit into the recommender system. A lightweight ontology is discussed which describes the domains of restaurants and sightseeing knowledge and allows venues comparative analysis to enhance the search for relevant venues. The system designed performs automated planning of tourist itineraries, flexible sights searching, and analysis of venues aspects extracted from reviews in Russian.

F. A. K. Hemant from **India** in the paper "Building an Information Extraction and Question Answering Model for Text Based on the Human Brain Process" showcase an information-extraction and question-answering model for text, which is based loosely on the human brain process. The ideology used is based on how humans perceive and interact with text, and the process of storing the text for future reference. Each word of each sentence is cross-referenced and linked with all available information and the answer is given based on matching information found. The model is basic, but the future applications and scope of improvement are also shown.

This issue of the journal will be useful to researchers, students, and practitioners working in the corresponding areas, as well as to public in general interested in advances in computer science, artificial intelligence, and computer engineering.

Alexander Gelbukh

Instituto Politécnico Nacional,
Mexico City, Mexico
Editor-in-Chief

ISSN 2395-8618

Comparing the Black Hole and the Soccer League Competition Algorithms Solving the Set Covering Problem

Adrián Jaramillo, Álvaro Gómez, Broderick Crawford, Ricardo Soto, Fernando Paredes, and Carlos Castro

***Abstract*—The development of techniques to solve the Set Covering Problem (SCP) have given rise a wide range of metaheuristic alternatives, some of them designed from the beginning to operate in binary search spaces, and other considering continuos spaces that requires adaptation intended to work with binary spaces. Black Hole and Soccer League Competition they were designed to work with continuous spaces and they have been adapted to operate in binary spaces: Binary Black Hole and Binary Soccer League Competition, respectively, aimed to solve problems in a binary domain, particularly the SCP. The present paper compare both implementation in a statistical way, involving the use of non-parametric tests and supported by R statistical computing enviroment, considering regularity and consistency of their results when both algorithm implementations are tested on the same benchmark sets.**

***Index Terms*—Optimization, set covering problem, constraint satisfaction, binary black hole algorithm, soccer league competition algorithm, algorithm adaptation, algorithm comparison**

I. Introduction

THE need to find solutions to optimization problems either using complete or approximative techniques, has allowed the development of several alternatives with different approaches and models. Metaheuristic alternatives are suitable for high dimensionality problems where the main target is to find good solutions, but not the ideal optimal, in an acceptable time spend. The question that arises is how to establish if one algorithm implementation has better behavior than other one, or how to quantify the improvements achieved when some modifications or tuning have been introduced to a specific implementation.

This paper address the comparison of two population-based metaheuristic algorithms adapted to work on binary search spaces to solve the Set Covering Problem (SCP). The comparison is performed from an statistical point of view considering the regularity and consistency of their results when they are tested in the same set of benchmarks.

As is the usual in the domain of complex optimization problems helped by the metaheuristics field, each solution strategy arises from behaviors observed from the nature and then mapped to algorithms. The first one, named Soccer League Competition Algorithm (SLC), is based on soccer competitions where the best teams conformed by exceptional players improve their chances to win each match and each player attempts to become a soccer star or a super soccer star player [1]. The second one, named Black Hole Algorithm (BH) [2], [3] is based on previous work of the particle swarm optimization algorithm with newly convergence elements [4], [5]. It defines an universe of a constant number of stars moving around static locations called black holes and when a star is swallowed by a black hole then a new random star is born.

Both algorithms work with a set of individuals moving around a search space but with different strategies, all of them aimed to reach best regions that improve they fitness and escape from the local optima. Both SCP and BH were designed to work in continuous search spaces and its adaptation to a binary domain have been performed by different ways: the Binary Black Hole (BBH) algorithm lies on transfer and binarization functions [6], Binary Soccer League Competition (BSLC) lies on the Hamming distance reduction approach instead.

As mentioned above, the comparison is performed considering a statistical approach based on regularity and consistency of the results. A methodological mean-analysis applying non-parametric statistical tests is performed according preconditions required by each of them. Shapiro-Wilk, Kolmogorov-Smirnoff-Lilliefors, Wilcoxon-Mann-Whitney, Levene, ANOVA an unpaired t-test are used to define a best choice in each of the 55 different scenarios.

The section II formulates the SCP with its main elements. The BH and how it works in searching for optimal is discussed in section III. The section IV describes the original SLC designed for continuous search spaces, while in section IV-C a binary adaptation of SLC is introduced. The comparison for both algorithm implementations are addressed in section VII and in section IX the conclusions are drawn.

Manuscript received on October 10, 2017, accepted for publication on February 21, 2018, published on June 30, 2018.

Adrián Jaramillo, Álvaro Gómez, Broderick Crawford, and Ricardo Soto are with the Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile (e-mail: {adrian.jaramillo.s,alvaro.gomez.r}@mail.pucv.cl, {broderick.crawford,ricardo.soto}@ucv.cl).

Fernando Paredes is with the Universidad Diego Portales, Santiago, Chile (e-mail: fernando.paredes@udp.cl).

Carlos Castro is with the Universidad Federico Santa María, Santiago, Chile (e-mail: carlos.castro@inf.utfsm.cl).

ISSN 2395-8618

II. The Set Covering Problem

The Set Covering Problem is one of 21 NP-Hard problems [7] presents in a wide variety of optimization scenarios.

Since its introduction in 1972 by Karp [8] it has been used in optimization problems of elements locations providing spatial coverage such as telecommunications antennas [9], community services [10], urban transportation crews planning [11], metallurgical industry [12], safety and robustness of data networks [13], construction structural calculations [14], focus of public policies [15] among others.

$$\min \quad C = \sum_{j=1}^{n} c_j x_j \tag{1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \forall i \in \{1, 2, ..., m\} \tag{2}$$

$$x_j \in \{0, 1\} \quad \forall j \in \{1, 2, ..., n\} \tag{3}$$

In general words, let $\mathcal{S}$ be the union of n sets. An element is covered by a set if the element is in the set. A *cover* of $\mathcal{S}$ is a group of the n sets such that every element of $\mathcal{S}$ is covered by at least one set in the group. The SCP challenge is to find a *cover* of $\mathcal{S}$ with minimum size. That is, minimizing expression Eq. (1) and complying with Eq. (2) and Eq. (3).

III. The Black Hole Algorithm

Farahmandian and Hatamlou presents in [2] a strategy intended to find solutions for optimization problems, conceiving the idea of an universe conformed by stars orbiting a unique and fixed center, a *black hole* refered as $\mathbf{X}_{BH}$, in a population-based algorithm approach similar to those used in genetic techniques [16] or particles swarm [17].

The $\mathbf{X}_{BH}$ is a fixed star in the search space, having the best fitness value regarding other stars or, equivalently, the lowest value for a defined function called *objective function* intended to minimize.

The star's motion is performed by an operator of rotation that moves each of them iteratively around $\mathbf{X}_{BH}$, causing along the process the collapse of some stars into the black hole by gravitational effect, the creation of new stars randomly as an exploration strategy, or bringing the creation of a new black hole as an exploitation strategy. The universe's motion process ends when a detention criteria is reached, being the current $\mathbf{X}_{BH}$ the best known solution found for the problem.

A. The Big Bang stage

This stage consists in the creation of an initial universe conformed by a set of $nStar$ stars built randomly. Stars may be replaced during the iteration process but its amount remains fixed throughout the process. The algorithm 1 shows the mechanism for building a new universe, also applied in intermediate steps of star replacement. Let $\mathbf{X}_i$ be a star, then: where $StarBuilder(n)$ function creates a new feasible random binary star, i.e. a feasible solution vector with dimension n.

Algorithm 1 Initial random star builder

```
for i ← 1, nStar do
    X_i ← StarBuilder(n)
```

B. Fitness evaluation

Let $f_{BH}(\mathbf{X}_i)$ be a fitness evaluation function, $f_{BH} : \mathbb{R}^n \rightarrow \mathbb{R}$. The black hole $\mathbf{X}_{BH}$ will be those $\mathbf{X}_i$ with the lowest fitness value regarding the rest of stars in the universe.

C. The rotation operator

The operator of rotation sets a new position for each star $\mathbf{X}_i$ other than $\mathbf{X}_{BH}$ which remains in a fixed position. The new position of $\mathbf{X}_i$ at iteration $t + 1$, considering its initial position at t iteration is defined by Eq. (4) below:

$$\mathbf{X}_i^d(t+1) = \mathbf{X}_i^d(t) + rand()(\mathbf{X}_{BH}^d - \mathbf{X}_i^d(t)), \tag{4}$$

where $i \in \{1, 2, ..., nStars\}$, $\mathbf{X}^d$ stands for any d-dimension of the solution, $\mathbf{X}_{BH}$ is black hole position, $rand()$ is a random number with uniform distribution in [0,1].

D. Collapsing into the black hole

A star closer to the black hole at a distance called *event horizon* is inevitably captured and permanently absorbed by it, being replaced by a new star generated randomly. In other terms, the collapse of a star occurs when it exceeds the radius of Schawarzchild (R).

Farahmandian and Hatamlouy intend in [16] to determinate the distance of a star $\mathbf{X}_i$ to the radius R as:

$$R = \frac{f_{BH}(\mathbf{X}_{BH})}{\sum_{i=1}^{nStars} f(\mathbf{X}_i)} \tag{5}$$

where $f_{BH}(\mathbf{X}_{BH})$ and $f_BH(\mathbf{X}_i)$ are the black hole and the $\mathbf{X}_i$ star fitness value, respectively.

A star $\mathbf{X}_i$ will collapse when its distance at the black hole is less than R as indicated in Eq. (5). Aimed to manage the tolerance threshold calculating the event horizon, we incorporate an additional parameter $s \in [0, 1]$ to the algorithm, to modify the minimum allowable proximity to the black hole, measured in function of its fitness. Thus, a star $\mathbf{X}_i$ will colapse into the black hole if:

$$|f_{BH}(\mathbf{X}_{BH}) - f_{BH}(\mathbf{X}_i)| < sR \tag{6}$$

IV. The Soccer League Competition Algorithm

SLC is introduced by Moosavian in [1] and defines a set of n_{teams} set of players or feasible solutions called $teams$. Each team $\mathcal{T}$ is conformed by n_{fp} fixed players $\mathbf{FP}$ and n_{sp} substitute players $\mathbf{SP}$.

A player $\mathbf{X} = (x^1, x^2, ..., x^d)$ will belong to the fixed or substitute class depending of its performance level rank. The performance level or *power player* is defined by a function $PP : \mathbb{R}^n \rightarrow \mathbb{R}$. If two solutions $\mathbf{X}_i$ and $\mathbf{X}_j$ verifies that

ISSN 2395-8618

$PP(\mathbf{X}_i) > PP(\mathbf{X}_j)$, then we will say that $\mathbf{X}_i$ has a better performance than $\mathbf{X}_j$.

For each team $\mathcal{T}$, the player having the higest player power value is called *super player*, $\mathbf{X}_{SP}$. Likewise, considering all teams we can find the *super star player*, $\mathbf{X}_{SSP}$, as the player with the best power player.

Given the player power function PP, we can generalize and define the *team power* TP as follow:

$$TP = \sum_{\mathbf{X}_k \in \mathcal{T}} \frac{PP(\mathbf{X}_k)}{n_{fp} + n_{sp}} \tag{7}$$

A. Stochastic criteria

Two teams faced in a match will result in one single winner always. If TP_A and TP_B are the team power for $\mathcal{T}_A$ and $\mathcal{T}_B$, respectively, the probability of victory for $\mathcal{T}_A$ facing $\mathcal{T}_B$ is given as follow:

$$PV_A = \frac{TP_A}{TP_A + TP_B} \tag{8}$$

In a similar way, we can calculate the probability of victory for $\mathcal{T}_B$, PV_B. It results that $PV_A + PV_B = 1$. Then, given a random number $r \in [0, 1]$ and PV_A defined as Eq. (8) we can define the winner team in a time t as shown in algorithm 2:

Algorithm 2 Definition of the winner team between $\mathcal{T}_A$ and $\mathcal{T}_B$

```
PV_A ← GetProbabilityOfVictory(T_A, T_B)
r ← rnd(0, 1)
if 0 ≤ r ≤ PV_A then
    T_A is the winner
else
    T_B is the winner
```

B. Movement operators

For the winner team defined above, the **imitation** and **provocation** operators are defined. In the other hand, the **mutation** and **substitution** operators are defined for the looser team. The **imitation** operator will attemp to move each fixed player of the winner team towards $\mathbf{X}_{SSP}$ or $\mathbf{X}_{SP}$ aimed to improve its player power, calculating two feasible candidate solutions, $\mathbf{X}_a$ and $\mathbf{X}_b$, using Eq. (9) and Eq. (10) as follow:

$$\mathbf{X}_a = \mu_1 \mathbf{FP}(t) + \tau_1(\mathbf{X}_{SSP} - \mathbf{FP}(t)) + \tau_2(\mathbf{X}_{SP} - \mathbf{FP}(t)) \tag{9}$$

$$\mathbf{X}_b = \mu_2 \mathbf{FP}(t) + \tau_1(\mathbf{X}_{SSP} - \mathbf{FP}(t)) + \tau_2(\mathbf{X}_{SP} - \mathbf{FP}(t)) \tag{10}$$

where $\mu_1 \sim U(\theta, \beta)$, $\mu_2 \sim U(0, \theta)$, $\theta \in [0, 1]$, $\beta \in [1, 2]$ and $\tau_1, \tau_2 \sim (0, 2)$ are random numbers with uniform distribution as is indicated in [1]. The algorithm 3 shows how imitation operation does work, moving $\mathbf{FP}(t)$ to the new position $\mathbf{FP}(t+1)$ when its player power is improved.

The **provocation** operator will attempt to move each substitute player $\mathbf{SP}$ towards the centroid or gravitational center $\mathbf{G}$ defined by Eq. (11), aimed to improve its player power.

Algorithm 3 Imitation operator

```
X_a ← GetCandidate_a()
X_b ← GetCandidate_b()
if PP(X_a) > PP(FP(t)) then
    FP(t + 1) ← X_a
else if PP(X_b) > PP(FP(t)) then
    FP(t + 1) ← X_b
```

$$\mathbf{G}^d = \frac{\sum_{\mathbf{FP}_i \in \mathcal{T}} \mathbf{FP}_i^d}{n_{fp}} \tag{11}$$

Then, two new candidates X_r and X_s are calculated as follow:

$$\mathbf{X}_r = \mathbf{G} + \chi_1(\mathbf{G} - \mathbf{SP}) \tag{12}$$

$$\mathbf{X}_s = \mathbf{G} + \chi_2(\mathbf{SP} - \mathbf{G}) \tag{13}$$

where $\chi_1 \sim U(0.9, 1)$, $\chi_2 \sim U(0.4, 0.6)$ are random numbers with uniform distribution as is indicated in [1]. The algorithm 4 shows how the provocation operator does work, moving $\mathbf{SP}(t)$ to the new position $\mathbf{SP}(t+1)$ when its player power is improved. In other case, it is replaced by a new random generated feasible solution.

Algorithm 4 Provocation criteria

```
X_r ← GetCandidate_r()
X_s ← GetCandidate_s()
if PP(X_r) > PP(SP(t)) then
    SP(t + 1) ← X_r
else if PP(X_s) > PP(SP(t)) then
    SP(t + 1) ← X_s
else
    SP(t + 1) ← NewPlayer()
```

For the looser team, the fixed players will attempt to apply small changes to avoid repeating the match failure by using some **mutation** operator like Genetic Algorithm (GA). Also, some substitute players will be replaced by promising young talents by applying a crossover operator but not considered in the binary adaptation of SLC.

C. Binary versions of BH and SLC

Gómez introduces in [18] a strategy to allow BH to work in binary search spaces, using transfer and binarization functions, thus mapping non-binary values to the $\{0, 1\}^n$ domain. Jaramillo presents in [19] a binary adaptation approach using Hamming distance reduction instead vectorial algebra, dicretization and binarization functions. For the **imitation** operator it proposes two new candidates $\mathbf{X}_a$ and $\mathbf{X}_b$ as follow:

$$\mathbf{X}_a^d = \begin{cases} \mathbf{X}_{SP}^d & \text{if } rand() \leq p_{imitation} \\ \mathbf{FP}^d(t) & \text{other case} \end{cases} \tag{14}$$

ISSN 2395-8618

$$\mathbf{X}_b^d = \begin{cases} \mathbf{X}_{SSP}^d & \text{if } rand() \leq p_{imitation} \\ \mathbf{FP}^d(t) & \text{other case} \end{cases} \quad (15)$$

where $rand() \sim U(0,1)$ is a random generated value with uniform distribution and $p_{imitation}$ is a probability of imitation defined as an initial parameter of the model. The **provocation** operator uses a new centroid point definition, $\mathbf{BG}$ built from $\mathbf{G}$ in Eq. (11) but considering the probability to have 1 or 0 in the dimension d as follow:

$$\mathbf{BG}^d = \begin{cases} 1 & \text{if } \mathbf{G}^d \geq 0.5 \\ 0 & \text{other case} \end{cases} \quad (16)$$

A **mutation** operator for fixed players could be considered as follow:

$$\mathbf{FP}^d(t+1) = \begin{cases} \mathbf{FP}^d(t) & \text{if } rand() \leq p_{mutation} \\ \neg\mathbf{FP}^d(t) & \text{other case} \end{cases} \quad (17)$$

where $rand() \sim U(0,1)$ is a random generated value with uniform distribution and $p_{mutation}$ is a probability of mutation defined as initial parameter of the model.

V. Solving SCP using BBH and BSLC

BBH and BSLC require both a fitness function definition. For BBH the Eq. (1) of SCP can be used to define $f_{BH}(\mathbf{X}) = 1/\sum x_i c_i$ when SCP faces a minimum optimization problem, or its inverse in case of maximum. In the same way, BSLC can define its player power function as $PP(\mathbf{X}) = 1/\sum x_i c_i$ when SCP faces a minimum optimization problem, or its inverse in case of maximum. Feasibility based on constraints Eq. (2) and Eq. (3) are specific of each implementation and are not covered in this paper.

VI. Performing BBH and BSLC experiments

The implementation of both algorithms were tested using the same SCP benchmark problem sets. Problem sets 4 - 6 are taken from Balas and Ho [20]. Problem sets A - D are from Basley [21]. Problem sets NRE and NRF are taken from [22]. The data sets is provided by the OR-Library website [23] and available for free download in Internet. Problem sets 4 and 5 contain 10 instances each. Problem sets 6, A - D, NRE and NRF contain 5 instances each. Table I shows details for each problem set. The density value corresponds to the percentage of ones in the constraint matrix.

The goal of the comparison is focused in contrasting regularity in the BBH and BSLC outcomes and proximity to the best known solution for each instance tested.

BSLC did use 10 teams, conformed each one of them by 15 fixed and 10 substitute players. BBH did use an universe conformed by 250 stars. Each instance test was run 31 times for both BBH and BSLC in order to obtain a median value for each experiment.

The summary of results is shown in the Table II. The number of constraints, dimensions and the best known solution value Z_{BKS} are shown. The relative percentage difference rpd is defined as $rpd = \frac{min - Z_{BKS}}{Z_{BKS}}$. Each implementation shows min, max, mean and median values obtained per instance.

VII. Statistical comparison approach

In the metaheuristic field, performing a statistical analysis using parametric tests is not suitable as result of the stochastic nature of the evolutionary algorithms. This due to the fact that required conditions as normality, homoscedasticity and independence are not satisfied, as is demonstrated in [24] and [25]. However, when normality is not guaranteed, **Wilcoxon** and **Wilcoxon-Mann-Whitney** might be an appropriate option in order to compare populations, considering medians instead means.

Lanza and Gómez use in [26] a methodological approach to compare different algorithms implementation, considering the regularity and consistency of their results, as is shown in Fig. (1).

Using the Mann-Whitney-Wilcoxon test is possible to check if the population distributions of two evolutionary algorithm outcomes are identical without assuming them to follow the normal distribution. For this purpose, the non-normality and independence conditions must be checked first in order to choose a suitable contrast test.

A. Non-Normality condition

In order to check that a normal distribution is not present in the outcomes, Kolmogorov-Smirnov-Lilliefor (KSL) and Shapiro-Wilk (SW) tests are performed for each instance, helped by R statistical computing environment.

KSL-test [27] is used to compare the accumulative distribution observed with a reference probability distribution, in this case a *normal* distribution; a $p_{value} > 0.05$ implies that there is no evidence to reject the null-hypothesis H_0, that accumulative distribution observed and the reference distribution are the same. In the other hand, SW-test [28] defines a null-hypothesis H_0 as the population is normally distributed; a $p_{value} > 0.05$ implies that there is no evidence to reject H_0.

As the result sets were obtained by evolutionary algorithms, the main idea of this stage is to prove that there is evidence to reject a normal distribution, i.e. the null-hypothesis H_0 in KSL, or SW or both, for each instance.

B. Independence of the samples

Two data samples are independent if they come from distinct populations and the samples do not affect each other. Each run executed corresponds to independiente process running in a virtualized computing environment; the values in one sample (run result) does not affect the values in the other sample, and values in one sample reveal no information about those of the other samples, thus we can establish the sample independence.

C. Statistical results

1) Non-normality.: The results obtained using the R's parametric tests $ks.test$ and $shapiro.test$ are summarized in Table III. It shows the p_{value} obtained per test, algorithm

ISSN 2395-8618

TABLE I
DETAILS OF SCP PROBLEM SETS.

Problem set	Constraints	Dimensions	Density	Instances
4	200	1000	2%	10
5	200	2000	2%	10
6	200	1000	5%	5
A	300	3000	2%	5
B	300	3000	5%	5
C	400	4000	2%	5
D	400	4000	5%	5
NRE	500	5000	10%	5
NRF	500	5000	20%	5

TABLE II
RESULTS PER BENCHMARK INSTANCE AND ALGORITHM IMPLEMENTATION.

Instance	Constr.	Dimension	Z_{BKS}	BBH					BSLC				
				min	$median$	$mean$	max	rpd %	min	$median$	$mean$	max	rpd %
4.1	200	1000	429	447	735.5	685	1268	4.2	429	723.3	722	1395	0
4.2	200	1000	512	517	741.5	696	1154	0.98	512	862.9	831	1375	0
4.3	200	1000	516	527	860.8	769	1780	2.13	517	898.7	824	1450	0.19
4.4	200	1000	494	499	900.8	843	1945	1.01	504	835.2	791	1290	2.02
4.5	200	1000	512	518	863.5	758	1452	1.17	518	851.2	777	1367	1.17
4.6	200	1000	560	612	983.8	861	1821	9.29	562	902.5	815	1496	0.36
4.7	200	1000	430	460	728.7	668	1187	6.98	435	707.8	650	1352	1.16
4.8	200	1000	492	496	843.1	773	1506	0.81	492	772.7	691	1406	0
4.9	200	1000	641	649	1052.8	1079	1818	1.25	646	1002.3	1029	1494	0.78
4.10	200	1000	514	575	837.1	813	1423	11.87	548	832.6	758	1419	6.61
5.1	200	2000	253	255	440.5	414	756	0.79	254	448.3	425	738	0.4
5.2	200	2000	302	316	535.6	489	1109	4.64	305	506.5	483	902	0.99
5.3	200	2000	226	227	372	334	804	0.44	232	387.2	360	684	2.65
5.4	200	2000	242	251	382.6	316	702	3.72	264	369.8	357	614	9.09
5.5	200	2000	211	217	338.2	321	613	2.84	213	329.6	312	519	0.95
5.6	200	2000	213	216	332.8	293	658	1.41	218	325.5	308	456	2.35
5.7	200	2000	293	304	468.5	430	1066	3.75	299	468.8	478	911	2.05
5.8	200	2000	288	302	499.8	487	804	4.86	309	471.2	424	911	7.29
5.9	200	2000	279	311	480.2	459	854	11.47	286	455.9	413	840	2.51
5.10	200	2000	265	287	435.1	399	890	8.3	290	451.4	421	653	9.43
6.1	200	1000	138	139	227	221	381	0.72	144	234.5	212	430	4.35
6.2	200	1000	146	149	251.3	247	410	2.05	154	241	233	453	5.48
6.3	200	1000	145	149	234.5	217	450	2.76	154	230.2	194	379	6.21
6.4	200	1000	131	131	208	181	396	0	136	224.9	223	382	3.82
6.5	200	1000	161	167	239.5	238	386	3.73	161	254.8	237	483	0
A.1	300	3000	253	258	397.1	351	654	1.98	258	397.7	383	722	1.98
A.2	300	3000	252	253	411.9	361	739	0.4	254	424.9	413	689	0.79
A.3	300	3000	232	234	355.4	336	687	0.86	237	375.2	337	709	2.16
A.4	300	3000	234	238	360.8	352	582	1.71	257	392.8	360	703	9.83
A.5	300	3000	236	240	343.9	308	700	1.69	238	352	349	569	0.85
B.1	300	3000	69	70	108.1	100	188	1.45	70	103.6	100	159	1.45
B.2	300	3000	76	78	124.5	116	195	2.63	77	122	112	201	1.32
B.3	300	3000	80	81	133.1	116	228	1.25	83	121	113	212	3.75
B.4	300	3000	79	80	134.8	133	250	1.27	80	127.8	118	283	1.27
B.5	300	3000	72	73	114.3	103	204	1.39	74	114.3	107	225	2.78
C.1	400	4000	227	232	360	322	695	2.2	229	365.1	362	578	0.88
C.2	400	4000	219	220	344.1	321	550	0.46	223	336.5	337	534	1.83
C.3	400	4000	243	246	419.2	381	815	1.23	248	384.9	383	626	2.06
C.4	400	4000	219	219	367.9	344	621	0	219	361.7	334	576	0
C.5	400	4000	215	222	385.3	379	644	3.26	220	333.5	291	569	2.33
D.1	400	4000	60	65	100	91	179	8.33	73	111.6	109	164	21.67
D.2	400	4000	66	72	109.9	100	176	9.09	67	99.6	90	172	1.52
D.3	400	4000	72	74	108.3	95	194	2.78	73	122.3	117	214	1.39
D.4	400	4000	62	66	99.6	96	155	6.45	69	101	93	178	11.29
D.5	400	4000	61	62	92.7	94	147	1.64	63	97.9	86	162	3.28
NRE.1	500	5000	29	30	47.5	44	92	3.45	30	49.5	50	80	3.45
NRE.2	500	5000	30	30	49.2	45	88	0	32	52.1	48	88	6.67
NRE.3	500	5000	27	27	40.8	37	83	0	29	45.2	44	68	7.41
NRE.4	500	5000	28	29	46.3	42	77	3.57	29	46.2	43	76	3.57
NRE.5	500	5000	28	28	44.9	44	81	0	28	41.9	42	68	0
NRF.1	500	5000	14	14	22.7	23	45	0	16	22.2	21	36	14.29
NRF.2	500	5000	15	15	26.5	24	45	0	15	23.9	24	36	0
NRF.3	500	5000	14	15	24	24	44	7.14	14	22.7	22	37	0
NRF.4	500	5000	14	14	21.9	21	35	0	14	23.1	23	43	0
NRF.5	500	5000	13	14	20.9	18	43	7.69	14	21.6	20	37	7.69

ISSN 2395-8618

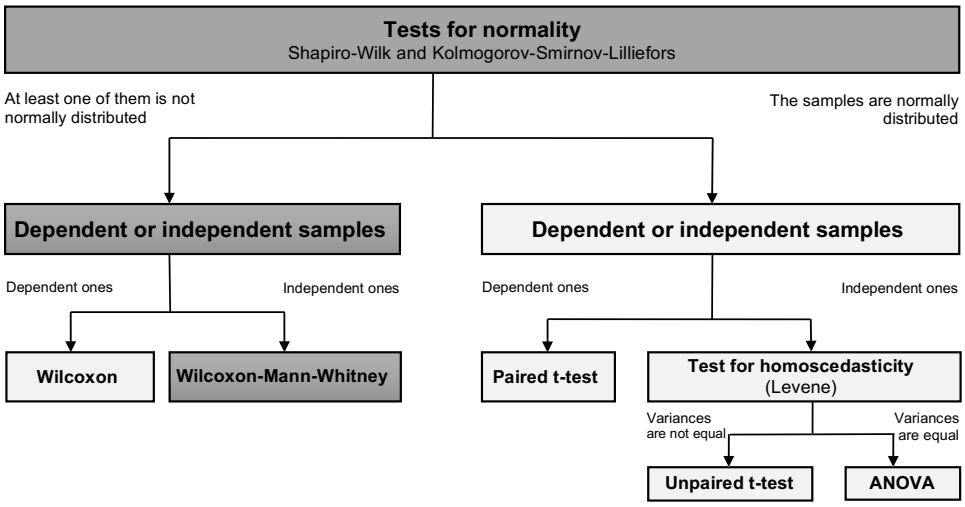


Fig. 1. Statistical methodology chart for 2 samples

implementation and instance. Values less than 0.05 are marked with * symbol. The $\not{H_0}$ in the labeled column $result$ indicates that there is no evidence for normality assumption, i.e. rejecting H_0. When BBH and BSLC normality test give simultaneously evidence to accept H_0, then we can be facing unexpected outcomes, needing a second revision.

Figure (1) addresses the test to apply in each instance and the Table III shows the results. Note that the Wilcoxon-Mann-Whitney test is the predominant. This fact is expected due the no-normality and independence conditions is met by almost all the instance outcomes.

2) Wilcoxon-Mann-Whitney test execution: A cross test between the BBH and BSLC outcomes is performed using R. In both cases, a redefinition of the alternative-hypothesis H_1 is set and the main idea is to reject the null-hypothesis H_0 in order to accept H_1.

The first test defines a H_1 as $\overline{X}_{BBH} > \overline{X}_{BSLC}$. If a $p_{value} < 0.05$ is obtained, then there is evidence to reject H_0, accepting H_1, i.e. BSLC's outcomes are statistically better than BBH. In a similar way, the second test defines a H_1 as $\overline{X}_{BSLC} > \overline{X}_{BBH}$. If a $p_{value} < 0.05$ is obtained then there is evidence to reject H_0, accepting H_1, i.e. BBH's outcomes are statistically better than BSLC.

The summary of results for each instance is shown in Table IV. Values less than 0.05 are marked with * symbol. As we can see, there is not significant evidence to choose one implementation or other for most instances. Figures (2-4) shows boxplot for each instance.

In respect of instances 4.1, 4.2, 6.5, the unpaired test was defined as the suitable test to perform according the methodology in Fig. (1). In a similar way, the ANOVA test has been selected in order to perform comparison in instances 4.9, 5.5, A.3, NRE.5, NRF.2 and NRF.4. However, a normal distribution is an unexpected result in evolutionary algorithms, as previously mentioned in Section VII due to its stochastic nature.

As suggested by Demšar in [29], the Kolmogorov-Smirnov and similar normality test have a little power in detecting abnormalities on small samples. With the purpose of having a third evidence, D'Agostini-Pearson (DA) normality test is applied. The values obtained are shown in Table V, keeping the same evidence of KSL and SW tests. Figures (5-6) show a boxplot for these instances. Regardless the results evidenced in Table V, Wilcoxon-Mann-Whitney test is performed on these datasets to address location comparison under a non-normality assumption.

VIII. Analysis of results

Based on the statistical results obtained by Wilcoxon-Mann-Whitney test, there is a slight tendency to define the BBH algorithm as better than BSLC, regarding all instances covered in the experiments performed in this work.

For instances 5.3, D.1, D.3, NRE.3, NRF.5 there is evidence towards BBH and it can be confirmed by their respectives boxplot in Fig. (2) and Fig. (4) where BBH have better location of the median than BSLC. In the other hand, instances C.5 and D.2 show evidence towards BSLC, and its respective boxplots in Fig. (4) confirms the above, where BLSC shows better location of the median than BBH.

The non-normality condition appears to be not required in order to perform a Wilcoxon-Mann-Whitney test when data sets have been generated by an evolutive algorithm. This is evidenced comparing numerical results in Table VI against the boxplot representation for this instances in Fig. (5 - 6), where both conclusions that can emerge do fit. In particular, the instance 4.2 shows a better location of the BBH's median value than BSLC and it can be evidenced numerically by the result in Table VI . The other instances referred in the same group with unexpected normality distribution evidence, as result of

ISSN 2395-8618

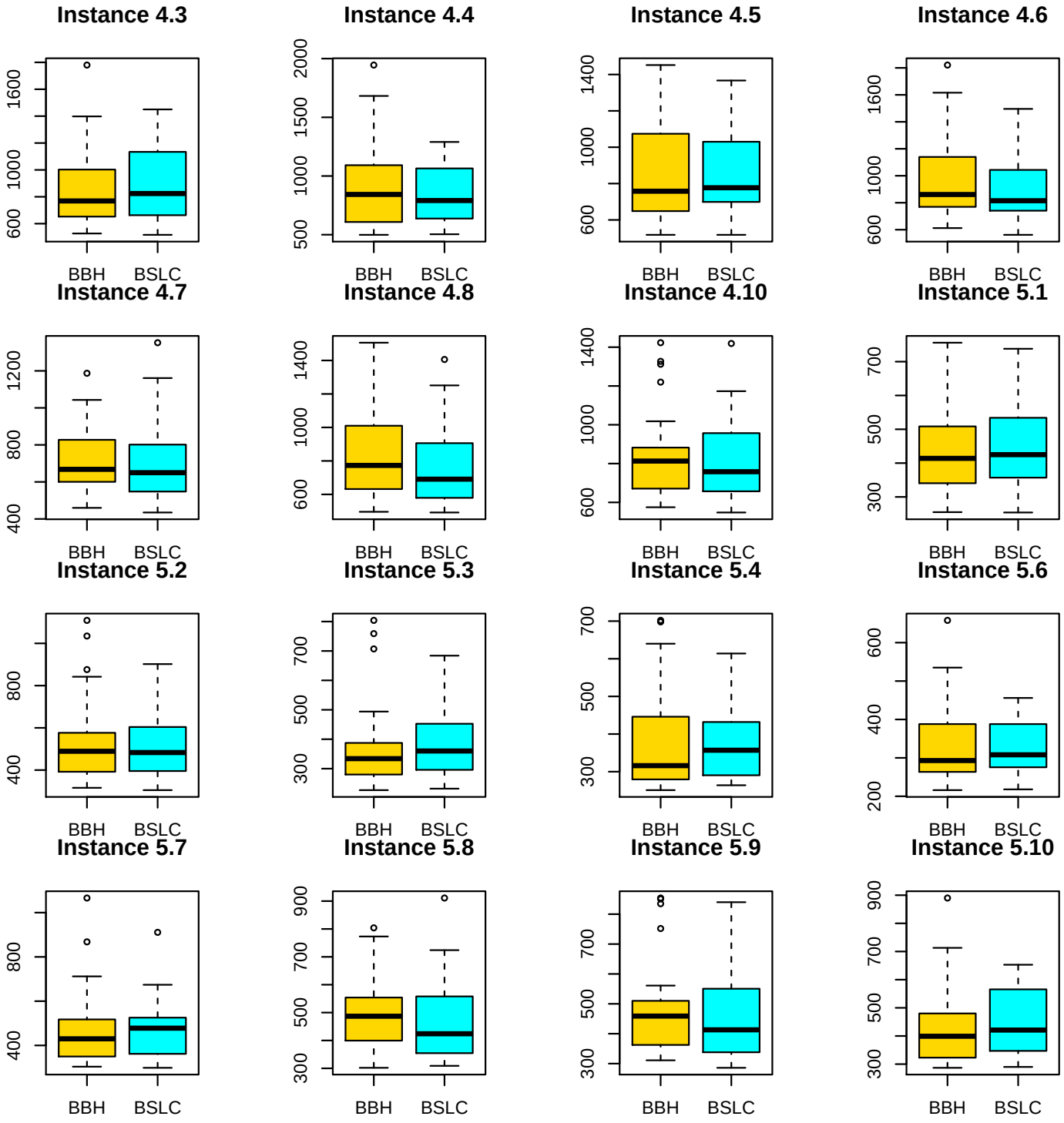


Fig. 2. Boxplot for outcomes addressed by Wilcoxon-Mann-Whitney test.

ISSN 2395-8618

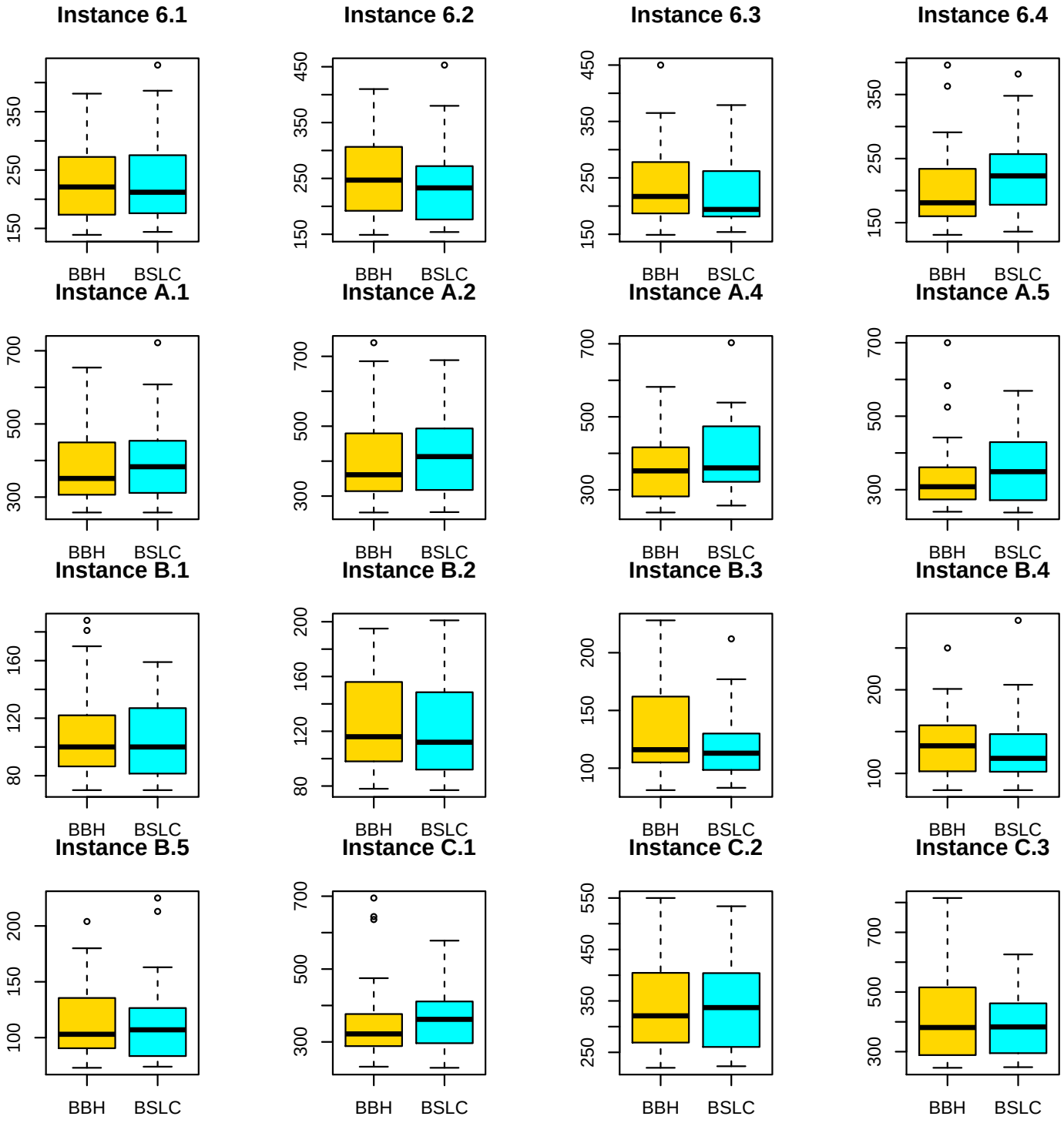


Fig. 3. Boxplot for outcomes addressed by Wilcoxon-Mann-Whitney test.

ISSN 2395-8618

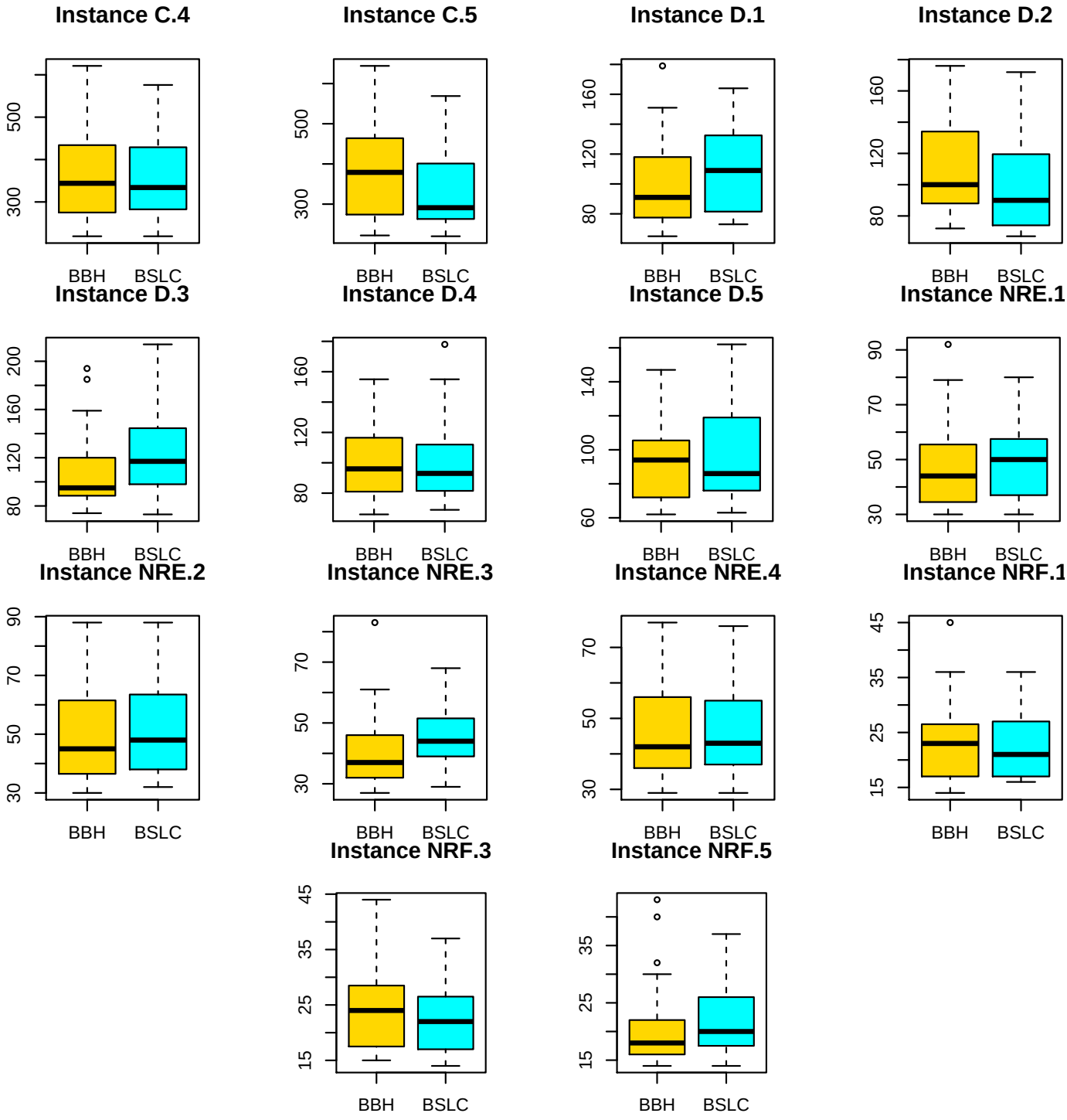


Fig. 4. Boxplot for outcomes addressed by Wilcoxon-Mann-Whitney test.

ISSN 2395-8618

TABLE III
SUMMARY OF RESULTS FOR THE NORMALITY PARAMETRIC TEST AND THE TEST TO APPLY FOR ALGORITHM OUTCOMES.

Instance	BBH			BSLC			Levene	Test to apply
	KSL p_{value}	SW p_{value}	result	KSL p_{value}	SW p_{value}	result	p_{value}	
4.1	0.078	0.933	H_0	0.857	0.220	H_0	0.012	Unpaired t-test
4.2	0.052	0.065	H_0	0.713	0.080	H_0	0.019	Unpaired t-test
4.3	0.589	*0.013	~~H_0~~	0.453	0.053	H_0		Wilcoxon-Mann-Whitney
4.4	0.711	*0.029	~~H_0~~	0.711	*0.046	~~H_0~~		Wilcoxon-Mann-Whitney
4.5	0.327	*0.025	~~H_0~~	0.481	0.259	H_0		Wilcoxon-Mann-Whitney
4.6	0.411	*0.006	~~H_0~~	0.327	*0.024	~~H_0~~		Wilcoxon-Mann-Whitney
4.7	0.364	0.059	H_0	0.483	*0.024	~~H_0~~		Wilcoxon-Mann-Whitney
4.8	0.251	*0.008	~~H_0~~	0.421	*0.018	~~H_0~~		Wilcoxon-Mann-Whitney
4.9	0.051	0.370	H_0	0.634	0.059	H_0	0.319	ANOVA
4.10	0.965	0.378	H_0	0.513	*0.036	~~H_0~~		Wilcoxon-Mann-Whitney
5.1	0.623	*0.039	~~H_0~~	0.272	*0.021	~~H_0~~		Wilcoxon-Mann-Whitney
5.2	0.860	*0.007	~~H_0~~	0.441	*0.049	~~H_0~~		Wilcoxon-Mann-Whitney
5.3	0.891	0.071	H_0	0.495	*0.007	~~H_0~~		Wilcoxon-Mann-Whitney
5.4	0.079	*0.000	~~H_0~~	0.655	*0.008	~~H_0~~		Wilcoxon-Mann-Whitney
5.5	0.690	0.060	H_0	0.860	0.084	H_0	0.969	ANOVA
5.6	0.271	*0.028	~~H_0~~	0.639	0.093	H_0		Wilcoxon-Mann-Whitney
5.7	0.495	*0.017	~~H_0~~	0.633	0.074	H_0		Wilcoxon-Mann-Whitney
5.8	0.944	0.351	H_0	0.632	*0.011	~~H_0~~		Wilcoxon-Mann-Whitney
5.9	0.309	*0.022	~~H_0~~	0.539	*0.025	~~H_0~~		Wilcoxon-Mann-Whitney
5.10	0.398	*0.003	~~H_0~~	0.350	*0.007	~~H_0~~		Wilcoxon-Mann-Whitney
6.1	0.543	0.069	H_0	0.456	*0.013	~~H_0~~		Wilcoxon-Mann-Whitney
6.2	0.919	0.214	H_0	0.864	*0.030	~~H_0~~		Wilcoxon-Mann-Whitney
6.3	0.645	*0.043	~~H_0~~	0.104	*0.000	~~H_0~~		Wilcoxon-Mann-Whitney
6.4	0.220	*0.032	~~H_0~~	0.916	0.127	H_0		Wilcoxon-Mann-Whitney
6.5	0.052	0.168	H_0	0.674	0.056	H_0	0.009	Unpaired t-test
A.1	0.294	*0.005	~~H_0~~	0.847	0.070	H_0		Wilcoxon-Mann-Whitney
A.2	0.085	*0.001	~~H_0~~	0.357	0.072	H_0		Wilcoxon-Mann-Whitney
A.3	0.938	0.095	H_0	0.508	0.061	H_0	0.256	ANOVA
A.4	0.937	0.068	H_0	0.505	*0.043	~~H_0~~		Wilcoxon-Mann-Whitney
A.5	0.695	0.072	H_0	0.593	*0.050	~~H_0~~		Wilcoxon-Mann-Whitney
B.1	0.536	*0.029	~~H_0~~	0.783	0.093	H_0		Wilcoxon-Mann-Whitney
B.2	0.630	*0.018	~~H_0~~	0.475	0.053	H_0		Wilcoxon-Mann-Whitney
B.3	0.118	*0.012	~~H_0~~	0.683	*0.012	~~H_0~~		Wilcoxon-Mann-Whitney
B.4	0.387	0.137	H_0	0.585	*0.049	~~H_0~~		Wilcoxon-Mann-Whitney
B.5	0.409	*0.036	~~H_0~~	0.715	0.086	H_0		Wilcoxon-Mann-Whitney
C.1	0.515	*0.039	~~H_0~~	0.908	0.077	H_0		Wilcoxon-Mann-Whitney
C.2	0.377	*0.026	~~H_0~~	0.718	0.089	H_0		Wilcoxon-Mann-Whitney
C.3	0.722	*0.010	~~H_0~~	0.615	0.120	H_0		Wilcoxon-Mann-Whitney
C.4	0.709	*0.020	~~H_0~~	0.608	*0.035	~~H_0~~		Wilcoxon-Mann-Whitney
C.5	0.602	0.134	H_0	0.110	*0.009	~~H_0~~		Wilcoxon-Mann-Whitney
D.1	0.372	*0.014	~~H_0~~	0.483	*0.008	~~H_0~~		Wilcoxon-Mann-Whitney
D.2	0.267	0.063	H_0	0.216	*0.001	~~H_0~~		Wilcoxon-Mann-Whitney
D.3	0.313	*0.014	~~H_0~~	0.918	0.292	H_0		Wilcoxon-Mann-Whitney
D.4	0.713	0.183	H_0	0.430	*0.035	~~H_0~~		Wilcoxon-Mann-Whitney
D.5	0.420	*0.021	~~H_0~~	0.307	*0.005	~~H_0~~		Wilcoxon-Mann-Whitney
NRE.1	0.687	*0.015	~~H_0~~	0.665	0.080	H_0		Wilcoxon-Mann-Whitney
NRE.2	0.642	*0.041	~~H_0~~	0.665	*0.028	~~H_0~~		Wilcoxon-Mann-Whitney
NRE.3	0.108	*0.001	~~H_0~~	0.994	0.794	H_0		Wilcoxon-Mann-Whitney
NRE.4	0.237	*0.002	~~H_0~~	0.510	0.051	H_0		Wilcoxon-Mann-Whitney
NRE.5	0.763	0.057	H_0	0.926	0.222	H_0	0.119	ANOVA
NRF.1	0.154	0.051	H_0	0.366	*0.005	~~H_0~~		Wilcoxon-Mann-Whitney
NRF.2	0.070	0.019	H_0	0.062	0.098	H_0	0.281	ANOVA
NRF.3	0.698	*0.046	~~H_0~~	0.812	0.082	H_0		Wilcoxon-Mann-Whitney
NRF.4	0.919	0.058	H_0	0.900	0.055	H_0	0.134	ANOVA
NRF.5	0.188	*0.007	~~H_0~~	0.093	*0.011	~~H_0~~		Wilcoxon-Mann-Whitney

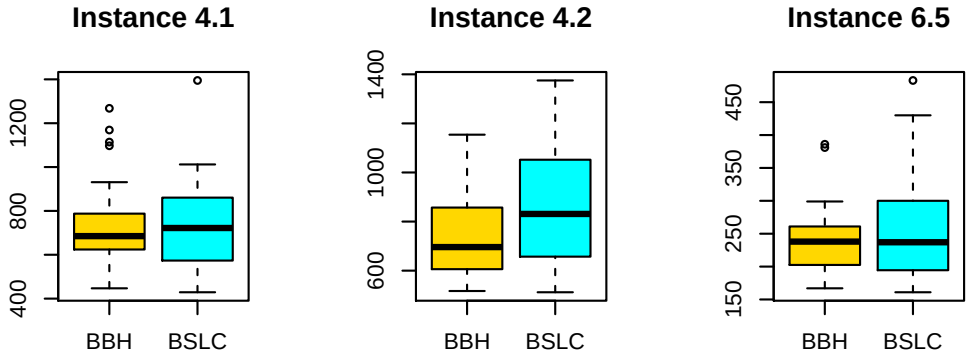


Fig. 5. Boxplot for outcomes with normal distribution evidence for unpaired-test.

ISSN 2395-8618

TABLE IV
WILCOXON-MANN-WHITNEY TEST RESULTS.

Instance	BKS	Median		RPD		Wilcoxon-Mann-Whitney test		Algorithm Selection
		BBH	BSLC	BBH	BSLC	$H_1 : \overline{X}_{BBH} > \overline{X}_{BSLC}$	$H_1 : \overline{X}_{BSLC} > \overline{X}_{BBH}$	
4.3	516	769	824	0.02	0	0.8	0.2	indistinct
4.4	494	843	791	0.01	0.02	0.36	0.64	indistinct
4.5	512	758	777	0.01	0.01	0.54	0.46	indistinct
4.6	560	861	815	0.09	0	0.27	0.74	indistinct
4.7	430	668	650	0.07	0.01	0.2	0.81	indistinct
4.8	492	773	691	0.01	0	0.11	0.89	indistinct
4.10	514	813	758	0.12	0.07	0.75	0.26	indistinct
5.1	253	414	425	0.01	0	0.57	0.43	indistinct
5.2	302	489	483	0.05	0.01	0.76	0.24	indistinct
5.3	226	334	360	0	0.03	0.95	*0.05	BBH
5.4	242	316	357	0.04	0.09	0.7	0.31	indistinct
5.6	213	293	308	0.01	0.02	0.68	0.33	indistinct
5.7	293	430	478	0.04	0.02	0.79	0.22	indistinct
5.8	288	487	424	0.05	0.07	0.12	0.88	indistinct
5.9	279	459	413	0.11	0.03	0.58	0.42	indistinct
5.10	265	399	421	0.08	0.09	0.88	0.12	indistinct
6.1	138	221	212	0.01	0.04	0.51	0.5	indistinct
6.2	146	247	233	0.02	0.05	0.18	0.83	indistinct
6.3	145	217	194	0.03	0.06	0.34	0.66	indistinct
6.4	131	181	223	0	0.04	0.94	0.06	indistinct
A.1	253	351	383	0.02	0.02	0.44	0.57	indistinct
A.2	252	361	413	0	0.01	0.78	0.23	indistinct
A.4	234	352	360	0.02	0.1	0.86	0.15	indistinct
A.5	236	308	349	0.02	0.01	0.91	0.09	indistinct
B.1	69	100	100	0.01	0.01	0.55	0.45	indistinct
B.2	76	116	112	0.03	0.01	0.45	0.56	indistinct
B.3	80	116	113	0.01	0.04	0.09	0.91	indistinct
B.4	79	133	118	0.01	0.01	0.21	0.79	indistinct
B.5	72	103	107	0.01	0.03	0.34	0.67	indistinct
C.1	227	322	362	0.02	0.01	0.92	0.08	indistinct
C.2	219	321	337	0	0.02	0.44	0.56	indistinct
C.3	243	381	383	0.01	0.02	0.35	0.66	indistinct
C.4	219	344	334	0	0	0.43	0.58	indistinct
C.5	215	379	291	0.03	0.02	*0.04	0.96	BSLC
D.1	60	91	109	0.08	0.22	0.97	*0.04	BBH
D.2	66	100	90	0.09	0.02	*0.05	0.95	BSLC
D.3	72	95	117	0.03	0.01	0.99	*0.01	BBH
D.4	62	96	93	0.06	0.11	0.37	0.64	indistinct
D.5	61	94	86	0.02	0.03	0.68	0.33	indistinct
NRE.1	29	44	50	0.03	0.03	0.86	0.14	indistinct
NRE.2	30	45	48	0	0.07	0.77	0.23	indistinct
NRE.3	27	37	44	0	0.07	0.99	*0.01	BBH
NRE.4	28	42	43	0.04	0.04	0.62	0.39	indistinct
NRF.1	14	23	21	0	0.14	0.63	0.38	indistinct
NRF.3	14	24	22	0.07	0	0.22	0.78	indistinct
NRF.5	13	18	20	0.08	0.08	0.97	*0.03	BBH

TABLE V
NORMALITY TEST RESULTS FOR INSTANCES WITH NORMAL DISTRIBUTION EVIDENCE ACCORDING KS AND SW TESTS.

Instance	BBH			BSLC		
	KSL p_{value}	SW p_{value}	DA p_{value}	KSL p_{value}	SW p_{value}	DA p_{value}
4.1	0.078	0.933	**0.89**	0.857	0.220	**0.36**
4.2	0.052	0.065	**0.07**	0.713	0.080	**0.12**
4.9	0.051	0.370	**0.31**	0.634	0.059	**0.12**
5.5	0.690	0.060	**0.13**	0.860	0.084	**0.13**
6.5	0.052	0.168	**0.26**	0.674	0.056	**0.11**
A.3	0.938	0.095	**0.23**	0.508	0.061	**0.08**
NRE.5	0.763	0.057	**0.18**	0.926	0.222	**0.35**
NRF.2	0.070	0.019	**0.18**	0.062	0.098	**0.33**
NRF.4	0.919	0.058	**0.36**	0.900	0.055	**0.29**

KSL and SW tests, show statistically no preference towards BBH or BSLC. That is a preliminary conclusion that must be confirmed or rejected by performing more comprehensive experiments that are beyond the scope of this paper.

IX. CONCLUSIONS

In this paper an approach to compare two algorithms implementation has been performed using non-parametric test. The outcomes for each algorithm implementation have been analyzed by a statistical approach which concludes that BBH's outcomes shows in general a better regularity and consistency than BSLC when they are tested over 55 benchmark for the SCP. It is possible to observe a correlation between the statical and the empirical implementation selection, indicating that 85.5% of the instances there is no preferences towards BBH or BSLC.

The Shapiro-Wilk and Kolmogorov-Smirnov tests could be not enough in order to confirm o reject evidence for a normal distribution when a sample is obtained from an evolutionary algorithm, as has been shown in the experiments for certain instances in this paper. This can be due irregularity in the sample or a result related with the sample size, where small values can give distorted results, as

ISSN 2395-8618

TABLE VI
WILCOXON-MANN-WHITNEY TEST RESULTS.

Instance	BKS	Median		RPD		Wilcoxon-Mann-Whitney test		Algorithm Selection
		BBH	BSLC	BBH	BSLC	$H_1 : \overline{X}_{BBH} > \overline{X}_{BSLC}$	$H_1 : \overline{X}_{BSLC} > \overline{X}_{BBH}$	
4.1	429	685	722	0.04	0	0.68	0.33	indistinct
4.2	512	696	831	0.01	0	0.96	*0.04	BBH
4.9	641	1079	1029	0.01	0.01	0.31	0.7	indistinct
5.5	211	321	312	0.03	0.01	0.51	0.49	indistinct
6.5	161	238	237	0.04	0	0.76	0.25	indistinct
A.3	232	336	337	0.01	0.02	0.73	0.28	indistinct
NRE.5	28	44	42	0	0	0.17	0.83	indistinct
NRF.2	15	24	24	0	0	0.28	0.73	indistinct
NRF.4	14	21	23	0	0	0.75	0.25	indistinct

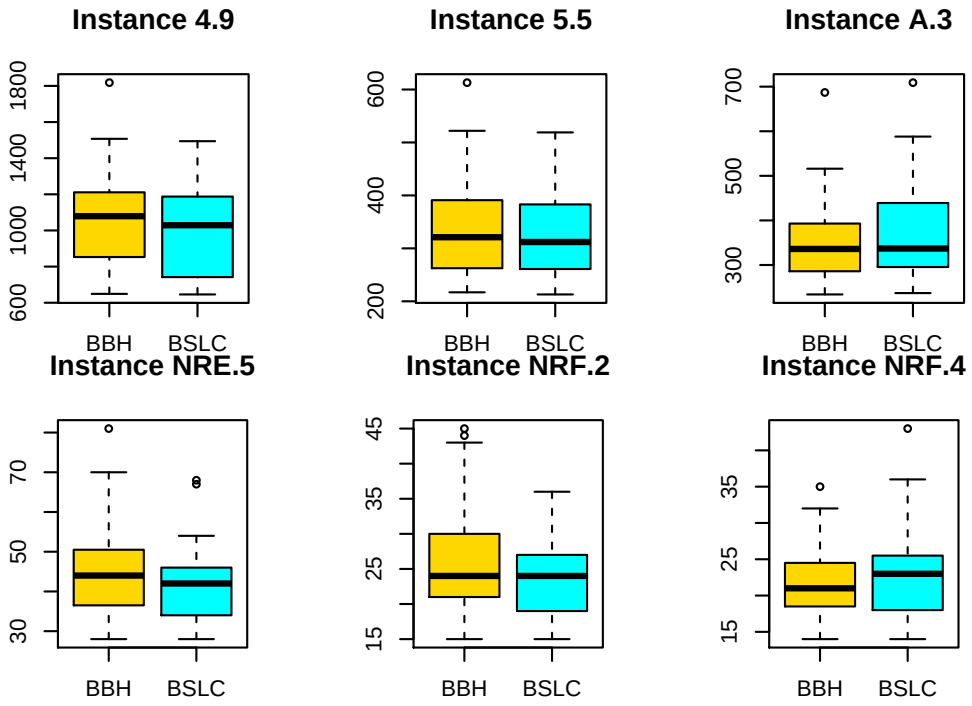


Fig. 6. Boxplot for outcomes with normal distribution evidence for ANOVA test.

is suggested by Demšar in previous papers, requiring more experiments to be addressed in order to set a conclusion properly. How ever, regardless Kolmogorov-Smirnov and Shapirho-Wilk, non-normality condition appears to be not required in order to perform a Wilcoxon-Mann-Whitney test when data sets have been generated by an evolutive algorithm.

ACKNOWLEDGEMENTS

Broderick Crawford is supported by grant CONICYT / FONDECYT / REGULAR 1171243. Ricardo Soto is supported by grant CONICYT / FONDECYT / INICIACION / 11130459. Adrián Jaramillo and Álvaro Gómez are supported by Postgraduate Grant Pontificia Universidad Católica de Valparaiso 2017 (INF-PUCV 2017).

REFERENCES

[1] N. Moosavian, "Soccer league competition algorithm, a new method for solving systems of nonlinear equations," *Scientific Research*, vol. 4, pp. 7–16, 2014. [Online]. Available: http://www.scirp.org/journal/PaperDownload.aspx?paperID=40822

[2] M. Farahmandian and A. Hatamlou, "Solving optimization problems using black hole algorithm," *Journal of Advanced Computer Science & Technology*, vol. 4, no. 1, p. 68, feb 2015. [Online]. Available: https://doi.org/10.14419/jacst.v4i1.4094

[3] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information sciences*, vol. 222, pp. 175–184, 2013.

[4] A. P. Piotrowski, J. J. Napiorkowski, and P. M. Rowinski, "How novel is the "novel" black hole optimization approach?" *Inf. Sci.*, vol. 267, pp. 191–200, May 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2014.01.026

[5] J. Zhang, K. Liu, Y. Tan, and X. He, "Random black hole particle swarm optimization and its application," in *2008 International Conference on Neural Networks and Signal Processing*, June 2008, pp. 359–365.

[6] S. Mirjalili and A. Lewis, "S-shaped versus v-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.swevo.2012.09.002

[7] D. Hochba, "Approximation algorithms for np-hard problems," *ACM SIGACT News*, pp. 40–52, 1997.

[8] Karp, "Reducibility among combinatorial problems," urlhttp://http://www.brunel.ac.uk/ mastjjb/jeb/info.html, 1972.

[9] F. E.Alba, G.Molina, "Optimal placement of antennae using meta-heuristics," *NMA'06 Proceedings of the 6th international conference on*

ISSN 2395-8618

Numerical methods and applications, pp. 214–222, 2006.

[10] F. Q. Kevin Curtin, Karen Hayslett-Mc Call, "Determining optimal police patrol areas with maximal covering and backup covering location models," *Netw Spat Econ*, 2007.

[11] M. Desrochers and F. Soumis, "A column generation approach to the urban transit crew scheduling problem," *Transportation Science*, vol. 23, no. 1, pp. 1–13, 1989.

[12] F. Vasko, F. Wolf, and K. Stott, "Optimal selection of ingot sizes via set covering," *Operations Research*, vol. 35, no. 3, pp. 346–353, 1987.

[13] M. Bellmore and H. D. Ratliff, "Optimal defense of multi-commodity networks," *Management Science*, vol. 18, no. 4-part-i, pp. B–174, 1971.

[14] F. Amini and P. Ghaderi, "Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers," *Applied Soft Computing*, vol. 13, no. 5, pp. 2272–2280, 2013.

[15] D. Forney, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, 1966.

[16] D. Goldberg, "Genetic algorithms in search, optimization and machine learning," *Addison-Wesley Longman Publishing Co., Inc.*, 1989.

[17] J. K. Russ Eberhart, "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS '95*, pp. 39–43, 1995.

[18] Á. G. Rubio, B. Crawford, R. Soto, A. Jaramillo, S. M. Villablanca, J. Salas, and E. Olguín, "An binary black hole algorithm to solve set covering problem," in *Trends in Applied Knowledge-Based Systems and Data Science - 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*, 2016, pp. 873–883. [Online]. Available: https://doi.org/10.1007/978-3-319-42007-3_74

[19] A. Jaramillo, B. Crawford, R. Soto, S. M. Villablanca, Á. G. Rubio, J. Salas, and E. Olguín, "Solving the set covering problem with the soccer league competition algorithm," in *Trends in Applied Knowledge-Based Systems and Data Science - 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*, 2016, pp. 884–891. [Online]. Available: https://doi.org/10.1007/978-3-319-42007-3_75

[28] S. S. SHAPIRO and M. B. WILK, "An analysis of variance test for normality (complete samples)?" *Biometrika*, vol. 52, no. 3-4, p. 591, 1965. [Online]. Available: +http://dx.doi.org/10.1093/biomet/52.3-4.591

[20] E. Balas and A. Ho, *Set covering algorithms using cutting planes, heuristics, and subgradient optimization: A computational study*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1980, pp. 37–60. [Online]. Available: https://doi.org/10.1007/BFb0120886

[21] J. Beasley, "An algorithm for set covering problem," *European Journal of Operational Research*, vol. 31, no. 1, pp. 85 – 93, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037722178790141X

[22] J. E. Beasley, "A lagrangian heuristic for set-covering problems," *Naval Research Logistics (NRL)*, vol. 37, no. 1, pp. 151–164, 1990. [Online]. Available: http://dx.doi.org/10.1002/1520-6750(199002)37:1⟨151::AID-NAV3220370110⟩3.0.CO;2-2

[23] "OR-Library a collection of test data sets for a variety of or problems," http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html, accessed: 2017-04-21.

[24] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, p. 617, May 2008. [Online]. Available: https://doi.org/10.1007/s10732-008-9080-4

[25] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011. [Online]. Available: https://doi.org/10.1016/j.swevo.2011.02.002

[26] J. M. Lanza-Gutiérrez and J. A. G. Pulido, "Assuming multiobjective metaheuristics to solve a three-objective optimisation problem for relay node deployment in wireless sensor networks," *Appl. Soft Comput.*, vol. 30, pp. 675–687, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2015.01.051

[27] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, no. 318, pp. 399–402, 1967. [Online]. Available: http://www.jstor.org/stable/2283970

[29] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1248547.1248548

ISSN 2395-8618

Exploration, Exploitation Phenomena and Regression Analysis: Propensity Metric, Anomaly Reduction, Dimensionality Reduction

Chaman Lal Sabharwal

***Abstract*--The classical Ordinary linear Least Square approximation (OLS) model has been used as the best fit regression for linear trend data. In data analysis, the accuracy of analysis depends on the model as well as the metric used to measure the error. Singular Value Decomposition (SVD) is also applied for Normal linear Least Square (NLS) approximation along the perpendicular to the approximating line. The OLS line is not sensitive to temporal variation in time variables whereas SVD is sensitive, it renders OLS less suitable for time sensitive data. Both OLS and SVD use quantitative metric for regression analysis, and SVD has inherent constraints. Propensity score analysis is an innovative class new technique for qualitative error analysis. Propensity score method is easier to communicate to non-expert audience. Moreover, propensity score estimates are often more robust than the percent error estimates of predicted values over the true values. Herein we present a hybrid algorithm that achieves a balance between quantitative and qualitative approximation accuracy of both OLS and NLS (SVD). This metric has also proved useful for evaluating the effects of treatments in real patient data. This technique is also suitable for anomaly removal. Visualization is a preferred way to ascertain the quality of a new algorithm and is used to demonstrate the hybrid algorithm. We have applied this criteria for comparison with other existing methods. We have found that this technique is reliable and preferable to explain to the expert as well as non-expert. The empirical tests show the accuracy improvements over conventional methods.**

***Index Terms*--least square regression, singular value decomposition, propensity, anomaly, accuracy, precision, learning management systems**

I. INTRODUCTION

The regression analysis is used to determine the correlation between variables and predicting value of dependent variables. The linear regression is a reliable model to predict the value of a dependent variable[1]. This assumes that only one of the variables has error. In empirical data, sometimes the error permeates both the variables [2]. There may be other ways also to establish relation between independent and dependent variables, then we have to distinguish between different universally accepted minimum-error algorithms [3]. For data analysis, most of the time raw data is not directly applicable to analysis algorithms directly. It is mandatary to cleanse data for reliable and accurate regression analysis. Thus, it is expected that the data is accurate otherwise the prediction analysis will be unreliable. If the data is correlated and noisy, it is indispensable to transform the data into uncorrelated and noise free data to prevent overfitting. In such cases, anomaly reduction is mandatory. Some smoothing operation is performed to bring data in line with the approximation concept before applying the learning algorithm. Cleansing is a natural phenomenon, e.g. the physicians use sharp blades to perform incisions, we wash edibles before eating to stay healthy. Data smoothing may be performed via filtering with some kernel or via data noise reduction. Furthermore, the numerical data may be standardized by mean-centering and unit standard deviation etc. Some related algorithms are not equivalent [4]. The approximation error measurement depends on the metric applied to analyze approximation. Cognitive modelling is one of the representative research methods in cognitive sciences. For cognitive model to be viable, it must be verifiable by using well thought metric [5]. We leverage these techniques to devise a cognitively acceptable minimum-error scheme based on propensity metric in conjunction with Euclidean metric. Data dimensionality reduction can be effectively done via SVD, SVD uses dimension reduction operation in the latent space. This step yields noise reduction when the data is transformed back to original space. Such algorithms use relaxation technique to obtain improved hybrid approximation algorithm.

For multivariable data, (x,y), x, y are vectors, most of the time y is scaler valued. The simplest case occurs in 2D when both x, y are scalar valued, it is easy to comprehend. In the simplest case, linear least square regression is a straight line.

Manuscript received on January 10, 2018, accepted for publication on May 7, 2018, published on June 30, 2018.

Chaman Lal Sabharwal is with the Missouri University of Science and Technology Rolla, MO-65409, USA (e-mail: chaman@mst.edu).

ISSN 2395-8618

This linear representation model approximates the non-parametric data points (x_i,y_i) with points (x_p,y_p) on a parametric line. Since a line is uniquely defined by two points, it has two parameters, intercept, a, and slope or elevation, b. The line is a parametric representation of data. One of the models, measures error along the y-axis. In other words, $x_i = x_{ip}$, $y_{ip} = a + b\ x_{ip}$ such that the sum of squares of errors is minimum, error $E_1 = \sum_{k=1,n} (x_k - x_{kp})^2 + (y_k - y_{kp})^2 = \sum_{k=1,n} (y_k - a - b\ x_k)^2$.

In statistical analysis, the accuracy of approximation depends on several parameters. One such parameter is the metric used to measure the approximation error. Each metric has its own merits. For OLS, there are several issues. For least square approximation, it is in fact approximation in y direction, not min distance perpendicular to the approximation line [6], [7], [8]. In order to correct this, we devise a true line at min-distance from the input data, normal distance least square fit line, NLS [9]. We call it normal linear least square approximation (NLS) similar to ordinary linear least square approximation (OLS). NLS may become complicated for multiple dimensions, we also show that linear algebra SVD can be leveraged to achieve NLS more easily [10]. Finally, we see that OLS is not sensitive to data spread, NLS will also correct this deficiency of OLS. We also define a new metric, propensity scoring metric (PSM) for OLS, NLS and hybrid algorithms for pairwise comparison. Propensity score has been used in other areas for estimating the effect of a treatment, policy or other causal effects [11]. We will show the effect of new metric as compared to OLS and NLS metrics. We show that the hybrid algorithm achieves a balance between quantitative and qualitative approximation accuracy of both OLS and SVD. Also, it will be shown that it can be used for noise and anomaly reduction. Thus, there are several approaches to approximate data linearly: ordinary linear least square regression (OLS), (new) normal linear least square regression (NLS), singular value decomposition linear least square regression (SVD), (new) hybrid linear least square regression (HLA). To measure the accuracy of approximation, there are several measures: quantitative and qualitative. Knowing what technique and metric to use makes all the difference in analysis and makes most out of data. That way one spends less time on justifying the conclusions. The challenge is the decision making on the metric used to approximate. The intent of this paper is the design a greedy(hybrid) algorithm that yields better approximation than the OLS and NLS/SVD approximation algorithms, also a way to detect and remove anomalies in the training data.

The paper is organized as follows: Section II gives background and justification for the work. It describes OLS, NLS in R^n and computation by mean-centering data, Section III derives new NLS formulation, Section IV describes SVD and it connection to NLS, Section V gives new hybrid greedy algorithm and its implementation, error analysis of OLS, NLS/SVD, and Hybrid algorithms is provided with respect to both metrics, it introduces propensity score metric (PSM) and anomaly reduction, Section VI is conclusion, Section VII is an appendix giving all the necessary details about linear algebra.

II. BACKGROUND

Here data is represented as a matrix of real values. It is easier to work with data if it is standardized. Simple example of standardization is mean-centering data with unit standard deviation. Ordinarily the reference point of data is the origin, mean-centering implies that the centroid of data is translated to the origin. We will soon see how mean-centering simplifies the computations.

Let the data be represented by an m×n real matrix A, i.e., m rows of n-vectors or n columns of m-vectors. If **x** is column of A, the mean of **x** is denoted by $\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \frac{\sum_{i=1,m} x_i}{m}$. To centralized **x,** it is translated to $\mathbf{x} - \bar{\mathbf{x}}$. Similarly, if **y** is row of A, it is centralized as $\mathbf{y} - \bar{\mathbf{y}}$, where the mean of **y** is $\bar{\mathbf{y}} = \frac{\sum_{i=1,n} y_i}{n}$. Further, if **x** and **y** are both rows (or both columns), the mean of dot product of **x** and **y** is denoted by $\overline{\mathbf{xy}} = \frac{\mathbf{x \bullet y}}{n} = \frac{\sum_{i=1,n} x_i y_i}{n}$, for **x** = **y**, it is denoted by $\overline{\mathbf{x}^2} = \frac{\mathbf{x \bullet x}}{n} = \frac{\sum_{i=1,n} {x_i}^2}{n}$. Most of the linear transformations are performed by means of matrix multiplication, for example, centralization is a linear transformation for mean-centering a matrix [12]. There is an immaculate transformation T_m to mean-center the columns of A as follows. Let I_m be m*x*m identity matrix, $\mathbf{e_m}$ be a column m-vector of ones, and $T_m = I_m - \mathbf{e_m e_m}^T/m$. This T_m is called the column centralizer. For example, if **x** is a column vector then

$T_m\mathbf{x} = I_m\mathbf{x} - \mathbf{e_m e_m}^T\mathbf{x}/m = \mathbf{x} - \mathbf{e_m e_m \bullet x}/m = \mathbf{x} - \bar{\mathbf{x}}e_m$

or in short $\mathbf{x} - \bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is the mean of **x**. This T_m applied on the left of A, it centralizes columns of the matrix. Similarly, if T_n is multiplied on the right of A, the AT_n mean-centers the rows of A. For example, for row vector **y**:

$\mathbf{y}T_n = \mathbf{y}\ I_n - \mathbf{y}\ \mathbf{e_n e_n}^T/n = \mathbf{y} - \mathbf{y} \bullet \mathbf{e_n}\ \mathbf{e_n}^T/n = \mathbf{y} - \bar{\mathbf{y}}\mathbf{e_n}^T$

Centralizing data simplifies computations by reducing the number of parameters to be computed. After preforming analysis on mean-centered data, data origin is translated back to the centroid. This is a standard technique used for computational simplification and for visualization in graphics [13], [14].

A.1 Linear Least Square Approximation

There are two ways to compute linear least squares approximation. It depends on the concept of approximation. One way is to find line at shortest distance perpendicular to the desired line. Another way is to minimize the distance along a vertical coordinate axis, e.g, y -axis. Both methods accomplish specific tasks and the corresponding approximation errors are different. A hybrid approach is doubly robust estimator at increased cost and reduced error, propensity metric shows a remarkable improvement. The hybrid technique generalizes the line to polygonal line that effectively improves the pointwise

ISSN 2395-8618

accuracy without the risk of overfitting data. This method is qualitative for measuring the accuracy of points are closer to the approximation rather than the quantitative distance error. We focus on more qualitative accuracy in data approximation rather than absolute error, that may be attributed to anomalies/outliers.

A.2 Conventional Ordinary Linear Least Square Approximation (OLS)

For n× (m+1) input data, the rows are of the matrix are composite **x** (**x** is m-vector) and y coordinates of data points, that is, m-vector **x** elements are attributes, and scalar y is an associated value. For notation, $\mathbf{x_i}$ refers to the ith row, x_{ij}, refers to the element in the i-th row, j-th column. There is short-cut notation x_{*k} represents a column of the k-th element of all rows, and $\overline{x_k}$ is the mean of the column of kth elements x_{*k} of all row vectors. For clarity, note that $\mathbf{x_k}$ refers to the kth row/ attribute vector of vectors, whereas $\overline{x_k}$ represents mean of the k-th attribute. We want to find a linear least square approximation hyperplane. First, for hyperplane

y = a + $\mathbf{b}^{\mathrm{T}}$ **x** = a + $\sum_{k=1,m} b_k x_k$, we need to calculate *parameters* a and **b** that minimize the function

f(a, **b**)=$\sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x_i})^2$.

That leads to two equations

$\frac{\partial f(a,\mathbf{b})}{\partial a} = \sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x}_i) = 0$ (1)

and

$\frac{\partial f(a,\mathbf{b})}{\partial b_k} = \sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x}_i)x_{ik} = 0$ (2)

Let $\overline{x_k} = \frac{\sum_{i=1,n} x_{ik}}{n}$, $\bar{y} = \frac{\sum_{i=1,n} y_i}{n}$, $\overline{x_k y} = \frac{\sum_{i=1,n} x_{ik} y_i}{n}$, $\overline{x_k{}^2} = \frac{\sum_{i=1,n} x_{ik}{}^2}{n}$,

the first equation (1) becomes

$\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ which implies that the regression plane passes through the centroid ($\bar{\mathbf{x}}$, $\bar{y}$).

The second equation (2) implies that for k=1,m these m equations are

$\sum_{i=1,n}(y_i - a - \sum_{j=1,m} b_j x_{ij})x_{ik} = 0$

$\sum_{i=1,n}(x_{ik}y_i - a\mathbf{x}_{ik} - \sum_{j=1,m} b_j x_{ij}\, x_{ik}) = 0$

which means

$\overline{x_k y} = a\overline{x_k} + \sum_{j=1,m} b_j \overline{x_j x_k}$

$\overline{x_k y} = a\overline{x_k} + \boldsymbol{b}^T\, \overline{\mathbf{x}\, x_k}$.

Now $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ and

$\overline{x_k y} = a\overline{x_k} + \sum_{j=1,m} \mathbf{b}_j \overline{x_j \mathbf{x}_k}$ for k=1,m

That is

$\sum_{j=1,m} b_j \overline{x_j x_k} = \overline{x_k y} - a\overline{x_k}$ (1)

Also $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ can be expanded as

$\bar{y} = a + \sum_{j=1,m} b_j \overline{x_j}$

Multiply by $\overline{x_k}$

$\overline{x_k}\ \bar{y} = a\overline{x_k} + \sum_{j=1,m} \overline{x_k} b_j \overline{x_j}$

Or

$\sum_{j=1,m} \overline{x_k} b_j \overline{x_j} = \overline{x_k}\ \bar{y} - a\overline{x_k}$ (2)

Subtracting (2) from (1)

We get $\sum_{j=1,m} b_j \overline{x_j x_k} - b_j \overline{x_j}\ \overline{x_k} = \overline{x_k y} - \overline{x_k}\ \bar{y}$

or $\sum_{j=1,m}(\overline{x_k \boldsymbol{x_j}} - \overline{x_k}\ \overline{x_j}) b_j = \overline{x_k y} - \overline{x_k}\ \bar{y}$

in m unknowns b_j.

These equations can be rewritten in terms of symmetric coefficient matrix is

$[\ \overline{x_i \boldsymbol{x_j}} - \overline{x_i}\ \overline{x_j}\ ]\ \mathbf{b} = [\ \overline{x_i y} - \overline{x_i}\ \bar{y}\ ]$

This gives **b**. However since $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$, once **b** is known, the offset/bias term a can be efficiently computed from

$a = \bar{y} - \boldsymbol{b}^T \bar{\mathbf{x}}$

In the special case, m=1, then k=1, **x** has only one component say x_1 = x

We can solve for a and b to yield [15]

$b = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$ and $a = \frac{\overline{x^2}\bar{y} - \bar{x}\,\overline{xy}}{\overline{x^2} - \bar{x}^2}$

It may be noted that for mean-centered data, $\bar{x} = 0$, $\bar{y} = 0$, it results in a=0.

Briefly, for input data n×2 matrix, columns are x, y coordinates of data points, we find a linear least square approximation line. Before exploiting any approximation, it is assumed that data is accurate, else prediction will also be inaccurate. For linear approximation line y = a + bx, we need to calculate *two parameters, also called regression coefficients*, a and b for minimizing of

f(a, b)=$\sum_{i=1,n}(y_i - a - bx_i)^2$.

Using calculus criteria based on derivatives, it leads to two equations

$\frac{\partial f(a,b)}{\partial a} = \sum_{i=1,n}(y_i - a - bx_i) = 0$

$\bar{y} - a - b\,\bar{x} = 0$ (1)

and

$\frac{\partial f(a,b)}{\partial b} = \sum_{i=1,n}(y_i - a - bx_i)x_i = 0$

$\overline{xy} - a\bar{x} - b\,\overline{x^2} = 0$ (2)

The first equation (1) becomes $\bar{y} = a + b\,\bar{x}$, which implies that the regression line, y = a + bx, passes through the centroid ($\bar{x}$, $\bar{y}$). The two equations are

$\bar{y} = a + b\,\bar{x}$ and $\overline{xy} = a\bar{x} + b\,\overline{x^2}$

can be solved for a and b to yield

$b = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$ and $a = \frac{\overline{x^2}\bar{y} - \overline{xy}\,\bar{x}}{\overline{x^2} - \bar{x}^2}$

However since $\bar{y} = a + b\,\bar{x}$, once b is known, the offset/bias term a can be efficiently computed from $a = \bar{y} - b\,\bar{x}$.

A.3 Mean-Centered data formulation

Continuing in R^2, mean-centering allows us to consider regression line through the origin because centroid is translated to the origin. The bias term *a* becomes zero automatically and the data becomes unbiased. To take advantage of standardization, The OLS can be simplified for mean-centered data, we need to compute *only one* regression coefficient b for minimizing f(b)=1/n$\sum_{i=1,n}(y_i - bx_i)^2$

ISSN 2395-8618

or

$$f(b) = 1/n \sum_{i=1,n}(y_i - bx_i)^2$$
$$= 1/n \sum_{i=1,n} (y_i^2 - 2by_ix_i + b^2 x_i^2)$$
$$= \overline{y^2} - 2b\overline{xy} + b^2\,\overline{x^2}$$

That is

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$

For calculus based critical values, see [16]. *Calculus* based critical value criteria requires that f'(b) = 0. This leads to $-2\overline{xy} + \overline{x^2}\,2b = 0$ or

$$b = \frac{\overline{xy}}{\overline{x^2}}$$

So, for mean-centered data, OLS line is

$$y = bx, \text{ with } b = \frac{\overline{xy}}{\overline{x^2}}$$

which is a simpler expression than the raw data computations. Since f''(b) = $2\,\overline{x^2}$ is positive, the critical value is minimum.

However, if we want to go to the original frame, original reference point, we may translate the origin back to the centroid, then line translate into original coordinates

$$y - \bar{y} = b(x - \bar{x}) \text{ or } y = \bar{y} - b\bar{x} + b\,x$$

that is

$$y = a + b\,x \text{ where } a = \bar{y} - b\,\bar{x}$$

In this case, *only* b is to be computed, a is an automatic byproduct.

This gives a line through (0,a) and along the direction $\frac{(1,b)}{\sqrt{(1+b^2)}}$

Non-Calculus (algebraic) approach proceeds as follows.

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$

Since it is a quadratic (convex) function and $\overline{x^2} \geq 0$, Figure 1, there is *only one* minima. This expression simplifies to

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$
$$= \overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 + \frac{\overline{x^2}\,\overline{y^2} - \overline{xy}^2}{\overline{x^2}}$$

Since $\overline{x^2}\,\overline{y^2} - \overline{xy}^2 \geq 0$, f(b) is min when $b = \frac{\overline{xy}}{\overline{x^2}}$. This is what we got above using calculus.

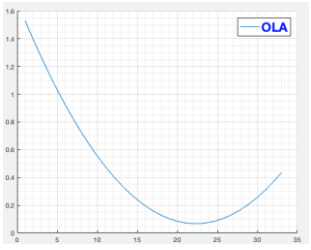


Figure 1. The convex function f(b) has only one global minima giving the slope b for OLS line.

In essence, this is a common sense three step approach to find the OLS line. The three steps are, (1) mean-center the data, translate the centroid $(\bar{x}, \bar{y})$ to the origin (0,0), (2) calculate the direction of least square error approximating line through the origin, (3) translate data origin back to centroid $(\bar{x}, \bar{y}\ )$ for original frame of reference. The computations using mean-centered data are simpler. In, Figure 2, Cyan dots are the raw training data, solid red line is the approximation line, and red dotted lines are errors between the training data and corresponding predicted approximations. In Figure 3, there the black dotted lines are normal(perpendicular, orthogonal) to the regression line where as red dotted lines are vertical, along the y-axis direction. Clearly the normal lines are shorter than vertical line.

We will explore and exploit further whether there are some other lines whose normal distance error is even smaller than this line error. That leads us to next section.

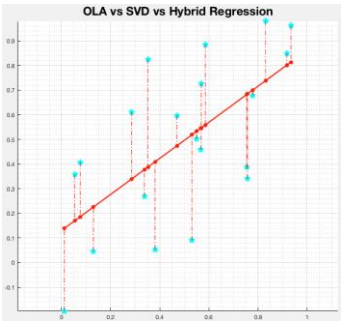


Figure 2. Data points, regression line, approximation errors

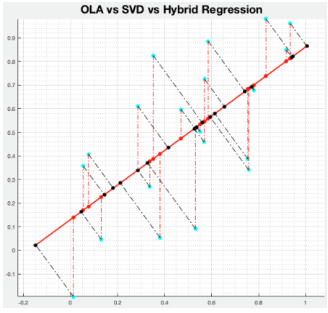


Figure 3. The red vertical dotted lines are error from OLS line along y-axis, the black orthogonal dotted lines are error from OLS line along the normal. Normal distance error is smaller than vertical distance error.

III. NORMAL LINEAR LEAST SQUARE APPROXIMATION (NLS)

NLS has not been used in social sciences because of its complexity [17]. The ordinary linear approximation (OLS) line is not as close to the data points as expected because distances/errors are measured along the y-axis. If distances are measured along the normal (perpendicular) to the approximation line, then line is more representative of data. The normal (perpendicular, orthogonal) distance problem is formulated below. SVD is a method that accomplishes the same goals, without resorting to calculus of extrema computations.

For the reasons stated above, we assume that the data (x_i, y_i), i=1, 2, …, n is mean-centered, otherwise we can use centralizer transformation to mean-center the data. The problem becomes that of finding the value of *only one parameter b* that minimizes f(b) where

ISSN 2395-8618

$$f(b) = 1/n \sum_{i=1,n} \left(\frac{y_i - bx_i}{\sqrt{1+b^2}}\right)^2 \quad \text{or}$$

$$f(b) = 1/n \sum_{i=1,n} \frac{(y_i{}^2 + b^2 x_i{}^2 - 2bx_i y_i)}{1+b^2}$$

$$= \frac{\overline{y^2} + b^2\,\overline{x^2} - 2b\overline{xy}}{1+b^2} \qquad (1)$$

Thus, for local minima of $f(b) = \frac{\overline{y^2} + b^2\,\overline{x^2} - 2b\overline{xy}}{1+b^2}$

$$f(b) = \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \overline{y^2}}{1+b^2} = \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \frac{\overline{xy}^2}{\overline{x^2}} - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \frac{\overline{xy}^2}{\overline{x^2}} - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{\overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{\overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 + \frac{\overline{x^2}\,\overline{y^2} - \overline{xy}^2}{\overline{x^2}}}{1+b^2}$$

Note that $\overline{x^2}\,\overline{y^2} - \overline{xy}^2$ always ≥ 0. It is equivalent to standard result $|\mathbf{x}\bullet\mathbf{y}| \leq |\mathbf{x}||\mathbf{y}|$ which can be quickly derived from triangle inequality or geometric definition of dot product.

We saw that in the unnormalized case, f(b) is minimum when

$$b - \frac{\overline{xy}}{\overline{x^2}} = 0 \text{ or } b = \frac{\overline{xy}}{\overline{x^2}}$$

This is *not true* in this case, see Figure 4. For OLS, f(b) is quadratic, convex and has only one extreme/minima blue curve. For NLS, f(b) is not convex, not quadratic red curve. It has two extrema, one maxima and one minima. In both OLS and NLS cases, the minima are close to each other, but not identical.

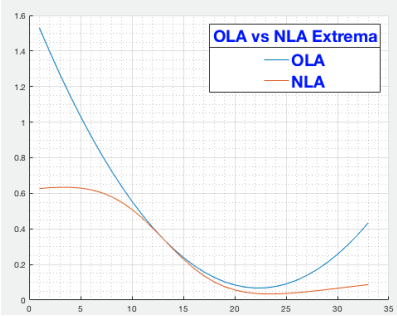


Figure 4 f(b) is convex for OLS case, f(b) is not convex for NLS case.

For NLS, f(b) is never negative. As b approaches zero, f(b) becomes $\overline{y^2}$ and as b approaches infinity, f(b) becomes $\overline{x^2}$.

To calculate the minimum, setting the first derivative of f(b) w.r.t b to zero, f'(b)=0, we get quadratic

$$\overline{xy}\,b^2 + \left(\overline{x^2} - \overline{y^2}\right) b - \overline{xy} = 0 \qquad (2)$$

Since it is a quadratic, it has two critical values, b_1, b_2

$$b = \frac{-(\overline{x^2} - \overline{y^2}) \pm \sqrt{(\overline{x^2} - \overline{y^2})^2 + 4\,\overline{xy}^2}}{2\,\overline{xy}} \qquad (3)$$

f(b) can't have both local minima, see Figure 4. If f"(b_1)>0, the b_1 is a local minima else f"(b_2)>0, then b_2 is a local minima.

However, from the Figure 4, it is clear the minimum occurs at larger of b1 and b2.

Once $b = \frac{-(\overline{x^2} - \overline{y^2}) + \sqrt{(\overline{x^2} - \overline{y^2})^2 + 4\,\overline{xy}^2}}{2\,\overline{xy}}$ is computed, we have a line through the origin (0,0) along the *direction* $\frac{(1,b)}{\sqrt{(1+b^2)}}$

The normal least square line (NLS) is shown in Figure 5. This is not the same as OLS regression line seen in Figures 2 and 3.

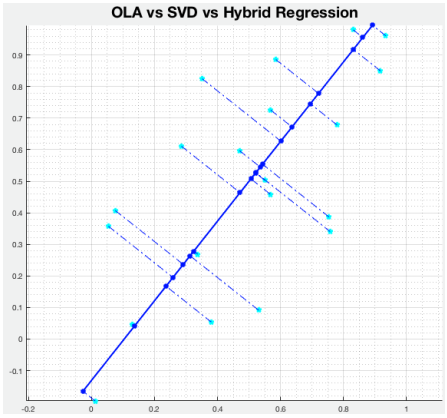


Figure 5. Cyan dots are the data points blue line is NLS line. Blue dots are the approximation, Blue dotted lines are normal errors from NLS line.

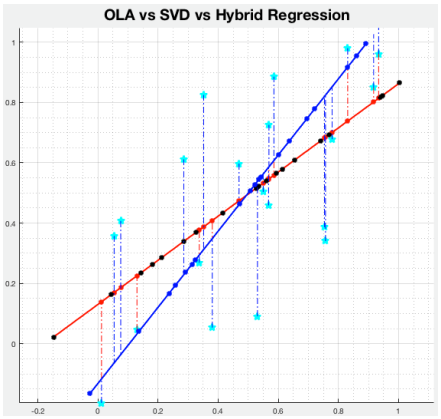


Figure 6 Red line is OLS, Blue line is NLS. Red dotted lines and Blue dotted lines are vertical errors form the Cyan data points. NLS vertical error from Blue line is *more* than OLS error from red line.

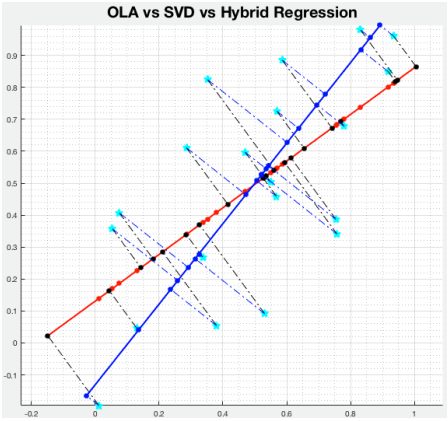


Figure 7 Red line is OLS, Blue line is NLS. Red dotted lines and Blue dotted lines are orthogonal errors form the Cyan data points. NLS normal error from Blue line is *less* than OLS error from red line.

ISSN 2395-8618

Further, the approximation error in both cases (OLS and NLS) is minimum depending on how the error is measured. Visual inspection shows that *majority* of the cyan dots are *closer* to blue line dots than the cyan dots to red line dots, see Figure 6, Figure 7. This visualization justifies, to some extent, to prefer NLS over OLS. Note when overall vertical error is larger for NLS line where as overall normal error is larger for OLS line. This confusion needs some resolution. We will give formal justification later in section V. Since NLS is based on calculus, its derivative is complex, the second derivative is quite complex, we explore an easier implementation of this idea by means of exploiting singular value decomposition (SVD).

In some applications, error is measured along vertical line y-direction, while in some application error is measured along the normal to the Least Squares Approximation line. When there is no other algorithm to compare, a technical indicator is used to measure to quality of approximation. Bollinger technique uses band of 1,2, 3 standard deviation bands to test the goodness of the model,

The third type of error is never used in numerical least square approximation. The propensity metric has been used for non-numerical data. Our goal is to blend the two algorithms and the corresponding measure of error into uniform error metric and compare the performance of two methods. The optimal approximation is better represented by propensity metric, no matter which method of error computation is used.

IV SINGULAR VALUE DECOMPOSITION (SVD)

Today, singular value decomposition is used in many theoretical and applied fields: computer science and engineering, psychology and sociology, atmospheric science and astronomy, health and medicine etc. [18], [19], [16], [20], [21]. It is also extremely useful in machine learning and in both descriptive and predictive statistics. There is no unique basis function for R^n. The goal is to determine a suitable basis function so that A can be expressed in response to the application. The normal least square approximation (NLS) hyperplane can also be obtained directly by using linear algebra singular value decomposition (SVD). Before we discuss the connection between NLS and SVD, we may note that SVD is important on its own right due to applications in various areas. For the sake of completeness, we give brief description of SVD.

Singular Value Decomposition (SVD) is a matrix factorization technique generalizing eigen-decomposition and principal component analysis. Every positive semi-definite real matrix can be decomposed into three matrix factors: left singular vectors matrix, right singular vectors matrix and a diagonal matrix of singular values in descending order on main diagonal. The goal is not to recreate the matrix, but to create the *best linear least square approximation* [22], [23]. There are various advantages of SVD. First, 150 years old *Principal Component Analysis* (PCA) is a specialization of eigen-decomposition to symmetric matrices with orthogonal eigenvectors such that $A = VDV^{-1} = VDV^T$. In case, A is not a square data matrix, PCA does not apply. However, A^TA is a symmetric square positive semi-definite matrix, then $A^TA = VDV^T$, [24], [25], [26]. Besides other benefits of this factorization, we are interested in *direction vector* only for least square approximation. The columns of V are eigenvectors of A^TA corresponding to eigenvalues arranged in descending order. Since eigenvectors correspond to directions of approximation lines, we show that direction vector of NLS corresponds to first eigenvector of SVD [27], [28], [16]. The following table, describes the distinction between eigen decomposition (ED), PCA and SVD. Briefly, for eigen decomposition of A, U is the matrix of eigenvectors of A, D is diagonal matrix of eigenvalues of A, conveniently eigenpairs are arranged on descending order of eigenvalues.

$$\text{ED} \rightarrow A = UDU^{-1}$$

For PCA and SVD, U and V are matrices of eigenvectors of symmetric matrices AA^T and A^TA, S is the matrix of singular values of positive semi-definite matrix and D is the matrix of eigenvalues of A such that

$$\text{PCA} \rightarrow A = UDU^T$$

and

$$\text{SVD} \rightarrow A = USV^T$$

The following Table 1 shows a summary of different aspects to express A in terms of ED, PCA, SVD using eigenvectors as basis vectors. Examples show the case where the matrix is (1) symmetric and positive semidefinite,(2) matrix is symmetric, but not positive semi-definite, (3) matrix is not symmetric, but is positive semi-definite, and (4) where matrix is not symmetric, and no positive semidefinite.

A. Connection between NLS and SVD

For simplicity, A is n×2, of data points in the xy-plane. To minimize the error between observed **P** and estimated direction **v**. Since $\mathbf{P} = \mathbf{P}\bullet\mathbf{v}\ \mathbf{v} + (\mathbf{v}\text{x}\mathbf{P})\text{x}\mathbf{v}$, minimizing $|(\mathbf{v}\text{x}\mathbf{P})\text{x}\mathbf{v}|$ means maximizing the distance $|\mathbf{P}\bullet\mathbf{v}\ \mathbf{v}|$ or $|\mathbf{P}\bullet\mathbf{v}|$ because **v** is a unit vector [16].

We derive the direction **v** so that sum of squares of distances from training data points to predicted direction vector **v** is least. Note, **v** passes through the origin because the data is mean-centered. Since data is mean-centered, the approximation line passes through the origin. By default, vectors **P** are column vectors in linear algebra, thus rows of A are position vectors $[x,y] = \mathbf{P}^\mathbf{T}$. As seen above, the vector **P** can be written as the sum of a vector along unit vector **v** and a unit vector **w** orthogonal to **v**, that is, using vector notation $\mathbf{P} = \mathbf{P}\bullet\mathbf{v}\ \mathbf{v} + (\mathbf{P} - \mathbf{P}\bullet\mathbf{v}\ \mathbf{v}) = v\ \mathbf{v} + w\ \mathbf{w}$. This means that minimizing the distance w amounts to maximizing v. We are to maximize over all data points $\mathbf{P}_i$. The problem becomes that of maximizing

$$\sum_{i=1,n} |\mathbf{P}_i\bullet\mathbf{v}|^2$$

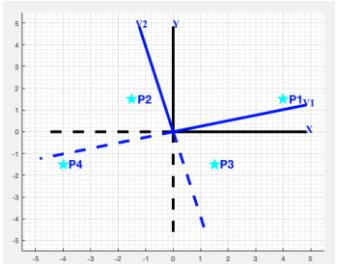
for *all* P_i for *some* vector **v** to be determined. Now

$$\begin{aligned}\sum_{i=1,n} |\mathbf{P}_i\bullet\mathbf{v}|^2 &= \sum_{i=1,n} \mathbf{P}_i\bullet\mathbf{v}\ \mathbf{P}i\bullet\mathbf{v} = \sum_{i=1,n} \mathbf{v}\bullet\mathbf{P}_i\ \mathbf{P}_i\bullet\mathbf{v} \\ &= \sum_{i=1,n} \mathbf{v}^\mathbf{T}\mathbf{P}_i\ \mathbf{P}_i^\mathbf{T}\mathbf{v} = \mathbf{v}^\mathbf{T}\left(\sum_{i=1,n} \mathbf{P}_i\ \mathbf{P}_i^\mathbf{T}\right)\mathbf{v} \\ &= \mathbf{v}^\mathbf{T}\ (A^TA)\mathbf{v}.\end{aligned}$$

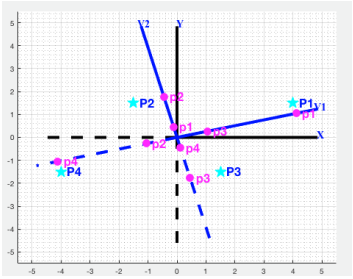
ISSN 2395-8618

TABLE 1 MATRIX TYPES AND THEIR ED, PCA, SVD

Matrix A	A-Symmetric	A-Pos Semi- Definite	ED matches A	PC matches A	SVD matches A
$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	√	√	√	√	√
$\begin{bmatrix}0 & 0\\0 & -1\end{bmatrix}$	√	x	√	√	x
$\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$	x	√	x	x	√
$\begin{bmatrix}1 & 0\\1 & -1\end{bmatrix}$	x	x	√	x	x



(a)



(b)

Figure 8. (a) Data points, standard x-,y-axes, v1-,v2- eignevectors axes, (b) Projection of Data points on v1-,v2- eignevector axes, Data points are closer to eigenvectors than the standard axes.

This means that $\sum_{i=1,n} |\mathbf{P}i \bullet \mathbf{v}|^2$ is maximum if **v** is an eigenvector of A^TA and corresponds to largest eigenvalue of A^TA. Similarly, all the other eigenvectors can be obtained incrementally one at a time, constraining each vector orthogonal to the previous eigenvectors. Thus, SVD is computed iteratively in descending order of eigenvalues and corresponding eigenvectors orthogonal to the previously computed eigenvectors. From this analysis, it is clear that the largest eigenvalue amounts to the largest spread of data along the corresponding eigenvector. The spread of projections of data on $\mathbf{v_1}$ is larger than that on $\mathbf{v_2}$, see Figure 8(b).

For example, $\mathbf{P^T}$'s are data points in 2D, $\mathbf{v_1}$, $\mathbf{v_2}$ are eigenvectors corresponding to largest eigenvalues of A^TA . For this consideration, the NLS requires only $\mathbf{v_1}$, the direction with largest eigenvalue, and with largest data spread.

Uniqueness of Eigenvectors. As a side remark, for the matrix, any non-zero multiple of an eigenvector is again an eigenvector. To make the eigenvectors unique, they are normalized to unit vectors. But if **u** is unit eigenvector, then – **u** is also a unit vector, see Figure A in appendix for MATLAB[30], svd computed eigenvectors [27], [28]. In the literature, it is an accepted convention to make the first non-zero component positive in the eigenvector, see Figure [see appendix]. Since eigenvectors are ordered, we use ordering to make the k-th element of k-th vector to be positive, see Figure A [see Appendix] that makes the vectors look more natural like a right- handed system. In case, the kth element is zero, then the first non-zero element is made positive. This is the approach we prefer to use [16]. Incidentally, recall that the direction vectors in OLS and NLS had first component as positive in the figure.

For example, consider the matrix $A = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$, then $AA^T = A^TA = A^2 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, PCA uses AA^T and A^TA (A, A^2 are symmetric; A^2 is a positive semi-definite matrix) for computing the eigen-pairs. In this example, except for signs, the eigenvalues of A are square roots of the eigenvalues of A^2 that are 1 and 1, the corresponding eigenvectors are eigenvectors of A^2 are $\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}$. In particular, the eigenvector of A corresponding to eigenvalue -1 and eigenvector of A^2 pertaining the eigenvalue 1 are identical. Matlab svd function does not reconstruct eigenvalues of A accurately. The vectors in V are superficially adjusted to match A. In our algorithm, we include the proper signs. Matlab R2017b computes SVD resurrects A as

$$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}.$$

This is inaccurate as $\begin{bmatrix}0\\-1\end{bmatrix}$ is not an eigenvector of A or A^2. Also, it may be noted that $\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}$ is not the transpose or inverse of $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$.

Here we used the proper sign for *square root of 1 to -1*, because -1 is eigenvalue of A. Consistent with the definition of SVD with correct sign of eigenvalue [28], [15], the correct eigen-decomposition $A = VDV^{-1} = VDV^T$ is

$$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}.$$

Recall, Singular Value Decomposition (SVD) is a generalization of PCA to include (1) non-square rectangular

ISSN 2395-8618

and (2) positive semi-definite matrices [21]. However, PCA and SVD are equivalent for symmetric positive semi-definite matrices. SVD uses covariance matrices AA^T and A^TA to determine two orthogonal matrices U, V of eigenvectors and a diagonal matrix S for singular values such that the eigenvectors in U, (and V) are (1) pairwise orthogonal, (2) normalized to unit vectors and (3) arranged in the descending order of singular values. A singular value of A is square root of the eigenvalue of A^TA and AA^T. Then SVD decomposes A into three factors U, V and S such that $A = USV^T$. By dropping the least significant singular values and corresponding singular vectors, best approximation of data matrix can be reconstructed, quantitative error can be estimated simply by using the discarded eigenvalues. The examples where A is not both symmetric and positive semi-definite are shown in the Table 1 to confirm SVD and PCA are not equivalent in general. In our work, A is symmetric positive semi-definite, consequently AA^T and A^TA turn out to be symmetric positive semi-definite [22], [10],[127],[26].

Example. To accommodate both PCA and SVD, we generalize the previous example matrix to symmetric, positive semi-definite (PSD) matrix $A=\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}$. Now $AA^T=A^TA=A^2=\begin{bmatrix}1 & 0\\ 0 & 4\end{bmatrix}$ has eigen values 1,4. Thus the singular values for A are 2,1; so, $D=S=\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$ which is same as S obtained from SVD. Thus, for PCA/SVD of A, the eigenvectors of AA^T, A^TA form orthogonal matrices $U = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, $V=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, and singular values become the diagonal entries of $S = \begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$. Now PCA as well as SVD factorization is

$$\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

which implies that $A = USV^T = VSV^T$. $= USU^T$.

Summarizing this discussion in Table 2, we see that several possibilities exist for an arbitrary matrix A. For example, in Table 2, there are some cases where A is (1) symmetric and (1.1) has PCA SVD decomposition equivalent on PSD matrix (1.2) has PCA, but SVD does not exist as A is not SPD matrix, and (2) not symmetric and (2.1) PCA does not exist because A is not square matrix, SVD decomposition exists on PSD matrix (2.2) has no PCA, no valid natural SVD decomposition for non-square non-PSD matrix. Also refer to Table 1 for square matrices.

TABLE 2. FOUR EXHAUSTIVE CASES

Data Matrix	Positive Semi-Definite	Not Positive Semi-Definite
Symmetric	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ PCA, SVD	$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$ PCA, ~~SVD~~
Not Symmetric	$\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}$ ~~PCA~~, SVD	$\begin{bmatrix}-1 & 0\\ 0 & -1\\ 0 & 0\end{bmatrix}$ ~~PCA~~, ~~SVD~~

V. HYBRID GREEDY ALGORITHM DESIGN

The idea of hybrid algorithm is not just to amalgamate these two algorithms in such a way that the new algorithm accuracy supersedes the accuracy of the base algorithm, but to extract the best features of both and improve on them with propensity metric and double SVD. The error analysis is metrically and cognitively appealing to humans. Interval of error, also known as Bollinger band of uncertainty, quantifies the range of uncertainty in a value and propensity score is a frequency metric used for comparing them pairwise to determine the best algorithm. Hybrid algorithm uses a doubly robust balancing method. it is responsive to treatment for data dealing with patient treatments. Hybrid algorithm is designed to overcome the limitations of traditional parametric methods.

We design a hybrid greedy algorithm leveraging the best of OLS and NLS/SVD approximation lines in two ways: non-parametric polygonal possibly overfitting; and parametric line in general. For each observed point, (x_0,y_0), we have seen in Figure 6 and Figure 7 that there is a corresponding predicted point (x_R, y_R) on regression line and a predicted point (x_S, y_S) on SVD line. If (x_0,y_0) is an observed value, (x_R, y_R) is predicted point value corresponding to the OLS line $y = a + bx$. The vertical distance is along y direction. The distance between (x_0,y_0) and (x_R,y_R) is the y-distance, the OLS regression error $e_R = |y_0-y_R|$. For normal distance from NLS or SVD approximation line, it is along perpendicular to the line, it turns out that $x_S \neq x_0$ in (x_S, y_S) , the distance between (x_0,y_0) and (x_S,y_S) is Euclidian normal distance $e_S = \sqrt{(x_0 - x_S)^2 + (y_0 - y_S)^2}$.

It is clear from Figure 6 and Figure 7 that for some points in observed data, $e_R < e_S$ while for some other points $e_S < e_R$. In each method, the total error E is sum of squares of pointwise distances (errors) for all data points, question arises which one (E_R for OLS and E_S for SVD) is acceptable due to the dual nature on error computation. There is no denying the fact if vertical distances are used for *both* lines, then $E_R<E_S$ and if normal distances are used for *both* lines, then $E_S < E_R$. Then how does the user determine which one preferable to use: OLS or NLS/SVD? For greedy algorithm, define the approximation point (x_H, y_H) to be that point which is closer to the observed point (x_0,y_0) in both ways. Euclidean distance is used to measure closeness. For each input, we will determine approximate line that represents the input data no matter how the error is computed, see Figure 10 for green color dots, these are closer to cyan dots than red line dots or blue line dots. Instead of measuring the quantitative distance we define a qualitative metric that is more useful in visualization and is cognitively acceptable. Non-parametric algorithm uses to regression coefficients of OLS and NLS, whereas the parametric version computes its own regression coefficients.

ISSN 2395-8618

A. Non-Parametric Hybrid Greedy Algorithm

Algorithm A:

Input: array of x and y mean-centered data values

Output: hybrid greedy approximation points (x_H,y_H),where $(x_R\ y_R)$ is on OLS, $(x_S\ y_S)$ is on SVD line

1. Calculate regression coefficients a and b for OLS regression from observed x,y
   Calculate predicted values by linear regression $y_R = a+bx$
   Calculate approximation error E_R
   Test Goodness of the regression line
2. Calculate A=[x ,y], x, y are columns of matrix A.
   Calculate SVD [U S V] = svd(A)
   Use first column of V to get b. a is automatic
   Calculate x_S,y_S of projected points $[x_S,y_S]$ on column vectors of V that is AVV'
   Calculate approximation error E_S
   Test Goodness of the NLS line
   Compare error E_R and E_S
3. Calculate greedy hybrid x_H, y_H using a variation of relaxation method
   for all point pairs $[x_R, y_R]$, $[x_S, y_S]$
   if d($[x_S, y_S]$, $[x_0, y_0]$)<= d((x_R, y_R), (x_0, y_0))
   $(x_H, y_H) = (x_S, y_S)$;

else

   $(x_H, y_H) = (x_R, y_R)$;

   end

end

   Calculate error E_H from pointwise e_H
   Test Goodness of the hybrid line
   Compare error E_S, E_R, E_H
   Compare by propensity values
4. x_H, y_H are arrays of predicted coordinates on hybrid polygonal line, cognistically appealing and lower metric error.

This algorithm gives non-parametric polygonal approximation and possibly overfitting. The next algorithm parametrizes it by using SVD on polygonal approximation, see Table 2. The hybrid representation (x_H,y_H) is closer to the input training data (x,y) that the OLS approximation (x_R,y_R) points and NLS approximation (x_S,y_S) points. Note that in practice we do not need to store the polygonal approximation values, it is more efficient to retain the regression coefficients of OLS and NLS/SVD for real time calculations.

B. Parametric Hybrid Algorithm

This algorithm is of theoretical interest and for visualization, Algorithm A is sufficient for practical use. The non-parametric *polygonal* approximation algorithm gives insight for improving the accuracy, Figure 11. It has two shortcomings it does conserve space, and it is subject to overfitting the input training data. Here we explore double approximation to design a general algorithm which conserves space as well as it is parametric, see Figure 11.

Algorithm B

Input: array of x and y mean-centered data values

Output: hybrid approximation line parameters for points (x_H,y_H),where $(x_R\ y_R)$ is on OLS, $(x_S\ y_S)$ is on SVD line

- As in algorithm A, polygonal greedy approximation is (x_H,y_H)
- Use SVD to fit computed points (x_H,y_H) with SVD algorithm to derive parameters for the direction of the line
- Use direction of this double SVD line to compute approximation (x_D, y_D)
- Test Goodness of the based on this double NLS line
- Compare E_R,E_S, E_H, E_D, and propensity metric values

Now almost all observed points are closer to greedy line than OLS and NLS/SVD approximation lines. It satisfies the general parametric and space conservation requirements, see Table2.

Note over the entire data set, *red dots have smallest error* from cyan dots when distances are measured along y, while *blue dots have smallest error* from cyan dots when distances are measured along the normal to the line, see Figure 9. Each green dot is at a smaller of the two distances from cyan dot, interestingly, it *does not mean* that green dots have *overall* smaller error than the two, in fact it will be bigger than each. The green dots can be connected by a polygonal line see Figure 10 or an SVD straight line approximation, see Figure 7. We have seen that NLS is better than OLS. We may use SVD to approximate data (x_H,y_H) to (x_D,y_D) line, see Figure 11.

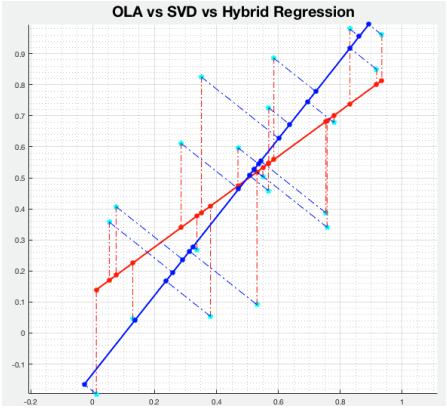


Figure 9. Cyan dots are data points, Red line is OLS line, Blue line is NLS/SVD line, Green dots are hybrid approximation dots

ISSN 2395-8618

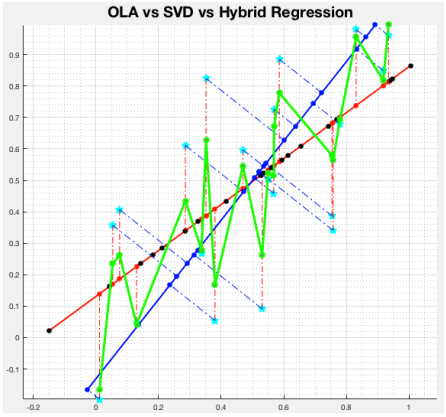


Figure 10 Non-Parametric polygonal Hybrid data points, Cyan dots are points which are closer to green dots than red or blue dots. Hybrid polygonal line, green polygonal line connects the green hybrid points (x_H,y_H).

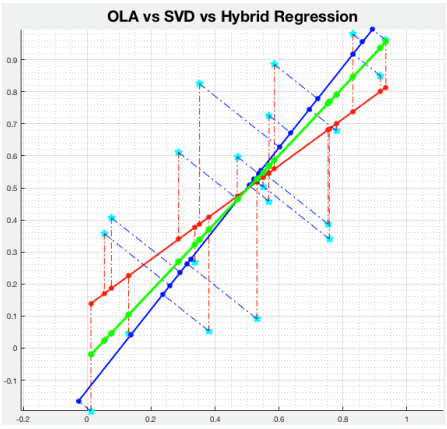


Figure 11 Non-Parametric polygonal line Green dots in Figure 10 are not shown here for clarity. SVD line is created to corresponding green points into the green Hybrid parametric line.

C. Precision and Propensity

We have seen three ways to process data. OLS is best when error is measured along y-axis. SVD is best when error is measured long normal is measured. Propensity is best when the frequency of nearness is used. The question arises which one is preferable. The propensity method is cognitively and visually preferable. The linear least square approximation error is *quantitative* measure. The precision and propensity are a *qualitative* measure of accuracy [10], [31], [11]. Quantitative error is a function of the location of data points, propensity depends on percentage of data points for pointwise binary outcome from comparing error due to a pair of methods. This is similar to precision metric used in data mining community confusion matrix. For percentage of data truly closer to OLS, SVD lines, Hybrid line pairwise, see Table 2 and Table 5. From Figure 10, it is clear that green construction is preferable, but the quantitative error comparison is inconclusive. However, we use propensity metric to determine the level of accuracy that hybrid line has as compared to OLS and SVD. When errors are measured in the respective methods, we can calculate the propensity value for one line relative to the other line to conclude the preference irrespective of which method is used to calculate errors. It is determined that overall SVD/NLS approximation is better approximation than OLS, see Figure 10. Similarly, propensity metric shows, that hybrid line is preferable to both OLS and SVD lines, see Figure 11, Table 2. Table 5.

D. Anomaly Detection and Removal

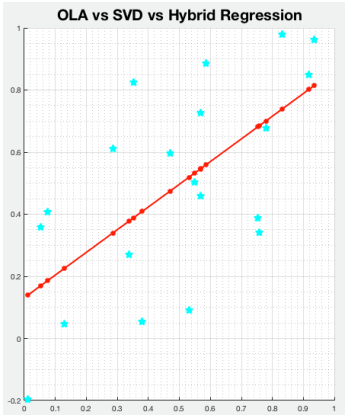
It is clear that pointwise vertical distance error, e_R, is always greater than normal distance error, e_S, from any line. Since sum of squares of errors for OLS line, E_R is smallest in the vertical distance metric, the regression error from any other line is bound to be larger than error, E_R, from OLS line. Pointwise error in OLS and NLS is inconclusive. Propensity score metric (PSM) is a qualitative measure to differentiate for better approximation line, where the distance metric fails. This will give insight to error measurement modeling to the algorithm designers. PSM can also leveraged identify the anomalies. To detect anomalies accurately, we create a confusion matrix for frequency of points within one standard deviation of both the lines, see Table 3 of confusion matrix for noise reduction using Bollinger band about OLS and NLS approximation lines.. Any point which is not within this Bollinger band about any of the two lines, is probably an anomaly (FF). Such is point is candidate for further scrutiny. After analyzing it with the hybrid line, it determined that hybrid line is a better differentiator for noisy data. After clipping suspicious points for the data, we reapplied our algorithm to ascertain that reduced data set gives better accuracy, see Table 4, and Table 5.

Table 3 Confusion Matrix for Anomaly

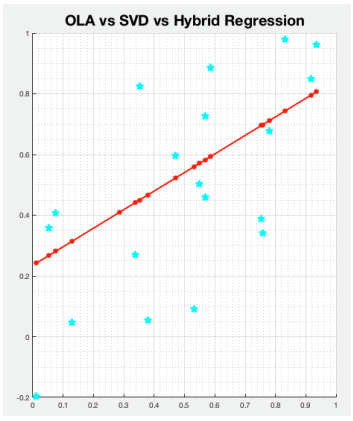
	OLS within	OLS outside
SVD within	TT	TF
SVD outside	FT	FF

Example: Noisy data, vertical distances error not realistic. In the Figure 12(b), we can see that if fifth point is noisy, it has affected the entire approximation line. In particular for the *neighboring* points, there is glaring offset. Experiments show that one outlier point can adversely affect the approximation line in the immediate neighborhood of noisy point, see Figure 12. Red line is least square regression line on raw data of 20 points. This regression line is noise sensitive, see Figure 12(a),(b). If one of data points is an outlier, it can create a large adverse effect on the outcome. Figure 12(c) shows the improvement on this shortcoming after removing noise.

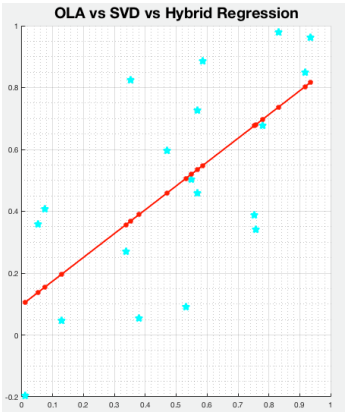
ISSN 2395-8618



(a)



(b)



(c)

Figure 12, (a) No noise, (b) Noise introduced in position 5, direction of line changes , (c) Noise removal, position 5 removed from the data, data has one less point.

Figure 12 (a) has no noise, (b) has noise in position 5, as a result the regression lines are different, (c) here noise is removed, now (a) and (c) are same, but (c) has one less points as point 5 has been removed. We do not see any major difference in the regression lines.

The goal of the new algorithm is to improve the prediction capability rather than numeric value of approximation error. Numeric error is a measure of divergence from the true value. The hybrid algorithm achieves a balance between quantitative and qualitative approximation accuracy of both OLS and NLS/SVD. We use STD-standard deviation for confidence interval about the approximation lines. If A is the set of points outside the confidence interval and B is the set of points where $e_R > e_S$, the $A \cap B$ is a candidate set of anomalies.

Table 4 Comparison of Algorithms

	OLA	SVD	Hybrid
Approximation Line Directi	[0.81, 0.59]	[0.62, 0.78]	[0.68, 0.73]
Approximation Error	7.06%	5.09%	3.32%
Confidence in one std Interv	75.00%	100.00%	100.00%
closeness OLA vs SVD	25.00%	75.00%	
closeness OLA vs Hybrid	0.00%		100.00%
closeness SVD vs Hybrid		15.00%	85.00%

Table 5 Comparison of Algorithms 5% Noise Removal

	OLA	SVD	Hybrid
Approximation Line Directi	[0.79, 0.62]	[0.66, 0.75]	[0.68, 0.73]
Approximation Error	6.46%	4.71%	3.32%
Confidence in one std Interv	88.89%	100.00%	100.00%
closeness OLA vs SVD	0.00%	100.00%	
closeness OLA vs Hybrid	0.00%		100.00%
closeness SVD vs Hybrid		16.70%	83.30%

E. Temporal Sensitivity

In health care environment, if the time interval for a treatment is changed, we expect to see the temporal change in response to a treatment. Using OLS, we see that there is no change in response to temporal change, that is, the computed error remains unchanged, see Figures 9-12. Figure 13 is the visual summary of quantitative and qualitative error in the methods. Using the same data set, on scaling the time interval, the NLS/SVD and Hybrid algorithms respond positively to the changes. This suggests that OLS is not suitable for such temporal applications. In the example, we also notice that as the slope of the hybrid line increase, the error decreases. Experiments confirm that the slope of 45 degrees is brake-even point with maximum error. Slope below or above 45 degrees accounts for reduction in error. For comparison of the three algorithms, see Table 2, Table 3. It shows the computed direction vectors of the approximation lines, approximation error in the Euclidean distance metric, and propensity how close is training data to one algorithm vs the other formulation, see Figures 13-17.

ISSN 2395-8618

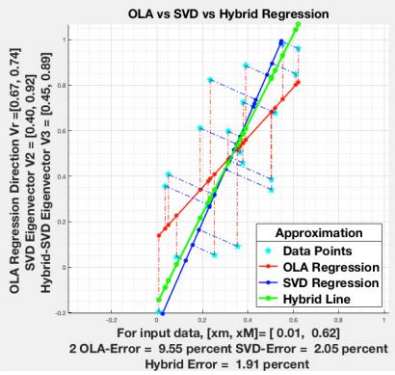


Figure 13 Relative errors one time interval [0.01,0.62]

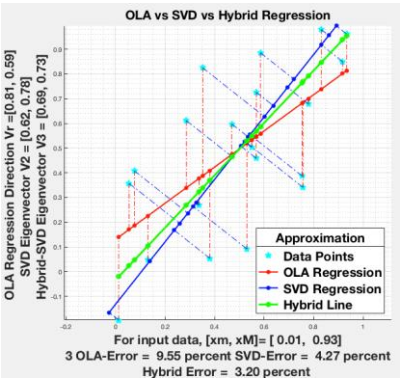


Figure 14 Relative errors on time interval [0.01,0.93]

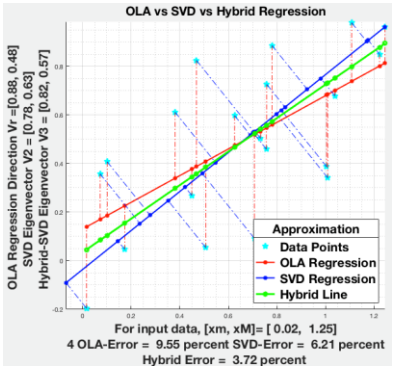


Figure 15 Relative errors on time interval [0.02,1.25]

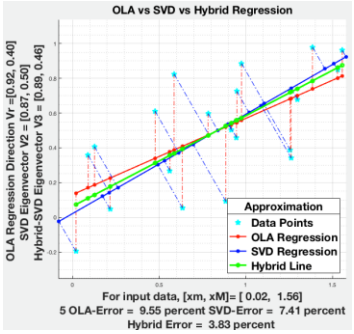


Figure 16 Relative errors on time interval [0.02,1.56]

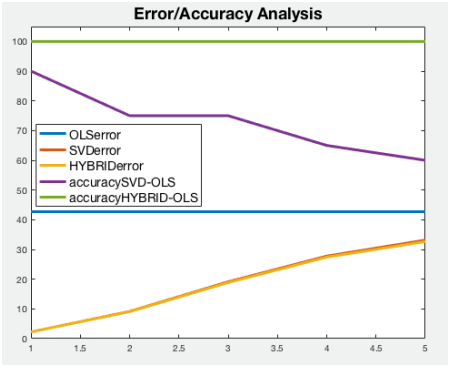


Figure 17. Green line shows percentage of Hybrid points closer to data points as compared to OLS. Purple line shows percentage of SVD points closer to data points as compared to OLS. Blue line shows percentage of error in OLS. Yellow and red (on top of each other) percentage of error in SVD and Hybrid algorithms.

VI. CONCLUSION

In the paper, we have explored several algorithms and several metrics to determine cognitively and visually acceptable criteria for least square regression. The algorithms are ordinary least squares regression (OLS), orthogonal least square regression(NLS) and Singular value decomposition (SVD). We explored these algorithms along with our hybrid algorithm. We exploited them using the quantitative and qualitative metrics. We explored 1. various ways to approximate numerical data, 2. Temporal versions of prediction, 3. how to reduce noise. Here we first removed noise by virtually using OLS and NLS. The hybrid data is then approximated by leveraging NLS/SVD, double approximation. It is determined that hybrid algorithm outperforms the other algorithms when applied and compared pairwise. This will give insight for error measurement modeling to the researchers. They will benefit from the hybrid linear least approximation algorithm.

OLS was found to be insensitive to temporal data spread, whereas SVD was implicitly modifying the independent (temporal) variables of the original input in pursuit of lower error. We designed a hybrid algorithm that overcomes these shortcomings and supersedes the accuracy of the existing algorithms. From the experiments, it follows that error is least for lines that are almost horizontal or vertical, the breakeven point occurs as the slope of the line becomes closer to 45 degrees. No matter what the slope is, the new hybrid regression line error is always bounded above by the error in OLS regression line. It is interesting to note that OLS remains unchanged while new regression line approximation error responds to the slope variation. We also showed how to improve svd algorithm of MATLAB[30] with correct directions of eigenvectors, a natural technique. The algorithm was implemented on MAC OS Mojave v 10.14, IntelCire i5, 8GB 1600MHZ using Matlab R1700b [30]. We have described the error measurement methods and propensity metric that is preferable for exploitation and visualization.

ISSN 2395-8618

VII. APPENDIX

A. Principal Component Analysis, and Singular Value Decomposition

This section is self-contained tutorial on PCA/SVD. The linear algebra concepts of *vector*, *transpose* of a vector, *scalar* product of a vector, Euclidean *norm*, *length* of a vector, *unit* vector, *sum* of two vectors, *dot product* of two vectors (analytical, geometric, matrix forms), *orthogonal* vectors, *Gram-Schmidt* orthogonalization, *matrix*, *square* matrix, *identity* matrix, *diagonal* matrix, *transpose* of matrix, *symmetric* matrix, *sum* of matrices, *scalar product* of a matrix, *determinant* of a matrix, *rank* of a matrix, *inverse* of a matrix, *norm* of a matrix, *orthogonal* matrix, *rotation* matrix, *rank* of a matrix, *determinant* of a matrix, *product* of matrices, *vector space*, and *basis* of a vector space, are standard terms in linear algebra. Additional terms that we use are an *eigenvector*, and an *eigenvalue*. All vectors are column vectors unless specifically stated. All matrices and vectors are real in this discussion. For details on linear algebra, reader may consult references [13, Jolliffe1995].

All the required transformations are built in the toolboxes of modern languages, Java, C++, Matlab, R, and Python. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Herein modeling tools are eigenvalues and eigenvectors of covariance matrix. MatLab and Python automatically rank the eigenvalues in descending order, and orders the eigen vectors accordingly. Descending order is more natural because eigenpairs are important for further analysis in dimension reduction. In statistics, Principal Component Analysis (PCA) [Jolliffe1995] is also known as discrete Karhunen-Lo`eve (KL) transform which is used for extracting patterns from complex data sets by reducing the dimensionality of complex data set.

B. Definitions and properties of Vectors

Definition. A *vector* is an ordered set of finite number of elements and is denoted by a *column vector* **u**. Almost all the time we encounter vectors with numeric values for elements. In fact, they can be of any valid type.

The vector *notation*: a vector is denoted by a *bold lowercase* character. The elements of a vector are *italic lowercase*. For example, $\mathbf{u} = [u_1, u_2, \ldots, u_n]$ is a row vector, $\mathbf{v} = \begin{bmatrix} v_1 \\ \ldots \\ v_n \end{bmatrix}$ is a column vector.

Definition. The *transpose* of a vector **u** is denoted by $\mathbf{u}^T$. Transpose of a column vector is a row vector, and transpose of row vector is column vector. The transpose of a column vector u is written as $\mathbf{u}^T = [u_1, u_2, \ldots, u_n]$ or u may also be written as $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$.

Definition. The scalar multiple of a vector **u** by a scalar s is denoted by s**u** and is obtained by multiplying each component of **u** by s: $s\mathbf{u} = [su_1, su_2, \ldots, su_n]^T$.

Definition. For any vector **u**, the *norm or length* of **u** is denoted by $|\mathbf{u}|$ and is the square root of the sum of squares of its components:

$$|\mathbf{u}| = \sqrt{(u_1^2 + \ldots + u_n^2)}, \quad |\mathbf{u}|^2 = u_1^2 + \ldots + u_n^2 = \mathbf{u} \bullet \mathbf{u} = \mathbf{u}^T\mathbf{u}.$$

Definition. A vector **u** is a *unit vector* if it is of unit length, $|\mathbf{u}|^2 = 1$, $\mathbf{u} \bullet \mathbf{u} = \mathbf{u}^T\mathbf{u} = 1$.

Definition. The *sum* of two vectors **u** and **v** is written as **u+v** and is defined as vector whose elements are sums of respective components of **u** and **v**, e.g., $\mathbf{u+v} = [u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n]^T$.

Definition. The *dot product* of **u** and **v** is denoted by **u•v**. It is defined in several different equivalent ways.

Analytically dot product **u•v** is the sum of products of components of **u** and **v**: $\mathbf{u} \bullet \mathbf{v} = u_1v_1 + \ldots + u_nv_n$.

Geometrically dot product is $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}||\mathbf{v}|\cos(\theta)$ where θ is the angle between the directions **u** and **v**.

Matrix product form the dot product is expressible as a row-column matrix multiplication $\mathbf{u} \bullet \mathbf{v} = \mathbf{u}^T\mathbf{v} = [u_1 \quad \ldots \quad u_n] \begin{bmatrix} v_1 \\ \ldots \\ v_n \end{bmatrix}$.

Property: If **u** and **v** and two vectors, it is true that $|\mathbf{u} \bullet \mathbf{v}| \leq |\mathbf{u}||\mathbf{v}|$. It follows trivially from geometric definition of dot product and cosine an angle that is less than or equal to 1.

Property. Dot product is *commutative*.

$\mathbf{u} \bullet \mathbf{v} = u_1v_1 + \ldots + u_nv_n = v_1u_1 + \ldots + v_nu_n = \mathbf{v} \bullet \mathbf{u}$, or $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$.

Definition. The vectors **u** and **v** are *orthogonal* if

$$\mathbf{u} \bullet \mathbf{v} = \mathbf{u}^T\mathbf{v} = 0.$$

Definition. A set of vectors is *orthonormal* if each vector is a unit vector, and any two different vectors are mutually orthogonal.

Matrices are used to represent data elegantly and efficiently for visual inspection. All the knowledge is hidden in the tables. Rows can be interpreted as classification rules with attribute values. One of the attributes can be a classification attribute.

The matrix *notation*: an mxn matrix is denote by $A = [a_{ij}]$ where the ij-th element of matrix A is denoted by a_{ij}. For example, any matrix can be represented systematically by using corresponding elements: $A = [a_{ij}]$, $U = [u_{ij}]$, $V = [v_{ij}]$, $S = [s_{ij}]$. If A is a matrix, $\mathbf{a_{i\bullet}}$ is a row vector representing the i-th row of matrix A, and $\mathbf{a_{\bullet j}}$ is a column vector representing the j-th column of A**.** Thus ith row is $\mathbf{a_{i\bullet}} = [a_{i1}, a_{i2}, \ldots, a_{in}]$. Similarly jth column is $\mathbf{a_{\bullet j}} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \ldots \\ a_{mj} \end{bmatrix}$.

Definition. If m=n , then mxn matrix is called a *square* matrix.

Definition. If every entry of a square matrix D is zero except for the diagonal entries d_{ii}, then the matrix D is called *diagonal* matrix. *In general*, for mxn matrix, if every entry except d_{ii}, is zero, it is also diagonal matrix.

ISSN 2395-8618

Definition. If every entry of a square matrix I is zero except for diagonal entries, I_{ii} , which are unity, then the matrix I is called *identity* matrix. A diagonal matrix with diagonal entries 1 becomes identity.

Definition. The *transpose* of a matrix A is denoted by A^T and is defined by interchanging rows to columns or interchanging columns to rows. If A =$[a_{ij}]$, then $A^T = [a_{ji}]$.

Definition. The trace of a matrix A is defined as the sum of entries on its main diagonal. If A = $[a_{ij}]$ then trace(A) = $\sum_i a_{ii}$

Property. For a square matrix A,

$\text{trace}(AA^T) = \text{trace}(A^TA) = \text{trace}(A^2) = |A|^2$

Proof. $\text{trace}(AA^T) = \sum_i \boldsymbol{a_{i\bullet}}\, a_{i\bullet}{}^T = \sum_i \boldsymbol{a_{i\bullet}} \bullet a_{i\bullet}$

$\text{trace}(A^TA) = \sum_i \boldsymbol{a_{\bullet i}}{}^T\, a_{\bullet I} = \sum_i \boldsymbol{a_{\bullet i}} \bullet a_{\bullet i}$

$= \sum_i \sum_k a_{ik}\, a_{ik,} = \sum_i \sum_k a_{ki}\, a_{ki,}$

In either case $= \sum_i \sum_k a_{ik}{}^2 = |A|^2$

It is the sum of squares of all the entries in A,

$|A| = \sqrt{\text{trace}(AA^T)} = \sqrt{\text{trace}(A^TA)} = \sqrt{\text{trace}(A^2)}$

Proposition. For a symmetric matrix A, trace(A) = $\text{trace}(UDU^T)$ = trace(D) = sum of eigenvalues of A, that is, $\text{trace}(A) = \sum_i \lambda_i$

Proof.

$\text{trace}(A) = \text{trace}(UDU^T) =$

$= \sum_i \boldsymbol{u_{i\bullet}}\, D \boldsymbol{u_{i\bullet}}{}^T$

$= \sum_i [\, u_{ik} \lambda_k] \boldsymbol{u_{i\bullet}}{}^T$

where $[\, u_{ik} \lambda_k]$ is a row vector with index k

$= \sum_i \sum_k \; u_{ik} \lambda_k u_{ik}$

$= \sum_k \; \lambda_k \sum_i u_{ik} u_{ik}$

$= \sum_k \; \lambda_k \boldsymbol{u_{\bullet k}} \bullet \boldsymbol{u_{\bullet k}}$ $\quad \mathbf{u}_{\bullet k}$ is a unit column vector.

$= \sum_k \; \lambda_k$

Thus it shows that trace(A) is the sum of eigenvalues of A.

Definition. The mxn matrix A and pxq matrix B are compatible for addition if m=p, and n=q. The *sum* is denoted by A+B and is defined by A+B==$[a_{ij} + b_{ij}]$

Definition. The mxn matrix A and pxq matrix B are compatible for multiplication AB if n=p. If matrices A, B are compatible for multiplication, then *matrix product* AB =$[\sum_{k=1,n} a_{ik}b_{kj}]$ = $[\, r_i c_j]$ = $[\, r_i{}^T \bullet c_j]$ = $[\, r_i \bullet c_j{}^T]$ where r_i is the ith row of A and c_j is the jth column of B.

For a matrix A, the product AA^T and A^TA is called *covariance* of matrix A.

Definition. A square matrix A is *invertible* if there is a matrix B such that AB = I, B is called the *inverse* of A. The *inverse* of invertible matrix A is denoted by A^{-1} so that $AA^{-1} = I$.

Definition. The matrix A is *orthogonal*, if the rows and columns are pairwise orthogonal, and $AA^T = I$, identity matrix.

Property. The *transpose of a product* is the product of transposes in reverse order: $(AB)^T = B^TA^T$.

Definition. The matrix A is *symmetric* if $A = A^T$. The matrix A is self-inverse.

Property. For any matrix A, the product A^TA is a *symmetric* matrix: $(A^TA)^T = A^TA^{TT} = A^TA$.

Definition. The Euclidean *norm* of a matrix A is denoted by |A| and defined by $|A| = \sqrt{\sum_{i,j} (a_{ij}{}^2)}$.

Property: If A and B and two matrices, it is true that $|AB| \le |A||B|$.

Proof. Let r_i be i-th row of A, $r_i{}^T$ be i-th column of A^T. Let c_j be j-th column of B, $c_j{}^T$ be j-th row of B^T. Then

$|AB|^2 = \sum_{i,j} (r_i{}^T \bullet c_j)^2 \le \sum_{i,j} \; |r_i{}^T|^2 |c_j|^2 \le \sum_i |r_i|^2 \sum_j |c_j|^2 \le |A|^2 |B|^2$

Definition. The *rank* of a matrix A is the number of linearly independent rows/columns in a matrix.

Property. The row rank and column rank of a matrix are the same.

Proof. Orthogonal transformation does not change the rank. Since $A = USV^T$, the rank of A is the same as rank of USV^T. It is the same as rank of S. Since S is a diagonal matrix, the row rank and columns rank of a diagonal matrix are same.

Definition. The *determinant* of a matrix A is denoted by det(A). The determinant is computed recursively in terms of row or column and it cofactors.

C. Eigenvalues and Eigenvectors

Definition. Let A be nxn matrix. If there exists a non-zero vector **u** and a number λ such that $A\mathbf{u} = \lambda\, \mathbf{u}$, then λ is called an eigenvalue and **u** is called a corresponding eigenvector.

The equation $A\mathbf{u} = \lambda\, \mathbf{u}$ is called the characteristic equation. If A is an nxn matrix, an eigenvalue of A is a solution of determinant$(A-\lambda I) = 0$. It is a polynomial of degree n, and has n solutions called eigenvalues. The eigenvalues are called eigen (proper, latent, characteristic, singular) values. The eigenvectors are also known as eigen (proper, latent, characteristic, singular) vectors. The eigenvectors and eigenvalues in tandem are referred to as eigenpairs. The coordinate system defined by eigenvectors is called the eigenspace or eigenframe. The transformation matrix is called *rotation* matrix.

Note. An eigenvector is not unique, if **u** is an eigenvector, then any non-zero multiple of **u** is also an eigenvector. To make it *unique*, it is a convention to normalize it to a unit vector, **u**. But **u** and – **u** are unit vectors. Many researchers make first non-zero element in the unit vector positive[Leskovec2014]. This is not satisfactory in some cases: (1) it requires search for the nonzero element and (2) it does not bring about a natural right handed tradition. For example, in Figure A (c), we show a better way to make eigen vectors unique.

ISSN 2395-8618

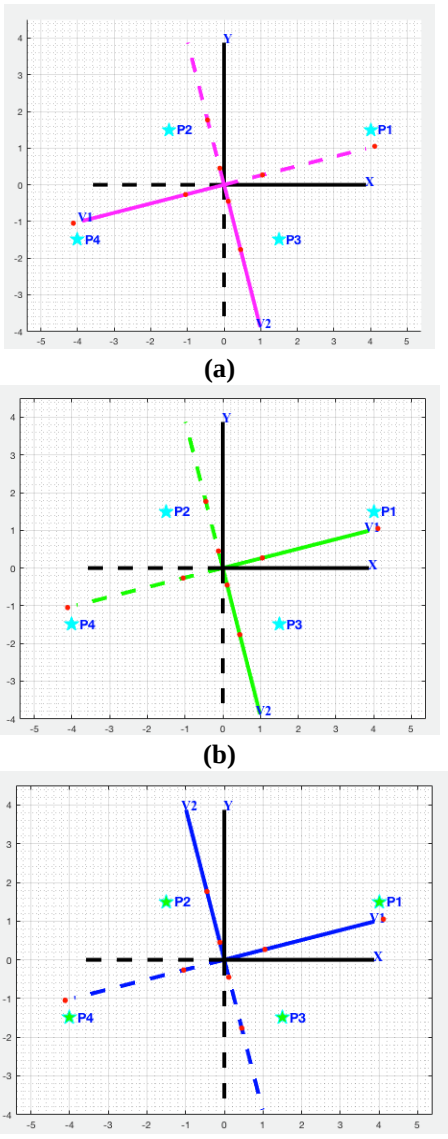


Figure A. (a) Eigenvectors as computed by MATLAB svd, (b) by convention, each vector has first non-zero element positive, (c) our approach, first eigenvector has first non-zero element positive, second eigenvector has second non-zero element positive by using ordering of eigenvectors so the eigenvectors form a right handed system.

Vector space basis is the set of vectors so that every other vector in the space can be expressed as a linear combination of the basis vectors. For R^n, $\mathbf{e_k} = (e_{kj})$ for k=1,n where $e_{kk} = 1$ and e_{kj} is zero for j≠k, $\{ \mathbf{e_k} \}$ is a basis of vectors. In fact any linearly independent set $\{ \mathbf{u_k} \}$ of n vectors can be a basis of R^n. Any n linearly independent, orthonormal unit vectors is an orthogonal basis of R^n.

For SVD, we use these special matrices A^TA and AA^T for calculating the eigenvectors of A^TA and AA^T. Herein, we elaborate the details of some results that we take for guaranteed. For an arbitrary non-symmetric rectangular mxn matrix A, the matrix AA^T is mxm and the matrix A^TA is nxn. Both are symmetric and square matrices.

Proposition The eigenvalues of a real symmetric matrix are real.

Proof. The complex conjugate of u is denoted by $\bar{u}$. Let λ be an eigenvalue of A, then Au = λ u, where u is a unit vector and $\bar{A}\,\bar{u} = \bar{\lambda}\,\bar{u}$. A is real, $A\,\bar{u} = \bar{\lambda}\,\bar{u}$.

$\lambda = \lambda\ \bar{u}^T u = \bar{u}^T \lambda u = \bar{u}^T A u = \bar{u}^T \bar{A}^T u$ for real symmetric A

$= (\bar{A}\,\bar{u})^T u = \bar{\lambda}\ \bar{u}^T u = \bar{\lambda}$

therefore $\lambda = \bar{\lambda}$. Hence λ is a *real* number.

Example. If the matrix is not symmetric, eigenvalues are not necessarily real. For example, let $A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$, it is non-symmetric, its eigenvalues are complex: $1 \pm \sqrt{-1}$.

Example. If A is matrix, it may have repeated eigenvalues. Let $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, it is symmetric, its eigenvalues are 1. The eigenvectors form a basis of the transformed space. Let $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, it is non-symmetric, its eigenvalues are 1. The eigenvectors do not form a basis of the transformed space.

In PCA, for any matrix A, we calculate eigenvalues and eigenvectors of covariance matrices A^TA (and AA^T) which form the basis of vector space of rows (and columns) of matrix A.

Proposition. The eigenvalues of special real symmetric matrices, A^TA and AA^T, are real and non-negative. Not all symmetric matrices have this property.

Proof. if λ is an eigenvalue and **v** is a unit eigenvector of A^TA, then

$A^TA\,\mathbf{v} = \lambda\,\mathbf{v}$ and $\mathbf{v} \bullet \mathbf{v} = 1$.

Now $\lambda = \lambda\,\mathbf{v} \bullet \mathbf{v} = \lambda\,\mathbf{v}^T\mathbf{v}$

$= \mathbf{v}^T \lambda \mathbf{v} = \mathbf{v}^T(A^TA\mathbf{v})$

$= (\mathbf{v}^TA^T)A\mathbf{v}) = (A\mathbf{v})^T(A\mathbf{v})$

$= (A\mathbf{v}) \bullet (A\mathbf{v}) \geq 0$.

That is non-zero eigenvalues of AA^T and A^TA are positive.

Example. Every symmetric matrix does not have this property.

Let $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, it is symmetric, its eigenvalues are -1 and 3. Thus all the eigenvalues of symmetric matrix are *not* always non-negative.

If $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$, it is symmetric, all the eigenvalues are non-negative: 0, 5.

If $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, it is symmetric, all the eigenvalues are positive: 1,3.

Proposition The eigenvectors corresponding to different eigenvalues of a matrix A are linearly independent.

ISSN 2395-8618

Proof. Let $\mathbf{u_1}$ and $\mathbf{u_2}$ be eigenvectors for distinct eigenvalues λ_1 and λ_2. We show that they are linearly independent. Let

$x\mathbf{u_1} + y\mathbf{u_2} = 0$,

then $A(x\mathbf{u_1} + y\mathbf{u_2}) = 0$ or

$x\lambda_1\mathbf{u_1} + y\lambda_2\mathbf{u_2} = 0$, eliminating y we get $x(\lambda_1 - \lambda_2)\mathbf{u}_1 = 0$, since $(\lambda_1 - \lambda_2)\mathbf{u}_1 \neq 0$, $x = 0$ similarly $y = 0$, hence they are linearly independent.

This is true for any number of eigenvectors corresponding to different eigenvalues. It is a useful method of solution of n linear equations.

Proposition The eigenvectors corresponding to different eigenvalues of a real symmetric matrix A are orthogonal.

Proof. Let $\mathbf{u}$ and $\mathbf{v}$ be eigenvectors for eigenvalues λ and μ where $\lambda \neq \mu$.

$$\text{Then } \lambda \mathbf{u^T v} = (\lambda \mathbf{u^T})\mathbf{v} = (A\mathbf{u})^T \mathbf{v} = \mathbf{v}^T(A\mathbf{u}) = (\mathbf{v}^T A^T)\mathbf{u}$$
$$= (A\mathbf{v})^T \mathbf{u} = (\mu v)^T \mathbf{u} = \mu \mathbf{v}^T\mathbf{u}$$
$$= \mu \mathbf{u}^T\mathbf{v}$$

Now $\lambda \mathbf{u^T v} = \mu \mathbf{u^T v}$ or $(\lambda - \mu)\mathbf{u^T v} = 0$.

Since $\lambda \neq \mu$, $\mathbf{u^T v} = 0$ or $\mathbf{u \cdot v} = 0$, therefore $\mathbf{u}$ and $\mathbf{v}$ are orthogonal.

In SVD, we use AA^T and A^TA which are naturally symmetric.

If the eigenvectors are not orthogonal, it will defeat the purpose of simplicity and efficiency. It is possible that an eigenvalue of a matrix is of multiplicity greater than one, that is, corresponding to an eigenvalue there may be several eigenvectors, not necessarily orthogonal. In that case, we can use Gram-Schmit orthogonalization process to create orthogonal set of eigenvectors.

Property [22] . Any real symmetric matrix A can be written as $A = UDU^T = UDU^{-1}$ for some invertible matrix U. Here U is the matrix of eigenvectors of A whereas D is the diagonal matrix of eigenvalues of matrix A.

Proof. Let U be matrix of eigenvectors of matrix A. If $\mathbf{u}_k$, λ_k is an eigenpair of A, the $A\mathbf{u_k} = \lambda_k \mathbf{u_k}$ or

$$A\mathbf{u_k} = \mathbf{u_k}\lambda_k.$$

Then

$$AU = A[\mathbf{u_k}] = [A\mathbf{u_k}] = [\lambda_k \mathbf{u_k}] = [\mathbf{u_k}\lambda_k] = [\mathbf{u_k}]D = UD$$

Since U is invertible matrix, we have

$$A = UDU^{-1}$$

The eigenvectors may be orthogonal, U is orthogonal matrix.

Thus $A = UDU^{-1} = UDU^T$

The eigenvalues may not be positive, except for signs, they are square roots of the eigenvalues of A^2.

Corrolary. Since AA^T is symmetric, therefore $AA^T = UDU^T$, where U is the eigenvector matrix and D is the eigenvalue matrix of AA^T.

Proposition For matrix AA^T, let u be an eigenvector corresponding to non-zero eigenvalue λ. Then A^Tu is an eigenvector of A^TA with the eigenvalue λ.

Proof. Let $\mathbf{u}$ be an eigenvector of AA^T and λ be the corresponding non-zero eigenvalue. Then

$$AA^T \mathbf{u} = \lambda \mathbf{u}$$

Since eigenvalue $\lambda \neq 0$ $\mathbf{u} \neq 0$, therefore $A^T\mathbf{u}$ is a non zero eigenvector and now

$$A^TA(A^T\mathbf{u}) = A^T(AA^T\mathbf{u})$$
$$= A^T\lambda\mathbf{u}$$
$$= \lambda A^T\mathbf{u}$$
$$= \lambda(A^T\mathbf{u})$$

Therefore $A^T\mathbf{u}$ is an eigenvector of A^TA with eigenvalue $\lambda \neq 0$.

Similarly if $\mathbf{v}$ is an eigenvector of A^TA and λ be a corresponding non-zero eigenvalue of A^TA, $A\mathbf{v}$ is an eigenvector of AA^T.

Proposition Let $\mathbf{v}$ be an eigenvector of A^TA and λ be a corresponding non-zero eigenvalue. Then $A\mathbf{v}$ is an eigenvector of AA^T.

Proof. Let $\mathbf{v}$ be an eigenvector of A^TA and λ be the corresponding non-zero eigenvalue. Then

$$A^TA\mathbf{v} = \lambda \mathbf{v}$$

Since eigenvalue $\lambda \neq 0$, $\mathbf{v} \neq 0$, therefore $A\mathbf{v}$ is non zero and now

$$AA^T(A\mathbf{v}) = A(A^TA\mathbf{v})$$
$$= A\lambda\mathbf{v}$$
$$= \lambda A\mathbf{v}$$
$$= \lambda(A\mathbf{v})$$

Therefore $A\mathbf{v}$ is an eigenvector of AA^T with eigenvalue $\lambda \neq 0$.

Proposition. Let $\mathbf{v}_k$ be an eigenvector of A^TA for non-zero eigenvalue λ_k. Then $A\mathbf{v}_k$ is an eigenvector of AA^T, say, $\mathbf{u}_k$, and that $A\mathbf{v}_k = \sigma_K \mathbf{u}_k$ or $\mathbf{u}_k = (1/\sigma_K) A\mathbf{v}_k$ where σ_K is the square root of the corresponding eigenvalue λ_k of A^TA.

Proof. Since $\mathbf{u}_k$ are unit vectors eigenvectors of AA^T , and $\mathbf{v}_k$ are unit vectors eigenvectors of A^TA, $A\mathbf{v}_k$ is some scalar multiple of $\mathbf{u}_k$.

Let $A\mathbf{v}_k = \sigma_k u_k$ for some non-zero σ_K. Since u_k is a unit vector,

$$\sigma_k^2 = \sigma_k \mathbf{u}_k \cdot \sigma_k \mathbf{u}_k = A\mathbf{v}_k \cdot A\mathbf{v}_k$$
$$= \mathbf{v}_k \cdot A^TA\mathbf{v}_k = \mathbf{v}_k \cdot \lambda_k \mathbf{v}_k = \lambda_k$$

or

$$\lambda_k = \lambda_k \mathbf{v}_k \cdot \mathbf{v}_k = A^TA\mathbf{v}_k \cdot \mathbf{v}_k$$
$$= A\mathbf{v}_k \cdot A\mathbf{v}_k = \sigma_k \mathbf{u}_k \cdot \sigma_k \mathbf{u}_k = \sigma_k * \sigma_k = \sigma_k^2$$

Therefore $\sigma_k^2 = \lambda_k$ or $\sigma_k = \sqrt{\lambda_k}$

Hence σ_k is a square root of eigenvalue λ_k .

D. Singular Value Decomposition

Any symmetric positive semi-definite matrix A can be represented as the product of three matrices U, S, V^T where U

ISSN 2395-8618

and V are orthogonal matrices of eigenvectors of AA^T and A^TA; and S is a matrix whose diagonal entries are square roots of eigenvalues of AA^T and A^TA.

Proposition The eigenvalues of AA^T and A^TA are identical except for the zero eigenvalues. Here λ is a non-zero eigenvalue of AA^T if and only if it is eigenvalue of A^TA.

Proof. λ is an eigenvalue of AA^T implies there is a non-zero vector $\mathbf{u}$ such that $AA^T\mathbf{u} = \lambda\mathbf{u}$

$AA^T\mathbf{u} = \lambda\mathbf{u}$ implies $A^TAA^T\mathbf{u} = \lambda A^t\mathbf{u}$

or $A^TA(A^T\mathbf{u}) = \lambda(A^T\mathbf{u})$

which means λ is an eigenvalue of A^TA.

Similarly if λ is an eigenvalue of A^TA, there is a non-zero vector $\mathbf{v}$ such that $A^TA\mathbf{v} = \lambda\mathbf{v}$ implies $AA^TA\mathbf{v} = \lambda A\mathbf{v}$

or $AA^T(A\mathbf{v}) = \lambda(A\mathbf{v})$

which means λ is an eigenvalue of AA^T.

Proposition. If $A = USV^T$ where matrices U and V are orthogonal then U is matrix of eigenvectors of AA^T, V is a matrix of eigenvectors of A^TA and S is diagonal matrix of square roots of non-zero eigenvalues, and conversely.

Proof.

Since $A = USV^T$

then $AA^T = USV^T(USV^T)^T$

$= USV^T(V^{TT}S^TU^T)$

$= USV^T(V^{TT}S^TU^T)$

$= USV^TVS^TU^T$

$= US^2U^T$

Therefore $AA^TU = US^2$.

That is $AA^T\mathbf{u_k} = \mathbf{u_k}s_k^2$ for vectors $\mathbf{u_k}$.

Thus U is matrix of eigenvectors $\mathbf{u}_k$ of AA^T. The diagonal entries s_k^2 of S^2 are eigenvalues of AA^T. Thus the entries s_k of S are square roots of eigenvalues of AA^T.

Similarly we can verify that V is the matrix of eigenvectors of A^TA.

Conversely, to prove the converse, let $\lambda_k, \mathbf{v}_k$ be eigenpair for A^TA, then $A^TA\mathbf{v}_k = \lambda_k\mathbf{v}_k$

We seen above that the eigenvalues of AA^T and A^TA are identical. Now as seen above that for non-zero eigenvalues, the relation between eigenvectors of A^TA and AA^T is $A\mathbf{v}_k = \sqrt{\lambda_k}\,\mathbf{u}_k$ where $\mathbf{v}_k$ is an eigenvector of A^TA and $\mathbf{u}_k$ is an eigenvector of AA^T

For any n-vector x, it can be expressed as linear combination of $\mathbf{v}_k$'s

$\mathbf{x} = \mathbf{x}\bullet\mathbf{v_1}\mathbf{v_1} + \ldots + \mathbf{x}\bullet\mathbf{v_n}\mathbf{v_n}$

$A\mathbf{x} = \mathbf{x}\bullet\mathbf{v_1}A\mathbf{v_1} + \ldots + \mathbf{x}\bullet\mathbf{v_n}A\mathbf{v_n}$

$A\mathbf{x} = A\mathbf{v_1}\mathbf{x}\bullet\mathbf{v_1} + \ldots + A\mathbf{v_n}\mathbf{x}\bullet\mathbf{v_n}$

$A\mathbf{x} = A\mathbf{v_1}\mathbf{v_1}\bullet\mathbf{x} + \ldots + A\mathbf{v_n}\mathbf{v_n}\bullet\mathbf{x}$

From $A\mathbf{v_k} = \sqrt{\lambda_k}\,\mathbf{u_k}$ for k=1,n

$A\mathbf{x} = \sqrt{\lambda_1}\,\mathbf{u_1}\mathbf{v_1}\bullet\mathbf{x} + \ldots + \sqrt{\lambda_n}\,\mathbf{u_n}\mathbf{v_n}\bullet\mathbf{x}$

$A\mathbf{x} = \sqrt{\lambda_1}\,\mathbf{u_1}\mathbf{v_1}^T\mathbf{x} + \ldots + \sqrt{\lambda_n}\,\mathbf{u_n}\mathbf{v_n}^T\mathbf{x}$

$A\mathbf{x} = (\mathbf{u_1}\sqrt{\lambda_1}\mathbf{v_1}^T + \ldots + \mathbf{u_n}\sqrt{\lambda_n}\,\mathbf{v_n}^T)\mathbf{x}$

Since this true for any vector x,

$\mathbf{A} = (\mathbf{u_1}\sqrt{\lambda_1}\mathbf{v_1}^T + \ldots + \mathbf{u_n}\sqrt{\lambda_n}\,\mathbf{v_n}^T)$

Therefore we have proved that $A=USV^T$

E. Calculating PCA from SVD.

We prove that for a *symmetric* matrix A with non-negative eigenvalues, PCA can be derived from SVD. If the columns of U are eigenvectors of AA^T, the columns of V are eigenvectors of A^TA, the diagonal entries of S square root of eigenvalues of A^TA, then SVD of A be $A=USV^T$.

Proposition. Let A be a symmetric matrix positive semi-definite, the $A=USU^T$, with columns of U are eigenvectors of $A^TA = A^2$ if and only if columns of U are eigenvectors of A.

Proof. By SVD algorithm

$A = USV^T$

where The columns of U are eigenvectors of A^2; the diagonal entries of S, square roots of eigenvalues of A^2.

Since A is a symmetric square matrix,

U=V and consequently $A=USU^T$

However,

$A = USU^T$ implies $AU = US$

It means that the columns of U are eigenvectors of A and the diagonal entries of S are eigenvalues of A.

Thus the columns of U are eigenvectors of A^2 if and only if eigenvectors of A, the diagonal entries of S are square roots of eigenvalues of A^2 iff the eigenvalue of A are non-negative.

Note. If A is not positive semi-definite, D can have negative entries corresponding to negative eigenvalues. In this case, PCA cannot be derived for SVD, see example below.

Example. The matrix $A=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$ is symmetric so is $AA^T = A^TA = A^2 = \begin{bmatrix}5 & 4\\ 4 & 5\end{bmatrix}$. The eigen values of A are 3, -1, singular value of A are 3,1.

Eigenvectors of $A=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$ are $\frac{\begin{bmatrix}1\\1\end{bmatrix}}{\sqrt{2}}$ and $\frac{\begin{bmatrix}1\\-1\end{bmatrix}}{\sqrt{2}}$ corresponding to eigenvalues 3and -1.

Eigenvectors of $A=\begin{bmatrix}5 & 4\\ 4 & 5\end{bmatrix}$ are $\frac{\begin{bmatrix}1\\1\end{bmatrix}}{\sqrt{2}}$ and $\frac{\begin{bmatrix}1\\-1\end{bmatrix}}{\sqrt{2}}$ corresponding to eigenvalues 9 and 1.

Eigenvectors are same, PCA and SVD are not some..

PCA: $UDU^T = \frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}}\begin{bmatrix}3 & 0\\0 & -1\end{bmatrix}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}3 & -1\\3 & 1\end{bmatrix}}{\sqrt{2}}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}2 & 4\\4 & 2\end{bmatrix}}{2} = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} = A.$

However, SVD: $USU^T = \frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}}\begin{bmatrix}3 & 0\\0 & 1\end{bmatrix}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}3 & 1\\3 & -1\end{bmatrix}}{\sqrt{2}}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}4 & 2\\2 & 4\end{bmatrix}}{2} = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$ is not the same as A.

Hence PCA cannot be derived from SVD if A is not postitive

ISSN 2395-8618

semi-definite.

Orthogonal matrix is also called a *rotation* matrix, because this matrix rotates the original coordinate axes. Rotation does not change lengths and relative angles as seen below.

Property. If R is orthogonal matrix and **u** is a vector, then

$$|R\mathbf{u}|=|\mathbf{u}|$$

Proof.

$|R\mathbf{u}|^2=(R\mathbf{u})^T R\mathbf{u} = \mathbf{u}^T R^T R\mathbf{u}$,
since R is orthogonal $R^TR=I$
$|R\mathbf{u}|^2 = \mathbf{u}^T I\mathbf{u} = \mathbf{u}^T\mathbf{u} = |\mathbf{u}|^2$
Therefore $|R\mathbf{u}|=|\mathbf{u}|$

Property. If R is orthogonal matrix and A is matrix, then

$$|RA|=|A|$$

Proof. Let $a_{\bullet j}$ be j-th column of A. Using the rotation property of vectors,

$$|RA|^2 = \sum_{j=1,n} |R\, a_{\bullet j}|^2 = \sum_{j=1,n} |a_{\bullet j}|^2 = |A|^2$$

Property. If U and V are orthogonal and S is a diagonal matrix, then

$$|USV^T|=|D|$$

Proof. Using the rotation property of matrices,

$$|USV^T| = |SV^T| = |VS^T| = |S^T| = |S| = |D|$$

Property. If rows/columns corresponding to smaller variation are deleted, there is smaller loss of information. If rows/columns corresponding to zero eigenvalues only are deleted, then there is no loss of information in the reduced dimensionality.

Proof. By SVD, there exist U, V, S such that $A = USV^T$. Now $A' = U(:,1:k)\ S(1:k,1:k)\ V(1:k,:)^T$ by deleting m-k columns after first k columns in U and n-k columns after first k columns in V, after deleting all rows and all columns after first k rows and k columns in S. Let S_{new} be S corresponding to dimension reduction, by zeroing all eigenvalues except first k diagonal entries. Let S_{new} correspond to dimension reduction. The A' is the same as $B= US_{new}V^T$. In this reduction, loss of information is |A-B|, whereas A and B have the same size mxn.

Now $A=USV^T$ $B=US_{new}V^T$

$$|A-B| = |USV^T - US_{new}V^T|$$
$$= |U(S - S_{new})V^T|$$

using orthonormality of column vectors of U and V we have

$$|A-B| = |S - S_{new}|$$
$$= |S - S_{new}|$$
$$= \sqrt{(\sum_{p>k} s_{pp}^2)}$$
$$= \sqrt{(\sum_{p>k} \lambda_p)}$$

This shows that the smaller the value of $\sqrt{(\sum_{p>k} \lambda_p)}$, the smaller the norm |A-B|, the closer A and B. If all eigenvalues λ_p with p>k are zero, then there is no loss of information.

Here are two interesting result.

Property. If A is symmetric positive semi definite, the $A= B^TB$ for some symmetric positive semi definite B.

Proof. By SVD, we have $A= USU^T$. The entries if S are non-negative. Let $B =U\sqrt{S}U^T$. Since A is symmetric, B is symmetric.

$$BB^T = U\sqrt{S}U^T(U\sqrt{S}U^T)^T$$
$$= U\sqrt{S}U^T\, U^{TT}\sqrt{S}^TU^T$$
$$= U\sqrt{S}U^T\, U\sqrt{S}U^T$$
$$= U\sqrt{S}\sqrt{S}U^T$$
$$= USU^T$$
$$= A$$

This property show that SVD transforms correlated data into uncorrelated data.

Property. If A is symmetric positive semi definite, there is a transformation M such that covariance matrix $MA(MA)^T$ is diagonal.

Proof. By SVD, we have $A= USU^T$.

Let $M=U^T$

Then $MA= SU^T$

Now $MA(MA)^T = SU^T (SU^T)^T$
$= SU^T US^T$
$= SS^T$
$= S^2$

Note. Let A is mxn, U is mxm, V is nxn, $Av_k = \sigma_k u_k$ and if $AA^t u_k = \lambda_k u_k$, then $A^t u_k = \sigma_k v_k$ and $\sigma_k = \sqrt{\lambda_k}$.

If m<n, we compute V first and then U*S=AV. If m>n, then we compute U first and then $V*S^I = A^TU$.

Since S is diagonal, its inverse is reciprocal of the diagonal entries, except for zero entries which are left unchanged. In case of zero entries, it becomes Pseudo inverse denoted by S^I. Pseudo inverse is left inverse if m>n otherwise it is left inverse.

Eitherway $U = AV\ S^I$ or $V^T = S^I\ A^TU$ or $V = U^TA\ S^I$ where S^I is pseudo inverse. This is computationally more stable.

Once U, and V are computed, S can be quickly verified from $S = U^TAV$.

REFERENCES

[1] Saraçlı, S., Yılmaz, V., & Doğan, İ. (2009b). Simple linear regression techniques in measurement error models. *Anadolu University Journal of Science and Technology*, 10(2), 335-342.

[2] Stefanski, L.A. (2000). Measurement error models. *Journal of the American Statistical Association*, 95(452), 1353-1358.

[3] McCartin, B. J, (2003). A geometric characterization of linear regression. *Statistics*, 37(2), 101–117. http://dx.doi.org/10.1080=0223188031000112881

[4] Ding, G., Chu, B., Jin, Y., & Zhu, C. (2013). Comparison of orthogonal regression and least squares in measurement error modeling for prediction of material property. *Nanotechnology and Material Engineering Research, Advanced Materials Research*, 661, 166-170.

ISSN 2395-8618

http://dx.doi.org/10.4028/
www.scientific.net/AMR.661.166

[5] Leng, L., Zhang, T. Kleinman, L., & Zhu, W. (2007). Ordinary least square regression, orthogonal regression, geometric mean regression and their applications in aerosol science. *Journal of Physics, Conference Series* 78(1), 1-5. Retrieved from http://iopscience.iop.org/1742-6596/78/1/012084

[6] Steven C Chapra and Raymond P Canale, Numerical Methods for Engineers, 7th Edition, ISBN: 978 0073397924, McGraw-Hill Publishers, 2015.

[7] Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates.

[8] Draper, N.R.; Smith, H. (1998). Applied Regression Analysis (3rd ed.). John Wiley. ISBN 0-471-17082-8.

[9] Taliha Keles, Comparison of Classical Least Squares and Orthogonal Regression in measurement error models, International Online Journal of Educational Sciences, 10(3), 200-20014.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. Computer 42, 8 (2009).

[11] Stephen Vaisey, Treatment Effects Analysis, https://statisticalhorizons.com/seminars/public-seminars/treatment-effects-analysis-spring17.

[12] Gwowen Shieh, Clarifying the role of mean centering in multicollinearity of interaction effects, British Journal of Mathematical and Statistical Psychology (2011), 64, 462–477.

[13] Jim Hefferon, Linear Algebra, Free Book, http://joshua.smcvt.edu/linearalgebra, 2014.

[14] John F. Hughes, AndriesVan Dam,Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akler Computer Graphics: principle and Practice, 3rd edition , Addison Wesley, 2014.

[15] Chaman Sabharwal, Hybrid Linear Least Square and Singular Value Decomposition Approximation, International Journal of Trend in Research and Development, Volume 5(3), ISSN: 2394-9333 www.ijtrd.com May-Jun 2018, pp. 1-8.

[16] Sabharwal, Chaman Lal, SVD Adaptive Algorithm for Linear Least Square Regression and Anomaly Reduction, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. III (Sep - Oct 2018), PP 33-48, www.iosrjournals.org

[17] Carr, J.R. (2012). Orthogonal regression: A Teaching perspective. *International Journal of Mathematical Education in Science and Technology*, 43(1), 134-143. http://dx.doi.org/10.1080/0020739X.2011.573876

[18] P. Groves, B. Kayyali, D. Knott, S. V. Kuiken, "The 'Big Data' Revolution in Healthcare", *Center of US Health System Reform Business Technology Office*, pp. 1-20, 2013.

[19] C. C. Yang, L. Jiang, H. Yang, M. Zhang, "Social Media Mining for Drug Safety Signal Detection" *ACM SHB'12*, October 29, 2012, Maui, Hawaii, USA.

[20] Jure Leskovec, Anand Rajaraman, Jeffrey D Ullman, Datamining of Massive Datasets, 2014.

[21] Patrick J.F. Groenen, Michel van de Velden, Multidimensional Scaling, Econometric Institute EI 2004-I5, Erasmus University Rotterdam, Netherlands, 2015.

[22] Jonthan Shlens A Tutorial on Principal Component Analysis, arXiv:1404.1100 [cs.LG], pp. 1-15,2014

[23] Sebastian Raschka Principal Component Analysis in 3 Simple Steps LSA-Least Squares Approximation http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html, 2015.

[24] Abdi, Hervé, Beaton, Derek, Principal Component and Correspondence Analyses Using R, Springer, ISBN 978-3-319-09256-0, Digitally watermarked, DRM-free, 2017.

[25] Caroline J Anderson, Psychology Lecture Notes: Principal Component Analysis, 2017.

[26] H. Y. Chen, R. LiÅLegeois, J. R. de Bruyn,and A. Soddu, "Principal Component Analysis of Particle Motion", *Phys. Rev.* E 91, 042308 - 15 April 2015.

[27] Karen Bandeen-Roche Nov 28, 2007, An Introduction to Latent variable Models, http://www.biostat.jhsph.edu/~kbroche/Aging/Intro to Latent VariableModels.pdf.

[28] Yusuke Ariyoshi and Junzo Kamahara. 2010. A hybrid recommendation method with double SVD reduction. In International Conference on Database Systems forAdvanced Applications. Springer, 365–373.

[29] Chaman Sabharwal, Principal Component Analysis and Qualitative Spatial Reasoning, 28th International Conference on Computer Applications in Industry and Engineering, CAINE 2015, October 12-14, 2015, San Diego, California, USA pp.23-28.

[30] Matlab, https://www.mathworks.com/downloads/

[31] Mark Tygert, Regression-aware decompositions, arXiv1710.04238v2, 12 Feb 2018.

ISSN 2395-8618

Simple and Effective Feature Based Sentiment Analysis on Product Reviews using Domain Specific Sentiment Scores

Nachiappan Chockalingam

***Abstract*—Reviews are a valuable resource. Conclusions drawn on analysis of reviews are of great help in improving the product, as far as the manufacturer is concerned, or with predicting sales figures, as far as the retailer is involved. However, employing human labor to go through all the reviews manually would be a time consuming and expensive process. This paper outlines a novel technique to extract features from a product's reviews along with the corresponding sentiment expressed, using POS tagging and Dependency Parsing in conjunction. The use of these of these allows both the context and the parts of speech of a word to be employed in feature and corresponding opinion word detection. The opinion word is given a sentiment polarity determined from a training set of positive and negative reviews.The method described in this paper is for large data sets, and requires no domain specific data for feature extraction.**

***Index Terms*—Review mining, dataset, sentiment analysis, features, parts of speech tagging, opinion word, dependency parsing.**

I. Introduction

REVIEWS are a set of sentiments expressed over a very short period of time about a product and it's features. The number of reviews and reviewers are only increasing by the day; a trend that shows no sign of abating. Hence, the idea of review analysis to tap into this goldmine of freely available data is alluring.

Numerous systems talk about sentiment analysis to gain the 'average' response for a product [1], [2]. This one dimensional take on the issue ignores the potential for a multi-faceted approach where even individual features of a product can be extracted and analysed. After all, why not use the average star rating? Why even enter text analysis if not to extract 'more' information about/from reviews.

The aim of the proposed system is to extract features from reviews using a series of techniques. Evaluation formulas of precision and recall allow for classification of problems of feature extraction. These being, either find a lot of features but accept a low precision score since there would be a number of unwanted features included in the feature list, or gain in precision by applying additional filtering to the feature list while contending with the possibility of loss of genuine features.

Following feature extraction, polarity classification is done. This step involves assigning scores to opinion words. The opinion words are associated with a feature, and hence the score for the opinion word is linked with that feature. This system works best when a large number of reviews are input (since each feature needs sufficient opinion words describing it).

A. Problem Statement

Given reviews of a particular product, the aim is to summarise the reviews by picking out features and their corresponding opinion words with polarity scores [3].

The Screen is bright and clear-Using this example sentence, the problem statement is explained in steps.

1. Extract all features from given reviews: As there is no previous data about the features to look for, they have to be generated on the go, from the data. Eg: Screen
2. Generate opinion word: The opinion words are extracted, again in the absence of a specific domain. Eg: Bright and Clear
3. Generate Opinion Scores: While feature extraction is not domain specific, the opinion word scores are machine learnt, and hence can be domain specific. Negation and conjunction must be handled. Eg: Bright (Positive), Clear (Positive)
4. Put all the feature analysis together to generate a feature score that is more accurate (hence the large dataset).

B. Literature Survey

Many systems have been proposed for the analysis of reviews and the work on this domain has been on-going for close to two decades. Growth in required and related fields such as e-commerce, computational power, machine learning, and most importantly, Text Analytics has allowed crossing of barriers previously applied on researchers working in this field.

The typical Text Analysis approach uses Cleaning (pre-processing), Analysis and result generation. However, within each broad step, techniques used differ between each sentiment analysis system.

There are different types of sentiment analysis including sentence Level, document Level, aspect-based mining, etc. [4]. All of these are dependent on the domain and aim.

Manuscript received on December 28, 2017, accepted for publication on March 15, 2018, published on June 30, 2018.

The author is with the Department of Computer Science and Engineering, College of Engineering, Guindy, Chennai, Tamil Nadu, 600025, India (e-mail: nach729@hotmail.com).

ISSN 2395-8618

One of the earliest sentiment mining methods included the classification of sentences into positive and negative [5], [6] groups. Further work involved a comprehensive entry into sentence and document level sentiment analysis. Document level analysis is used in a similar case as sentences level analysis since "sentences are just short documents" [7]. Aspect based sentiment analysis is a reference to the level of rating. It allows for identification of features and generating their polarity from the reviews, as opposed to polarity classification of reviews as a whole [4]. Aspect based sentiment analysis uses sentence/document analysis combined with aspect level rating.

Bag of words model is a famous Text Mining [8] approach where the un-needed parts of a text are discarded in favour of keeping ones that are necessary. Many older systems relied on the use of stop words removal as a method to extract desired data. Instead of that approach, the ability to POS tag a sentence coupled with tuple analysis allows for extraction of desired data directly [3], instead of discarding unwanted text. The use of dependency tagging helps maintain context of the word [9].

With regards to polarity determination, Ohana et all [6] used Senti-Word Net to get the word sentiments for identified opinion words. Synsets (sentiment scores) for a particular word were taken and averaged to generate it's polarity. But this lacks domain specific identity (Section 2.1) that provides an authentic score for any specific domain. This is its greatest pitfall.

II. Proposed System

The system proposed uses Parts of Speech Tagging (POS) to parse sentences into constituent elements while Dependency Parsing is used to determine the relationships between words. A rule based analysis can be applied, using which the features and opinion words are extracted. Finally, a sum of all the analysis gives us the perception of each feature.

A. Sentiment Scores

In the related work section, there are issues with determining of sentiment scores for other approaches using pre-determined or previously calculated sentiment scores [6] for opinion words. So, for example, the sentiment score for the opinion word 'sad' is applied across electronics reviews, as well as movie reviews. This leads to inaccurate results since the same opinion word does not correspond to the same sentiment across domains. While the word 'bad' might be acceptable as a universal negative sentiment modifier, many other words do not carry a universal sentiment.

A simple machine learning system with domain specific dataset is used in this system, where the input dataset is of the same domain as the reviews to be analysed. To begin, two datasets- positive and negative are input. Next, each review undergoes pre-processing (Section 2.3) and analysis (Section 2.4) steps outlined later in this paper. During tuple analysis sentences are parsed into nouns and adjectives(opinion words).

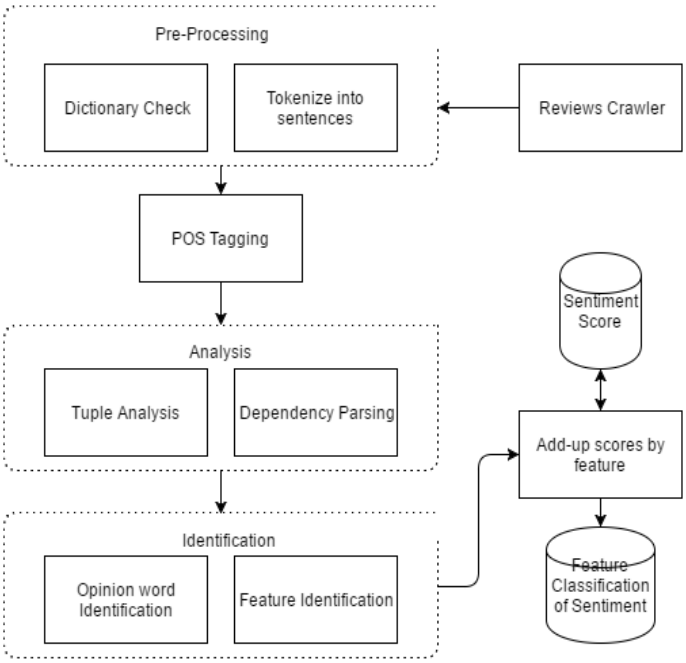


Fig. 1. Method overview.

The fact that we know the polarity of the review input allows us to classify the opinion words into two classes- positive and negative. Each opinion word has a negative and positive counter and every time an opinion word is identified, the counter is iterated for either the positive or negative respectively as found [2].

The training corpus for negative and positive reviews is from the work by Ganapathibhotla and Bing Liu [10].

$$PositiveScore = \frac{PositiveCounter}{PositiveCounter + NegativeCounter}$$

The opinion word score is positive biased. That is to say, all scores are a continuous from 0 to 1. 0 being the most negative and 1 being the most positive. When a word makes no appearance in negative or positive datasets as an opinion word, then the word (as per the formula) will be assigned a score of 1 or 0 respectively. A score of 0.5 represents a neutral sentiment.

B. Pre-Processing

In this stage, the input reviews need to be brought to a format convenient for analysis. Reviews are pushed through a dictionary correction module, parsed into sentences and sent into the analysis system one by one.

C. Analysis

The analysis stage is split into two- the tuple analysis and the dependency parsing. The input to both stages is done after the sentence is POS tagged.

ISSN 2395-8618

1) Tuple Analysis: Tuple analysis involves taking a sentence and turning it into a tuple. A tuple is a stripped version of a sentence, in that it contains only essential parts required for analysis. For example, the Person, Nouns, Adjectives, time, etc found in a sentence are stored in a tuple, and hence it represents what is relevant (to the analysis) in that sentence. For this system, nouns and adjectives are extracted in the absence of Domain Knowledge [3]. The nouns are henceforth referred to as 'potential features', while the adjectives are 'potential opinion words'. Hence a sentence is reduced to:

```
<Potential Features;
Potential Opinion Words>
```

A number of important relationships between words that affect the identification of features and their corresponding sentiment scores remain unknown such as which opinion word corresponds to which feature, conjuntion and negation in the sentence, etc.

2) Dependency Parsing: The dependency parser is effective in taking the POS tagged sentences and obtain the relationship between words. This section can also be called relation extraction, [3] as stated by Mukherjee et al:

Let Dependency Relation be the list of significant relations. We call any dependency relation significant, if

- It involves any subject, object or agent like nounSubject, dobject, agent etc
- It involves any modifier like adverbModifier, adjectiveModifier etc
- It involves negation
- It involves any adjectival or clausal component like clauseModifier

Dependency parsing gives us the relationship between words that can be exploited to generate features and their corresponding sentiment scores from potential features and sentiment scores respectively. Dependency parsing prunes the list of potential features and links them with the specific opinion word associated. The negation (explicitly) and conjunction (implicitly) handling is also done in this stage.

Example 1: **The Phone came yesterday and the display is not very good.**

After POS tagging we get: The(determinant) phone(noun) came(verb) yesterday(noun) and(conjunction) the(determinant) display(noun) is(verb) not(adverb) good(adjective).

Dependency Parsing using Stanford Parser(only relevant tags):

nounSubject(has, phone)
negative(good, not)
adjectiveModifier(display,great)
RelativeClauseModifier(performance, satisfactory)

The negation handling is done using the following algorithm:

```
if neg
    score=(1-score_of_opinion_word)
```

3) Example 2:: **The Phone has a great display and the performance is satisfactory**

After POS tagging we get: The(determinant) phone(noun) has(verb) a(determinant) great (adjective) display(noun) and(conjunction) the(determinant) performance(noun) is(verb) satisfactory(adjective).

Tuple Analysis: phone, display, performance; great, satisfactory

Dependency Parsing (only relevant tags):

nounSubject(has, phone)
adjectiveModifier(display,great)
RelativeClauseModifier(performance, satisfactory)

A combination of both tuple analysis and dependency parsing gives us the desired result. While the dependency parser identifies that the opinion word 'great' relates to display and that satisfactory relates to performance, it also identifies nounSubject(has, phone) which is irrelevant but is within potential relation tags. This irrelevant part is revealed using the POS tagger and pruned, as the relationship does not have a potential opinion and potential feature word. Hence, we get 2 relations: (display, great) and (performance, satisfactory).

D. Issues

This review based analysis technique has the potential to give a reasonably decent accuracy score, but will have low recall score because many sentences have their features mentioned implicitly as opposed to explicitly. Eg :-

It is bright

The system cannot recognize the reference to the screen, and hence will fail in such conditions. Similarly opinions that are not expressly stated will be overlooked. Eg :-

The phone held its own

While the phone is to get a positive polarity associated with it from this review, as the system does not recognize phrases- there is a failure in analysis. Phrase substitution [7] requires separate study to detail an effective method to determine polarity of phrases. This system therefore ignores phrase analysis.

Also, not all noun-adjective pairs are feature-opinion relations. This is the failure of the system.

E. Design Choices

N-gram extraction is a technique often used in review analysis. Since the reviews are so focused (single product) and the products used in analysis have but few features (unlike cars for example), the idea of using n-gram was dropped in favour of unigram extraction. This decision was made at evaluation because of duplicity of feature results like- display quality and screen clarity being classified as 2 different aspects. In the absence of an ontology (Section 4.1), the problem gets further compounded.

The system will recognize a feature only if a sentiment is associated with it. So the sentence: **"The Camera has a strap"** will not have strap recognized as a feature. The system

ISSN 2395-8618

TABLE I
EVALUATION OF FEATURE IDENTIFICATION.

Review Domain	Precision	Recall	F-measure	Comparable System
MP3 Player	0.71	0.82	0.87	0.64
Camera	0.60	0.83	0.697	0.60
Router	0.77	0.722	0.745	0.61
Portable Camera	0.69	0.72	0.70 8	0.70
Mobile Phone	0.69	0.76	0.72	0.66

TABLE II
EVALUATION OF FEATURE SCORING.

Review Domain	Accuracy
Video Player	0.69
Camera	0.83
Music Player	0.65
Portable Camera	0.77
Phone	0.79

is a feature based sentiment analysis system, and not a feature extraction system.

III. EVALUATION

The following formulae, from [11], will be employed for evaluation.

$$Precision = \frac{NumberofCorrect}{NumberofExtracted}$$

$$Recall = \frac{NumberofCorrect}{NumberofTrue}$$

$$F\text{-}measure = \frac{2 \times recall \times precision}{recall + precision}$$

$$Accuracy = \frac{CorrectOfQueries}{TotalQueries}$$

A. Feature Identification

The evaluation is done with the use of 'ground truths' for correct and incorrect because evaluation is done based on human perception and hence ranked as such, as opposed to clear mathematical precision of right or wrong.

5 corpus of reviews,taken from [5], belonging to different products were used to evaluate the system. The results of the system are shown below in table 1. The Comparable System refers to the results obtained by Subhabrata Mukherjee [3] using the same review corpus.

B. Sentiment Assignment to each Feature

The sentiment assignment forms the largest part of the proposed system. This section tests the proposed identification of opinion words and their corresponding polarity score.

Table 2 gives us the polarity classification correctness for each identified feature. The accuracy for each product would be higher if there is specific domain that the training set is from. So, a system well trained on mobile corpus positive / negative examples can be more effective in scoring a mobile domain corpus set.

IV. FUTURE WORK

A. Ontology

Using a domain knowledge system will improve feature identification. After a domain specific system is built, we can be sure that junk features will be discarded. On the other hand, features and their synonyms are also available to the analysis system to exploit. For example:

Worth the money

It is important to understand that the feature identification is linked in with the sentiment identification- that is in the absence of an associated opinion word- the sentiment system fails to identify the potential feature as a feature.

Worth the cost

Both cost and money are synonyms. But, in the absence of an ontology, both the words will be considered separate features. Much like the sentiment scores, the ontology must be generated prior to using the system for analysis, and stored for later use.

B. Status Array

Many researchers have remarked about the inability of existing systems to identify sarcasm [12]. This is a valid concern, and addressing this problem with an effective solution could help improve analysis by a great deal because angry reviewers often resort to sarcasm in their reviews.

Another issue is related to cross referencing nouns in sentence level analysis. For example: "The Speaker is great. It is loud." If the system knew that the noun in context was the 'speaker', it would have made an accurate classification that the speakers are loud.

To solve these problems, a review status array would be well suited. It could have multiple elements, but to simply deal with the two problems mentioned above:

```
<previous_noun;
previous_sentiment_polarity>
```

If no noun is identified in a sentence, the previously used (feature) noun would be used as the feature. Such a method implements continuity among the different sentences of a review.

Example for sarcasm handling:

The battery is great. It blew up on the second day.

Status Array for the above example,

```
<battery, positive>
```

Since, the second sentence is of negative polarity and references back to the previous noun, the system inverts the positive score assigned to the feature previously.

ISSN 2395-8618

C. Domain Pertinence

A potentially useful tool to filter out bad feature results would be using domain pertinence filter [13]. The same noise words are quite often found to populate multiple domains without belonging to one or the other. For example, the feature 'person' might be identified in both the agriculture and computer domain. In order to clean, we use the other domain's identified features as a filter.

V. Conclusion

Evaluation metrics would change dependent on the changes in tagging or parsing algorithm as well as the dataset used for training. The f-measure would be higher if the polarity training set was more relevant to the domain being analysed. The additions mentioned in the future work section could potentially give a significant improvement over the current system.

The outlined system could be used as a base upon which further improvements can be made. There is often a choice for the person building analysis system- he/she can opt for higher recall, and lower precision, or vice-versa. The designer will have to study the domain and requirements to achieve the targets for the project by striking a balance between recall and precision.

References

[1] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[11] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[2] B. Agarwal, N. Mittal, P. Bansal, and S. Garg, "Sentiment analysis using common-sense and context information," *Computational intelligence and neuroscience*, vol. 2015, p. 30, 2015.

[3] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 475–487.

[4] A. Collomb, C. Costea, D. Joyeux, O. Hasan, and L. Brunie, "A study and comparison of sentiment analysis methods for reputation evaluation."

[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[6] B. Ohana and B. Tierney, "Sentiment classification of reviews using SentiWordNet," in *9th. IT & T Conference*, 2009, p. 13.

[7] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[8] M. Radovanović and M. Ivanović, "Text mining: Approaches and applications," *Novi Sad J. Math*, vol. 38, no. 3, pp. 227–234, 2008.

[9] M.-C. de Marneffe and C. D. Manning, "The stanford typed dependencies representation," in *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, ser. CrossParser '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 1–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1608858.1608859

[10] M. Ganapathibhotla and B. Liu, "Mining opinions in comparative sentences," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 241–248.

[12] K. Vivekanandan and J. S. Aravindan, "Aspect-based opinion mining: A survey," *International Journal of Computer Applications*, vol. 106, no. 3, 2014.

[13] J. De Knijff, F. Frasincar, and F. Hogenboom, "Domain taxonomy learning from text: The subsumption method versus hierarchical clustering," *Data & Knowledge Engineering*, vol. 83, pp. 54–69, 2013.

ISSN 2395-8618

One Sense per Discourse Heuristic for Improving Precision of WSD Methods based on Lexical Intersections with the Context

Grigori Sidorov and Francisco Viveros-Jiménez

***Abstract*—Word sense disambiguation is the task of choosing a sense for a target word in a given text using some words from the text and, in some cases, hand-tagged samples or dictionary definitions. The sense list is taken usually from an explanatory dictionary for a given language. Note that since the word is part of the text, we rely on the context words for making the decision. The methods that use information from words in the (near) context are very simple, because they consider lexical intersections of the word with the context words and/or their definitions or samples of usage. These methods reach precision of up to 70%. There are also methods that have better performance, but they are much more sophisticated: they use expensive resources – usually hand crafted – and rely on complex algorithms. In this paper, we show how to increase precision for certain word classes of these simple methods to the level comparable with that of the most sophisticated ones. Namely, we observed that these methods usually disambiguate correctly those words that conform to the One Sense per Discourse heuristic (OSD words). We used Semcor and Wikipedia to find the OSD words and left non-OSD words without disambiguation, thus improving precision at the expense of recall. Our motivation for this situation – more precision, less recall – is: (1) if we need high quality disambiguation and use human evaluators, then we can reduce the cost by asking them to disambiguate only words that are really difficult for the algorithms; (2) in an automatic system, we can apply this method for disambiguation of the corresponding words, and use other more sophisticated method for disambiguation of other words, i.e., use different methods for disambiguation (meta-disambiguation). We experimented with the complete and simplified Lesk algorithms, the graph based algorithm, and the first sense heuristic. The precision of all algorithms increases and some algorithms reach the level of the inter annotator agreement.**

***Index Terms*—Word sense disambiguation, one sense per discourse heuristic, context, lexical intersections.**

I. Introduction

WORDS have different meanings depending on the context. For example, in the sentence "John is drawing a **tree**", the last word can mean a plant or a graph. Word sense disambiguation (WSD) is the task of identifying the sense (meaning) of a target word in a context [1]. Word senses are taken from a specific explanatory dictionary.

Manuscript received on March 15, 2016, accepted for publication on September 30, 2017, published on June 30, 2018.

Grigori Sidorov and Francisco Viveros-Jiménez are with the Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, Mexico (web: www.cic.ipn.mx/ sidorov, e-mail: pacovj@hotmail.com).

Generally speaking, WSD is a complex problem, which may require for its solution application of various methods of artificial intelligence. Currently, there are numerous solutions for tackling WSD. Simple methods that are based on the knowledge about the word itself and the words in its context have relatively low performance (the best methods obtain precision of about 60%). More complex supervised methods can reach precision above 70% [2], [3]. Still, these supervised methods need manually tagged training data, which is expensive and in real life is not always affordable.

WSD is useful for many NLP applications that deal with the meaning of texts, such as machine translation [4], [5], [6], wikification [7], information retrieval [8], etc. So there is a need in WSD systems with high precision when designing systems for these tasks. Note that a need in a reliable WSD system persists even if such system disambiguates only some target words (i.e., not all of them).

This paper describes the method that allows increasing precision of WSD systems at the expense of recall/coverage. The main idea is to disambiguate just those words that comply with the one sense per discourse (OSD) heuristic. Further this idea is analyzed in detail.

Previously, it was reported [9] that using features for selective disambiguation leads to a performance boost of about 5%. In that work the authors used word features, such as word grain, amount of positive and negative training examples and dominant sense ratio. They went even further and ensemble a back-off chain of three methods in a metaheuristic that selects the best method (of the three) using these word features. We propose to rely only on the One Sense per Discourse heuristic, but our precision boosts are greater than the ones reported by [9].

Our motivation for this situation – more precision, less recall– is: (1) if we need high quality disambiguation and use human evaluators, then we can reduce the cost by asking them to disambiguate only words that are really difficult for the algorithms; (2) in an automatic system, we can apply this method for disambiguation of the corresponding words, and use other more sophisticated method for disambiguation of

ISSN 2395-8618

other words, i.e., use different methods for disambiguation (meta-disambiguation).

In the following sections, we describe the corresponding experiments and present a discussion about the behavior of the proposed method.

II. Experimental Setup

As we already mentioned, our hypothesis is that disambiguating only OSD words increases precision, but it obviously disambiguates fewer words, so recall and coverage become lower. Experiments were conducted for confirming this hypothesis. We show that this cost is acceptable, i.e., the WSD systems benefit from the proposed method. We also show that this phenomenon does not depend on the disambiguating algorithm, the test set data, and the word sense inventory.

We tested four algorithms over four test sets using different explanatory dictionaries (sense inventories). The explanatory dictionaries were: the WordNet 3.0 [10], Wikipedia [11] and Spanish Wikipedia. The test sets were: Senseval 2 [2], Senseval 3 [3], and hand-picked English/Spanish Wikipedia articles. The used WSD algorithms were: the Simplified Lesk algorithm [12], the Graph Indegree with Lesk measure [13], the traditional Lesk algorithm [14] and the first sense heuristic [12]. We also present additional experiments with Conceptual Density [15], Naive Bayes [16] and GETALP [17], [18] for some test sets.

We used precision (P), recall (R), coverage (C) and F-measure (F1) for measuring the performance of the algorithms as specified in [1]. They were calculated with the following equations:

$$\begin{aligned} \mathrm{P} &= \frac{\text{correct answers}}{\text{answers}}, \\ \mathrm{R} &= \frac{\text{correct answers}}{\text{words}}, \\ \mathrm{C} &= \frac{\text{answers}}{\text{words}}, \\ \mathrm{F1} &= \frac{2 \times P \times R}{P + R}, \end{aligned}$$

where *answer* is the target word, for which the algorithm has selected a sense, and *correct answer* is the answer that coincides with the one provided by human annotators as the gold standard.

We compare the performance of the algorithms in three different situations:

1) Disambiguating each sentence independently.
2) Forcing the algorithms to use the one sense per discourse (OSD) assumption [19], [20]. In this case, the WSD algorithms disambiguate all instances of the target word independently using the corresponding sentences as the context.
3) Disambiguating only words that usually comply with the one sense per discourse heuristic (OSD).

A. Wikipedia Test Set

Besides using Senseval 2 and Senseval 3 data, we also chose 12 Wikipedia articles in Spanish and English languages as our empirical test set. Note that articles in Spanish Wikipedia are generally shorter and have less polysemy than their counterparts in English, as can be seen in Table I. These differences make disambiguating Spanish articles easier.

B. About our Implementation

We used Java as our main programming language and a computer with Intel Core I3 with 4GB in RAM for testing. Our Java implementation is available for using under a non-commercial license at *http://sourceforge.net/gannu*. It contains command line and graphical tools for performing the following tasks:

- Setting up the experiments.
- Running the experiments (test results are stored into XLS files).
- Searching for sense definitions.
- Creating gold standard files from raw text and Wikipedia articles.
- Loading samples into a dictionary.

The package also contains complete API documentation and tutorials.

C. Implementation Details Related with the WordNet

We used both glosses and samples as the base definitions. We discarded stop-words using the predefined list. The Stanford POS tagger [21] and the lemmatizer based on WordNet [10] were used for generation of the final definitions of words (word senses). Our test results differ from the reported results – less than $\pm 3\%$ in F1-measure – because we used the different – the latest – version of the WordNet. Note that our results can be easily reproduced, because the source code and the data are available.

D. Implementation Details Related with Wikipedia

We used the first paragraphs of Wikipedia definitions, which appear before the table of contents or a section mark of articles, as definitions of word senses. We used the manually inserted hyperlinks in the articles and the disambiguation pages as our gold standard. For example, if we want to disambiguate the word *Wolf*, a WSD system have to select between the 40 senses listed in the *wiki/Wolf_(disambiguation)* page. If this word is tagged with the hyperlink *wiki/Gray_wolf*, then we know that the correct sense is the one corresponding to this link.

III. Calculation of the One Sense per Discourse Condition

Some words very often comply with the one sense per document (OSD) heuristic, which tells us that these words

ISSN 2395-8618

TABLE I
SELECTED WIKIPEDIA ARTICLES AND SOME OF THEIR FEATURES.

	English			Spanish		
	Running	Target	Polysemy	Running	Target	Polysemy
Book	7152	406	7.9	3471	228	4.7
Calculator	6965	259	11.0	5837	202	1.8
Chemistry	8619	748	7.4	2849	223	3.7
Computer	9386	694	11.7	2885	195	3.0
Dog	15113	541	9.9	10161	366	4.7
Gray Wolf	16667	933	15.7	9438	289	6.1
Iron Man	11288	499	7.0	7371	225	3.3
Penicillin	4603	240	5.2	12087	692	2.9
Printing Press	6593	257	8.2	2492	86	4.9
Science	11123	753	10.4	17086	620	3.4
Spider-Man	9626	481	9.2	8519	239	4.5
Tiger	16744	639	13.1	4667	249	6.7
Average	**10323**	**537**	**10.4**	**7239**	**301**	**4.0**

usually have single meaning inside a text [19]. The OSD heuristic was successfully used for disambiguating some selected nouns in [20]. However, it was reported that not all words comply with this heuristic [22], so it is not a good idea to apply the OSD heuristic for all words (as we confirm later in this research).

We used two procedures for calculating if a word complies with the OSD heuristic or not. For WordNet based tests, we used the SemCor corpus [23]. A word complies with the OSD heuristic if it appears in this corpus with exactly one sense per document or it does not exist in the corpus.

For Wikipedia based tests, we used Wikipedia search counts. These search counts are stored in a matrix containing the search hits of all possible pair of senses, i.e. , our algorithm searches for the frequency of co-occurences of senses. Thus, each matrix element stores the frequency of a pair of senses. Diagonal elements contain the search counts of single senses. All counts are decreased by an empirical value of $2\times Polysemy$ hits due to the existence of disambiguation, category and list pages which contain sense pairs of the same word. A word complies with the OSD heuristic when all of the non-diagonal element of the matrix are less or equal to zero.

IV. PERFORMANCE ANALYSIS

Test results presented in Tables II, III, and V confirm that disambiguating just the words that comply with the OSD heuristic increases precision at the cost of recall/coverage. The precision boost was from 3% to 25%, being the average of 16%. The coverage loss was from 11% to 57%, being the average of 34%. Also, we observed that the first sense heuristic together with the OSD heuristic had the best approach in the tests: it obtained precision in the range from 79% to 99%. Note that forcing the OSD heuristic assumption does not lead to a consistent increase in precision (although, it often leads to a coverage boost). For further reference we added a table containing the best results observed in Senseval 2 and Senseval 3, see Table IV. Note that the first sense heuristic together with the OSD heuristic overcame the precision of the best systems in these competitions.

Figure 1 provides a graphical representation of the performance changes. This figure shows the precision obtained for different values of coverage. Coverage can be changed by using different window sizes in the range of [1,1024] (i.e., window size values are directly related to coverage). This figure confirms that all selected algorithms solve better the OSD words. Hence, our OSD filter allows other algorithms to outperform first sense heuristic. Moreover, the Graph InDegree algorithm was able to get the value of the inter annotator agreement for the OSD words.

TABLE II
TEST RESULTS CORRESPONDING TO GETALP SYSTEM OBSERVED IN SEMEVAL 2013 AND SENSEVAL 3 COMPETITIONS.

Semeval 2013 [18]	P	R	C	F1
GETALP with OSD	65.7	37.9	57.6	48.1
GETALP	51.6	51.6	100	51.6

V. WORDS THAT DO NOT COMPLY WITH THE OSD HEURISTIC

We analyzed the words that do not comply with the OSD heuristic. There are words of all grammar classes, as seen in Table VII. The amount of such words is in the range from 14% to 58%. Most of the words that comply with the OSD heuristic are domain words like *scientist, cell, cancer, strategy, treatment*, etc.

Words that do not comply with the OSD heuristic have at least one of these traits:

- their sense definitions are similar between themselves,
- their sense definitions have very few (usually, less than three) open-class words, and
- their meaning is related to their current syntactic functions rather than to a possible document domain.

ISSN 2395-8618

TABLE III
TEST RESULTS CORRESPONDING TO CONCEPTUAL DENSITY AND NAIVE BAYES ALGORITHMS OBSERVED IN SENSEVAL 2 AND SENSEVAL 3 COMPETITIONS.

	Senseval 2				Senseval 3			
	P	R	C	F1	P	R	C	F1
Conceptual Density with OSD	**57.1**	5.8	10.0	10.5	**64.7**	13.4	20.7	22.2
Conceptual Density	49.2	9.7	19.8	16.2	51.7	34.8	67.2	41.6
Naive Bayes with OSD	**73.7**	36.0	48.9	48.3	**74.5**	30.6	41.1	43.4
Naive Bayes	58.4	57.0	97.6	57.7	54.9	54.2	98.9	54.6

TABLE IV
SYSTEMS HAVING THE HIGHEST PRECISION IN SENSEVAL 2 AND SENSEVAL 3 COMPETITIONS.

Senseval 2	P	R	C	F1
First Sense with OSD	**78.8**	40.0	50.9	53.1
IRST [24]	74.8	35.7	47.7	48.3
SMUaw [25]	69.0	69.0	100	69.0
CNTS-Antwerp [26]	63.6	63.6	100	69.0
Senseval 3				
First Sense with OSD	**79.3**	33.1	42.3	46.5
IRST-DDD-09-U [27]	72.9	44.1	60.5	54.9
IRST-DDD-LSA-U [27]	66.1	49.6	75.0	56.6
Gambl-AW-S [28]	65.1	65.1	100	65.1

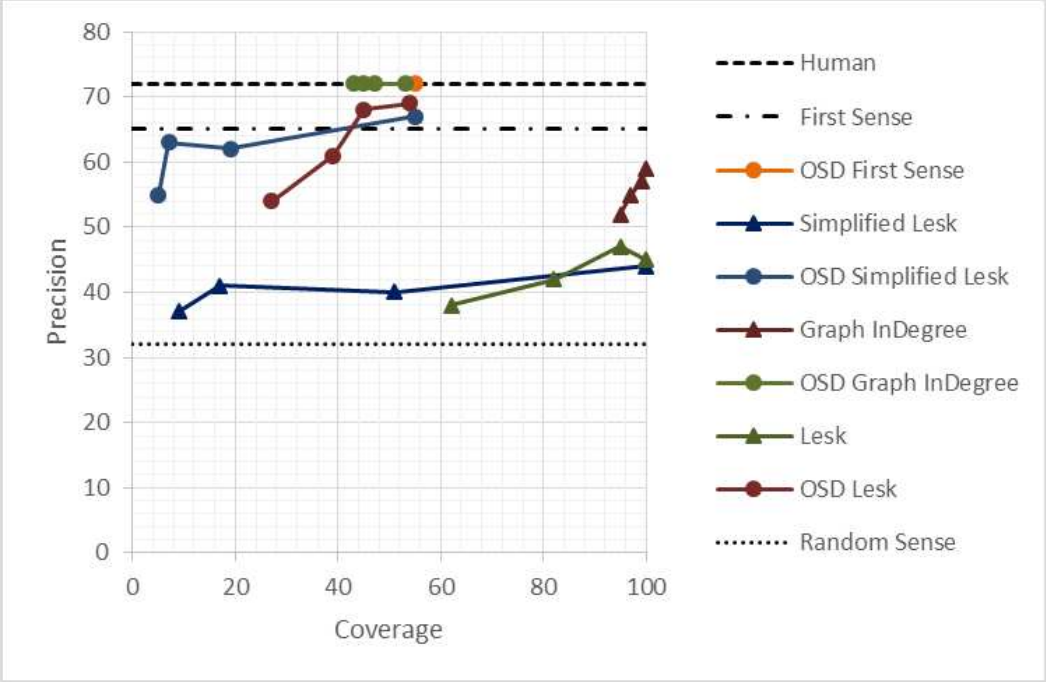


Fig. 1. Precision/coverage graph for some knowledge-based algorithms observed on Senseval 2 test set. Algorithms using our OSD filter (circles) overcame the first sense heuristic precision. Also, some algorithms overcame the human annotator agreement.

Table VI contains some sample definitions that are too similar to distinguish between them or too short for WSD systems.

The most discarded words are verbs. Common verbs (like *be, have* and *do*) have more than ten definitions in WordNet and are used widely across all domains. Often the main part of the meaning of verbs is heavily related to its complements. Take for example the following text: "*I started drinking some soda. Later, I decided to drink a cold beer.*" and the following definitions [$drink^1_V$:take in liquids] and [$drink^2_V$:consume alcohol] extracted from the WordNet. In this example, both definitions are clear for people but they are rather short for WSD algorithms. Also, the verb *drink* does not comply with the OSD heuristic. Furthermore, we can easily select the sense

ISSN 2395-8618

TABLE V
PERFORMANCE COMPARISON OF SOME BAG OF WORDS ALGORITHMS. ALL METHODS EXHIBIT A PRECISION BOOST AND A COVERAGE LOSS WHEN SOLVING JUST WORDS THAT COMPLY WITH THE OSD HEURISTIC. **ST** MEANS DISAMBIGUATION USING SENTENCE WORDS, **FG** MEANS FORCING OSD FOR ALL WORDS, AND **OSD** MEANS SOLVING JUST WORDS THAT COMPLY WITH THE OSD HEURISTIC.

	First sense			Simplified Lesk			Graph InDegree			Lesk		
Senseval 2												
	St	Fg	OSD	St	Fg	OSD	St	Fg	OSD	St	Fg	OSD
P	67.1	67.1	78.8	39.5	45.5	61.0	59.7	57.5	78.1	48.1	49.4	67.6
R	67.1	67.1	40.0	19.6	31.0	12.1	59.6	57.4	39.9	46.0	49.4	31.9
C	100	100	50.9	49.7	68.1	19.8	99.8	99.9	51.1	95.6	99.9	47.2
F1	67.1	67.1	53.1	26.2	36.8	20.2	59.6	57.4	52.9	47.0	49.4	43.3
Senseval 3												
P	66.1	66.1	79.3	30.3	30.9	52.7	50.5	51.1	70.1	38.4	41.0	64.3
R	66.1	66.1	33.1	19.5	23.3	10.4	50.1	50.9	29.5	36.7	40.8	24.3
C	100	100	42.3	64.4	75.5	19.8	99.2	99.7	42.1	94.2	99.4	37.8
F1	66.1	66.1	46.5	23.7	26.6	17.4	50.3	51.0	41.5	37.8	40.9	35.2
English Wikipedia												
	First sense			Simplified Lesk			Graph InDegree			Lesk		
	St	Fg	OSD	St	Fg	OSD	St	Fg	OSD	St	Fg	OSD
P	89.5	89.2	95.8	71.5	73.5	92.3	72.5	72.9	92.0	70.3	68.8	92.9
R	89.5	89.2	68.5	57.6	62.6	50.1	66.7	69.2	58.8	49.9	65.8	46.3
C	100	100	71.5	80.6	85.1	54.2	92.1	94.9	63.9	71.0	95.5	49.9
F1	89.5	89.2	79.9	63.8	67.6	64.9	69.5	71.0	71.8	58.4	67.2	61.8
Spanish Wikipedia												
P	96.3	96.3	99.6	87.2	87.4	98.7	87.0	87.0	98.5	85.3	84.8	98.1
R	96.3	96.3	85.5	76.1	79.2	72.5	80.3	82.2	76.9	59.4	66.6	57.2
C	100	100	85.8	87.2	90.6	73.4	92.4	94.4	78.1	69.7	78.6	58.3
F1	96.3	96.3	92.0	81.3	83.1	83.6	83.5	84.5	86.4	70.0	74.6	72.3

TABLE VI
SOME DEFINITIONS THAT ARE TOO SIMILAR (TOP) OR SHORT (BOTTOM).

$Medical^1_J$	relating to the study or practice of medicine
$Medical^2_J$	requiring or amenable to treatment by medicine as opposed to surgery
$Bell^5_N$	the shape of a bell
$Recent^1_J$	new

TABLE VII
AVERAGE WORDS DISCARDED OF EACH CLASS.

	Noun	Verb	Adjective	Adverb
Senseval 2	39%	74%	43%	50%
Senseval 3	47%	81%	38%	0%
English Wiki	28%	–	–	–
Spanish Wiki	14%	–	–	–

of the verb *drink* by looking at the direct object in both cases. It is typical lexical function. Hence, in our future research we will try to design a system for disambiguating these words by using syntactic information.

VI. CONCLUSIONS

This paper shows that WSD methods can attain high precision when solving just those words that comply with one sense per discourse heuristic at the cost of losing recall/coverage. The precision boost is high enough to overcome the first sense baseline: this achievement can be reached only by complex state-of-the-art WSD systems. Also, our experimental results show that words that do not comply with OSD have one of these traits: (1) their meaning depends on the sentence rather than the domain (like most of the verbs), and, (2) their sense definitions are not adequate for current systems (they are too short or too similar between them).

We recommend disambiguating just the OSD words for increasing precision of WSD algorithms for real life applications requiring high precision.

REFERENCES

[1] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv*, vol. 41, no. 2, pp. 10:1–10:69, 2009.
[2] M. Palmer, C. Fellbaum, S. Cotton, L. Delfs, and H. T. Dang, "English tasks: All-words and verb lexical sample," in *Proc. of ACL/SIGLEX Senseval-2*, 2001.
[3] B. Snyder and M. Palmer, "The English all-words task," in *Proc. of ACL/SIGLEX Senseval-3*, 2004.
[4] Y. S. Chan and H. T. Ng, "Word sense disambiguation improves statistical machine translation," in *Proc. of ACL 2007*, 2007, pp. 33–40.
[5] M. Carpuat, Y. Shen, X. Yu, and D. Wu, "Toward Integrating Word Sense and Entity Disambiguation into Statistical Machine Translation," in *Proc. of IWSLT*, 2006, pp. 37–44.

ISSN 2395-8618

[6] D. Pinto, C. Balderas, M. Tovar, and B. Beltran, "Evaluating n-gram models for a bilingual word sense disambiguation task," *Computación y Sistemas*, vol. 15, no. 2, 2011.

[7] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proc. of CIKM 2007*, 2007, pp. 233–242.

[8] C. Stokoe, M. P. Oakes, and J. Tait, "Word sense disambiguation in information retrieval revisited," in *Proc. of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003, pp. 159–166.

[9] H. M. Saarikoski, S. Legrand, and A. Gelbukh, "Defining classifier regions for WSD ensembles using word space features," in *MICAI 2006: Advances in Artificial Intelligence*, 2006, pp. 885–867.

[10] G. A. Miller, "A lexical database for English," *Communications of the ACM*, vol. 38, pp. 39–41, 1995.

[11] Wikipedia, "Wikipedia: The free encyclopedia," 2004.

[12] A. Kilgarriff and J. Rosenzweig, "Framework and results for English SENSEVAL," *Computers and the Humanities*, vol. 34, no. 1-2, pp. 15–48, 2000.

[13] R. Sinha and R. Mihalcea, "Unsupervised graph-based word sense disambiguation using measures of word semantic similarity," in *Proc. of ICSC 2007*, 2007, pp. 363–369.

[14] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proc. of SIGDOC*, 1986, pp. 24–26.

[15] D. Buscaldi, P. Rosso, and F. Masulli, "Finding predominant word senses in untagged text," in *Workshop Senseval-3, Proc. of ACL*, 2004, pp. 77–82.

[16] D. Yuret, "Some experiments with a Naive Bayes WSD system," in *Proc. of ACL/SIGLEX Senseval-3*, 2004.

[17] D. Schwab, J. Goulian, and A. Tchechmedjiev, "Worst-case complexity and empirical evaluation of artificial intelligence methods for unsupervised word sense disambiguation," *International Journal of Web Engineering and Technology*, vol. 8, no. 2, pp. 124–153, 2013.

[18] R. Navigli, D. Jurgens, and D. Vannella, "Semeval-2013 task 12: Multilingual word sense disambiguation," in *Proceedings of the 7th International Workshop on Semantic Evaluation*, 2013, pp. 222–231.

[19] W. A. Gale, K. W. Church, and D. Yarowski, "One sense per discourse," in *Proc. of HLT*, 1991, pp. 233–237.

[20] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. of ACL 2007*, 1995, pp. 189–196.

[21] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Proc. of EMNLP*, 2000, pp. 63–70.

[22] D. Martínez and E. Agirre, "One Sense per Collocation and Genre/Topic Variations," in *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.

[23] G. A. Miller, M. Chodorow, S. Landes, C. Leacock, and R. G. Thomas, "Sing a semantic concordance for sense identification," in *Proc. of ARPA Human Language Technology Workshop*, 1994, pp. 240–243.

[24] B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo, "Using domain information for word sense disambiguation," in *Proc. of ACL/SIGLEX Senseval-2*, 2001.

[25] R. Mihalcea and D. Moldovan, "Pattern learning and active feature selection for word sense disambiguation," in *Proc. of ACL/SIGLEX Senseval-2*, 2001.

[26] V. Hoste, A. Kool, and W. Daelemans, "Classifier optimization and combination in the english all words task," in *Proc. of ACL/SIGLEX Senseval-2*, 2001.

[27] C. Strapparava, A. Gliozzo, and C. Giuliano, "Pattern abstraction and term similarity for word sense disambiguation: IRST at senseval-3," in *Proc. of ACL/SIGLEX Senseval-3*, 2004.

[28] B. Decadt, V. Hoste, W. Daelemans, and A. Van den Bosch, "Genetic Algorithm Optimization of Memory-Based WSD," in *Proc. of ACL/SIGLEX Senseval-3*, 2004.

ISSN 2395-8618

TrazasBP: A Framework for Business Process Models Discovery Based on Execution Cases

Hugo Ordoñez, Armando Ordóñez, Victor Buchelli, and Carlos Cobos

***Abstract*—Execution of business processes generates data, which are commonly recorded in logs. Historical information of execution cases may be used for recommending future execution paths. This is useful when the control flow of the process is not known by the user. We present TrazasBP, a framework for BP indexing and searching based on execution cases. It indexes BPs based on execution cases (traces) retrieved from log files. TrazasBP not only takes into account the textual information of BP elements, but also the causal dependence between these elements. Furthermore, due to its low computational cost, TrazasBP may be used as indexing mechanism in order to reduce the search space. Experimental evaluation shows promising values of graded precision, recall and F-measure when compared with results obtained from human search.**

***Index Terms*—Business process, execution cases, Logs, repository, evaluation**

I. Introduction

INNOVATON in products and services is mandatory for competitiveness in today's market. Commonly, tasks and functions related to commercial activities of companies are represented within Business processes (BP) [1]. A BP consists of a set of logically-related tasks executed sequentially in order to generate valid outputs for the business. BP executions must follow guidelines given by internal policies, standards, best practices and laws. For example, doctors should only perform surgeries within the scope of their specialty area. Furthermore, this surgery should be preceded by an authorization from the patient and the hospital. Another example, in sales processes, an order should be archived only after customer confirms reception of ordered items [2].

Manuscript received on October 25, 2017, accepted on February 27, 2018, published on June 30, 2018.

H. Ordonez is with the Research Laboratory in Development of Software Engineering of the Universidad San Buenaventura, Colombia (e-mail: haordonez@usbcali.edu.co).

A. Ordonez is with the Foundation University of Popayan, Colombia (corresponding author, e-mail: jaordonez@unicauca.edu.co).

V. Buchelli is with the Foundation University of Valle, Colombia (e-mail: victor.bucheli@correounivalle.edu.co).

C. Cobos is with the Foundation University of Cauca, Colombia (e-mail: ccobos@unicauca.edu.co).

Execution of tasks and processes generates a set of data which is recorded in logs [3]. Logs contain information of executed processes, namely: roles, resources, participants, interaction with other systems, transactions performed and execution dates, among other data. When a BP is initiated, an instance (execution) of the BP is created, therefore Logs store information of many instances or executions of the same process [4]. Specifically, historical execution traces containing information of actually executed instances are known as *execution cases*. These execution cases contain information of the path followed by the control flow during actual execution of a BP instance [5].

This paper presents TrazasBP, a framework for BP indexing and searching based on execution cases. TrazasBP indexes BPs based on execution cases (traces) retrieved from log files. A log file is created when a BP is executed for the first time, and it is updated by adding new execution cases as executions are carried out. Execution cases register information about a specific BP execution (i.e. what activities were executed at a certain moment in time during BP execution) [6]. Thus, a BP contains only one log file, but multiple execution cases included in this file. TrazasBP considers in addition to textual information of BP elements, the causal dependence between these elements. Furthermore, due to its low computational cost, TrazasBP may be used as indexing mechanism in order to reduce the search space.

The main contributions of TrazasBP are twofold: i) it may be used for indexing generation based on the execution cases, and ii) TrazasBP allows ranking a set of executed BPs in concordance with their similarity with a query BP. Historical information of *execution cases* may be used for recommending future execution paths. This is useful when the control flow of the process is not known by the user, for example, when the doctor doesn't know about new treatments or when a company cannot foresee the behavior of potential customers [7].

The rest of the paper is organized as follows: Section 2 presents related works, section 3 describes TrazasBP architecture, Section 4 shows the evaluation and results, finally, conclusions and the implications of the results are given in Section 5.

ISSN 2395-8618

II. Related works

Several approaches for measuring similarity of BP are available in the literature. These approaches are based on different BP characteristics such as: linguistics [8], [9], structure [10], [11] and behavior [12], [13]. As the approach presented here considers similarity of execution cases, therefore this section study approaches considering causal dependency between activities, and common sets of execution traces

Bae et al [14] present a dependency graph to compare two BPs taking into account differences between arcs or edges that links activities in both BPs. This approach does not consider gateway types. Weidlich et al [15] define causal behavioral profiles representing dependencies between activity pairs. Similarity is calculated by identifying activity pairs for which there are corresponding pairs of activities. Then corresponding pairs sharing the same relations are analyzed.

Dijkman et al [16] represent precedence relations between activities as loopback links and causal footprints. Causal footprints are in turn represented as vectors of index terms. This approach builds vectors of high dimension, which increase the computational cost of the method. Other existing approaches consider direct precedence of activities represented as Transition Adjacency Relation [17], n-grams [18], and behavioral profiles [15]. In those cases similarity is calculated analyzing correspondence between direct

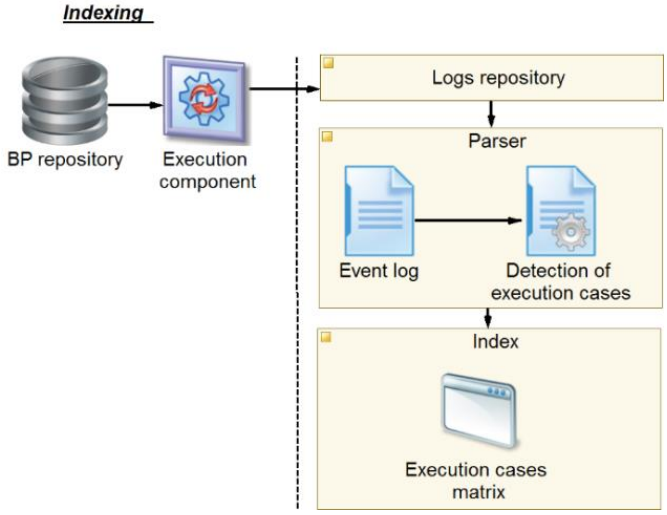


Fig. 1. Components of the indexing phase

precedence of activities in the trace.

Gerke et al [19] , Wang et al [20] compare the compliance between BPs calculating the longest common subsequence of traces, i.e., similarity degree of ordering rules of activities between two BPs. However, this approach is computational expensive when there are large sets of traces. Weerdt et al [6] deal with real execution traces of BPs in order to discover BPs, i.e., this method aims for inferring which BPs can produce such traces. Medeiros et al [9] compare BPs by studying frequency of traces obtained from actual or simulated executions.

TrazasBP integrate characteristics of the aforementioned approaches by considering causal dependence between activities of actual execution cases. Additionally, Due to TrazasBP low computational cost, it may to be used as indexing mechanism preceding other expensive algorithms during BPs similarity calculation; the last is possible because TrazasBP reduces the search space. Next section presents the architecture of TrazasBP and describes its components.

III. Architecture of trazas BP

TrazasBP allows indexing and searching BPs stored in a repository according to their similarity with a query represented as a set of node pairs (PS_q). Three kinds of query options are supported: execution cases, minimal behavior, and log-files (These types of queries are detailed later). TrazasBP works both during indexing phase and querying phase. During indexing phase (See Figure 1), logs are indexed and a matrix *Mec* of execution-cases is generated. Then, during querying phase (See Figure 3), a query (Execution case, minimal behavior, or log-file) is received and processed in order to obtain a set of node pairs. Finally, when the set of node pairs are obtained, the query matrix is generated *Mq* and the repository is ranked. Next both phases are described

3.1 Indexing phase

3.1.1 BP Repository

This repository stores a set of BPs, these BPs are executed in order to generate the execution cases. The current implementation of the repository includes 100 BPs modeled with BPMN (Business Process Modeling Notation). Those BPs were graphically designed by experts of the Telematics Engineering Group of the University of Cauca (Colombia) based on real processes provided by Telco operators in Colombia and examples found in different web sites (e.g the TM Forum2). A real repository of a Telco operator couldn't be used due to privacy and security policies of Telecom operators.

3.1.2 Execution component

This component executes BPs and collects log-files containing execution cases. The current version of this component is implemented using the Bizagi BPM suite which is a popular tool for BP modeling [19]. BPs were executed in the lab (in order to simulate real executions) and log-files were then stored in a second repository named "logs repository".

ISSN 2395-8618

3.1.3 Log Repository

This repository stores all the log-files obtained from the execution of BPs. Each BP contains only one log-file with multiple execution cases. The current implementation of this repository stores the log-files in the file system.

3.1.4 Parser

This component extracts and processes execution cases from each log-file stored in the "*logs repository*". Here, execution cases are represented as vectors (*execution case vectors*) that associate execution cases with *BPs*. Afterwards, each execution case vector is processed to form pairs of adjacent nodes in order to keep causal relationships. Once node pairs are formed, they are arranged together with node pairs of other execution cases in the same *BP* in order to create a new vector (*node pairs vector*). This procedure is repeated for the entire *BP* repository obtaining one vector of node pairs for each *BP*.

3.1.5 Index

This component processes node pairs vectors and generates an index. First at all, node pairs from each vector are analyzed with the Porter Stemming[10] algorithm that transforms node labels to their lexical root (e.g. words "helping" and "helped" are transformed to their lexical root "help"), later the same algorithm removes special characters, void words, and accents. Next, the indexer creates a "matrix of execution cases" (*Mec*) whose rows are the BPs stored in the repository, and the columns are the node pairs of all the BPs of the repository but avoiding the pairs that are duplicated. The matrix *Mec* is filled by counting the number of times that a pair is found in each BP (i.e. in the vector of node pairs of each BP).

	p_1	p_2	...	p_j	...	p_k
BP_1	0	0	...	2	...	1
BP_2	3	0	...	0	...	0
...	...	...	...	..	...	...
BP_i	2	0	...	3	...	2
...	...	...	...	..	...	...
BP_n	1	5	...	0	...	3

Fig. 2. Example of execution cases matrix

Let $R = \{BP_1, ..., BP_i, ...BP_m\}$ be a repository of BPs. Each $BP_i \in R$ contains a log file $l_i = \{ec_{i1}, ..., ec_{ij}, ...ec_{ik}\}$ that is updated each time the BPi is executed by adding a new execution case ecij . Each execution case is composed of a sequence of BP elements (nodes) which may be activities or gateways (XOR (Join-Split), AND(Join-Split)). These elements are ordered according to the execution flow followed by the BP. The first step in the BP indexing mechanism is to collect all the nodes of each execution case ecij = {n1, ..., np} and form pairs of nodes keeping their causal dependence (i.e., adjacent nodes in the execution case). For example, in ecij the set of node pairs is PSij = {(n1 , n2), (n2 , n3), ..., (ni−1 , ni),(ni , ni+1), ..., (np−1 , np)}.

After collecting pairs of the execution cases of the entire repository, a matrix named "execution cases matrix" Mec is created. In this matrix columns represent node pairs found in the execution cases for the entire repository but avoiding those repeated (i.e. there are not two columns representing identical node pairs), and rows are all the BPs stored in the repository. Therefore, the size of the matrix is m × k, where m is the number of BPs in the repository, and k is the number of all pairs found in execution cases avoiding those which are repeated.

Finally, the matrix *Mec* is completed with the number of times a pair is found in the execution cases of a given *BP*, e.g., if a pair p_j is found three times in the log $l_i \in BP_i$, then the number 3 is inserted on the cell (i; j) (Figure 2). Thus, the index of execution cases is created and represented by the matrix *Mec*. In the present approach, this matrix is similar to the "term-document matrix" of the vector space model in the Information Retrieval (IR) field proposed by Salton in 1989. Therefore, the *Mec* matrix can be normalized in the same way as the "term-document matrix", which is composed of cells w_{ij} representing textual components (in their lexical root) detected in a log-file. Then, each w_{ij} is weighted with the equation 1, where F_{ij} is the observed frequency in the component *j* of the BP_i; $Max(F_i)$ is the highest observed frequency of the BP_i; *N* is the number of *BP* in the repository; and n_j is the number of *BP* in which the execution case *j* has been detected

$$w_{i,j} = \frac{F_{i,j}}{\max(F_i)} \times \log\left(\frac{N}{n_j + 1}\right) \quad (1)$$

3.2 Query phase

First This phase has two functions: firstly, it transforms queries into node pairs in order to create a query matrix (*Mq*). *Mq* contains information about the frequency of each pair in the query. Secondly, this phase ranks BPs according to their similarity to the query. (See Figure 3)

3.2.1 Query Processor

This module receives a query and transforms it into a set of node pairs. The current implementation of the query module supports the following 3 kinds of queries:

Execution case: the query is a textual string that represents a BPs execution case. Therefore, the string must contain a sequence of nodes (activities and gateways) that will be transformed into a set of node pairs.

Minimal behavior of execution flow: the query is a list of node pairs obtained from the execution cases of the BPs in the

ISSN 2395-8618

repository. Then a user can choose a combination of node pairs to build a query.

Log file: the query is a log-file that is processed to identify the execution cases and subsequently the sets of node pairs. In this option, the user can choose one of the found sets of execution cases in order to rank the BPs in the repository that have executed similar execution cases. Once, the query is transformed, the set of node pairs are processed with the "PorterSteeming" algorithm as explained before. Then, the duplicated pairs are counted and inserted in a query vector which contains the number of occurrences of each pair.

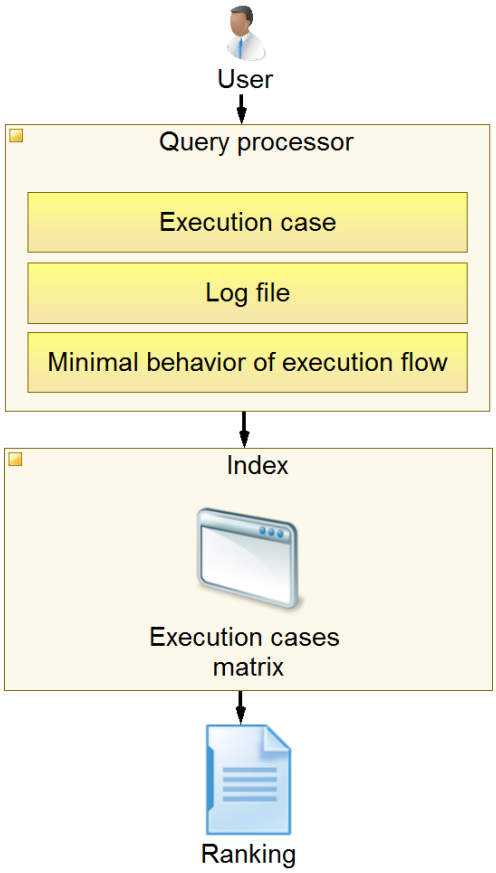


Fig. 3. Components of the indexing phase

3.2.2 Ranking

In this phase the query vector vq and the execution cases matrix Mec are integrated in the query matrix (Mq) as described in section 3.23. The Mq matrix is useful for measuring similarity between each BP of the repository and the query and to produce a ranking of BPs according to this degree of similarity.

3.2.2 Querying the index of execution cases

To query the index of execution cases a query set of node pairs (PS_q) is required. The set PS_q is processed in order to find repeated node pairs, and to create a query vector *vq* that registers the number of occurrences of each pair. For example, let $PS_q = \{p_{q1}, p_{q2}, \ldots p_{qi}, \ldots p_{qt}\}$, if $p_{q1} = p_{q2}$ then the number of occurrences of p_{q1} is 2. This value is then inserted in the corresponding cell for the p_{q1}. Figure 4 shows an example of a query vector.

p_{q1}	p_{q3}	...	p_{qj}	...	p_{qt}
2	1	...	2	...	0

Fig. 4. Example of the query vector

Subsequently, each pair of the vector *vq* is searched in the index (matrix *Mec*) in order to obtain the number of times it is found in each *BPs* stored in the repository. This number is then multiplied by the corresponding value in the vector vq, and the resulting value is inserted in a new matrix named "query matrix" (*Mq*) where rows are the *BPs* of the repository and columns are the node pairs of the query vector vq (See Figure 5)

	p_{q1}	p_{q3}	...	p_{qj}	...	p_{qt}	**ec-sim**
BP_1	0	0	...	4	...	0	**4**
BP_2	6	0	...	0	...	0	**6**
...	...	...	...	...	...	...	...
BP_i	4	0	...	6	...	0	**10**
...	...	...	...	..	...	...	...
BP_n	2	5	...	0	...	0	**7**

Fig. 5. Example of the query matrix (Mq) plus the similarity for each BP

For example, the vector query of Figure 2 and the execution cases matrix of Figure 1, suppose that $p_{q1} = p1$, $p_{q3} = p_2$, $p_{qj} = p_j$ *and* $p_{qt} = p_k$. The pair p_{qj} with an occurrence of 2 in the vector query *vq* is found three times in the execution cases matrix, hence by multiplying those values we get 6; this value is inserted on the cell (*i; j*) of the query matrix *Mq*. Finally, in order to rank the BPs of the repository, the values of each row are added obtaining a value of execution cases similarity (*ec-sim*) for each *BPs*. Accordingly, the BPs are ranked from the greatest value to the lowest one. The complete resulting query matrix of the example is presented in Figure 5 where the resulting ranking is r = {BP_i(10), BP_n(7), BP_2(6), BP_1(4)…}.

3.3 A tool for implementing Trazas BP

The tool that implements TrazasBP was developed in Java and integrates a user centered interface. This tool incorporates some usability criteria defined by objective and subjective attributes. Among the objective attributes, the tool integrates: ease of learning and memorization, efficiency, effectiveness, operability and ease of understanding, equally. Additionally, subjective attributes (oriented to user satisfaction) supported in the tool are: accessibility, functionality, usefulness and credibility. The tool includes a simple interface with panels

ISSN 2395-8618

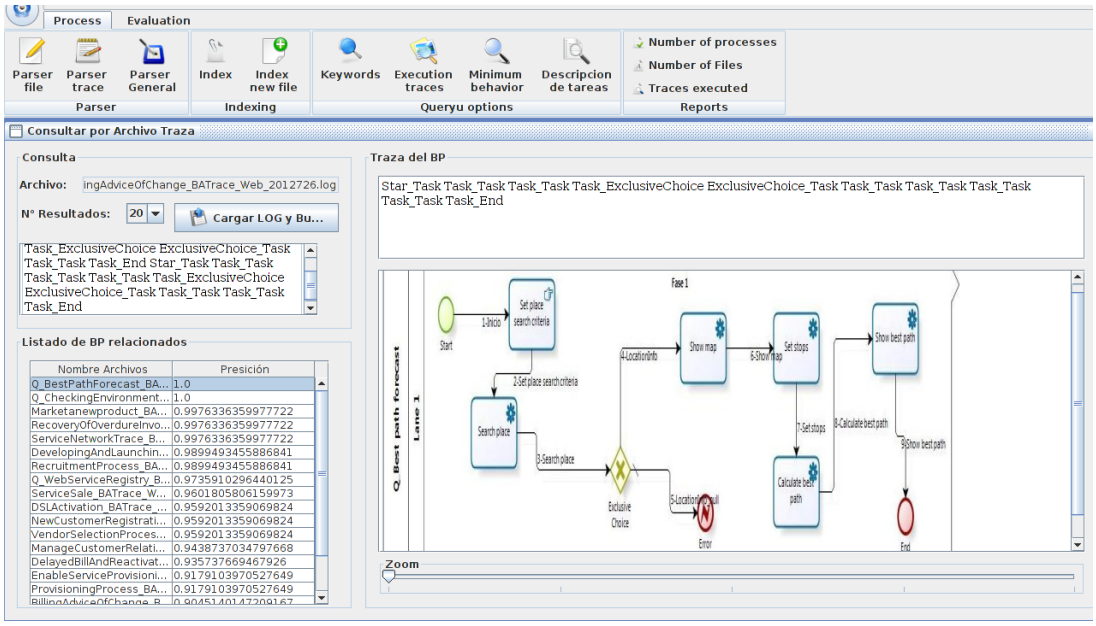


Fig. 6. User interfaces for performing queries

containing the functionalities of the model. Equally, the user may choose a BP model from the result list in order to visualize and thus check the validity of the query.

Figure 6 shows the results of one example query, in this figure results are displayed (red square) when the user performs the query. The results contain the more relevant BP models according to the similarity between the query and the BP in the repository.

IV. Evaluation and Results

Because For the experimental evaluation, TrazasBP was used for generating rankings of 20 BP according to the similarity with the Query BP. This procedure evaluates the relevance of the results retrieved in each search. The evaluation was performed using the measured widely used for evaluating information retrieval systems: Graded Precision, graded recall and F-measure.

Relevance evaluation of results in TrazasBP includes two phases: The first one evaluates relevance and quality of ranking, in order to find the best query option between: Execution case, Minimal behavior, and Log-file. The second phase compares results obtained using TrazasBP with the results of the manual evaluation performed on a closed test set, which was previously described in [21]. This closed test set was created collaborative by 59 experts. Moreover, the ranking generated by evaluators and the ranking automatically generated using the TrazasBP were compared using the measure A(Rq) presented in [22]. A(Rq) measure was used to determine the degree of coincidence of the position of each BP in each one of the rankings generated by each request.

Figure 7 shows results of the first phase, where for each querying option the Graded Precision (Gp), graded recall(Gr) and F-measure (Gf) are calculated. Graded precision reached values between 81% and 90% which means that the present approach is less likely to retrieve non-relevant BPs (i.e. false positives). Nevertheless, the lower values of recall from 19% to 28% demonstrate that the approach doesn't retrieve a high number of relevant BPs (i.e. false negatives). With regards to the F-measure, the approach obtained values from 30% to 42% for the different query options showing acceptable values of harmony between the precision and recall measures. Results of phase one show that query option based on Log-file achieved the best results, this is because each log file integrated many execution cases of the same BP, which extends the possibility for finding BP with similar execution cases in the repository. Consequently, the query option based on log files was selected for phase two.

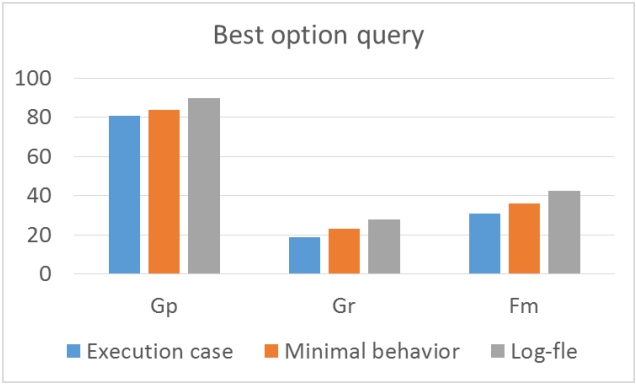


Fig. 7. Results of querying options comparison

ISSN 2395-8618

Results of phase two are shown in Figure 8. These results show that Trazas Bp achieved a 94% of Gp, therefore search results are precise and keep high similarity with the ranking generated by human evaluators. In other words, TrazasBP retrieves most of the BP that human evaluators considered as relevant for each query.

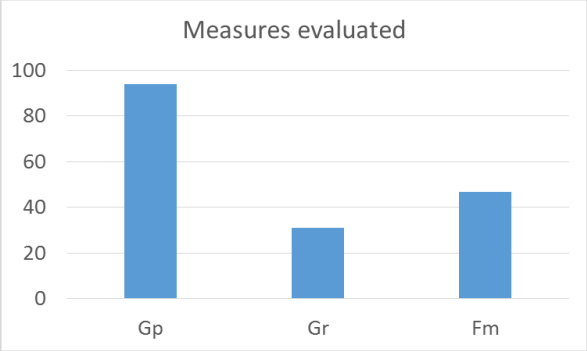


Fig. 8. Results of the phase 2

With regards to Graded precision, TrazasBP reached a value of 31%. This is due to TrazasBP generated rankings limited to 20 results; and it left aside other BPs relevant for the query. The results for graded F-measure evidenced harmony in the results of Gp and Gr. The average value of Gf is 47% which indicates that classifications generated by TrazasBP presented high similarity with the human generated ranking described in [21].

Figure 9 depicts the level of agreement A(Rq) between the ideal ranking generated by evaluators and the automatic classification generated using TrazasBP. Note that for each query the proposed approach generated classifications that match considerably with those generated by experts (ideal classifications). For example, in query 1 (Q1) the similarity of classification for the proposed method reached 85%. Finally, in the classification of global similarity (considering all the queries) TrazasBP reaches 81%. This result indicates an increase in quality of the generated ranking when Log files are used as query element. TrazasBP retrieves the most relevant list for each query and avoids retrieving no-relevant BP

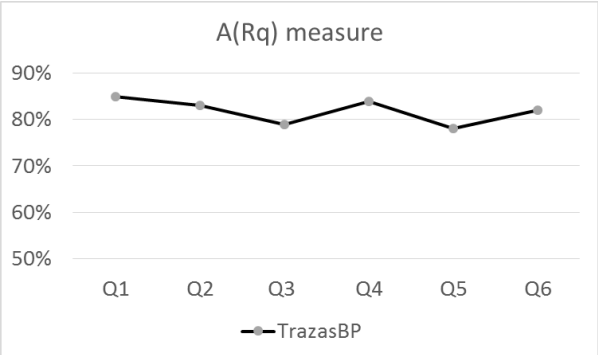


Fig. 9. Ranking concordance for each query (A(Rq))

V. Conclusion and Future Works

This paper presents TrazasBP, a framework for BP indexing and searching based on execution cases. TrazasBP indexes BPs based on execution cases retrieved from log files. Additionally, it considers not only textual information of BP elements but also causal dependence between BP elements. TrazasBP was evaluated in two phases: The first one evaluated relevance and quality of ranking using different querying options, and found as the best ranking option the one based on log-files. During the second phase, the present approach was compared with results obtained by human experts. Results obtained in this phase allow evidence that TrazasBP generates rankings of results with high similarity to the rankings generated by humans.

Experimental evaluation evidenced high values precision (90%) for different query options. Additionally, the F-measure reached values around 42% which is an acceptable value for the relation between precision and recall. Equally, When comparing TrazasBP with a closet test created by human experts, the Graded precision reached 94% which shows that the ranking generated with TrazasBP is highly similar to the ranking generated by human experts. Due to TrazasBP low computational cost, it may be effectively used as indexing mechanism and may precede other expensive algorithms during BPs similarity calculation since it reduces the search space. Additionally, TrazasBP approach can be extended by adding new query options.

Future work includes incorporating new search options: i) semantic options by adding domain ontologies that represent user queries. ii) multimodal options which consider structural, behavioral, and linguistic information in one search space. Equally, future work will include integration of clustering algorithms (like K-means, Clicks, Start, and C–means) to the model and compare the created groups with other results reported in the state of the art. On the other hand, it is planned to develop an automatic evaluation module that generates graphs and relevance measures. Finally, evaluation will be expanded by applying new measures for the BP search.

References

[1] H. A. Reijers, R. S. Mans, and R. a. van der Toorn, “Improved model management with aggregated business process models,” *Data Knowl. Eng.*, vol. 68, no. 2, pp. 221–243, Feb. 2009.

[2] F. M. Maggi, M. Dumas, and F. B. Kessler, “Predictive Monitoring of Business Processes,” In Proc. *26th International Conference, CAiSE 2014*, Thessaloniki, Greece, pp.457-472.

[3] F. Rahimi, C. Møller, and L. Hvam, “Business process management and IT management: The missing integration,” *Int. J. Inf. Manage.*, vol. 36, no. 1, pp. 142–154, Feb. 2016.

[4] I. Khodyrev and S. Popova, “Discrete Modeling and Simulation of Business Processes Using Event Logs,” *Procedia Comput. Sci.*, vol. 29, pp. 322–331, 2014.

[5] W. M. P. van der Aalst, H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. Alves de Medeiros, M. Song, and H. M. W. Verbeek,

ISSN 2395-8618

"Business process mining: An industrial application," *Inf. Syst.*, vol. 32, no. 5, pp. 713–732, 2007.

[6] J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens, "A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs," *Inf. Syst.*, vol. 37, no. 7, pp. 654–676, Nov. 2012.

[7] I. Bider, K. Gaaloul, J. Krogstie, S. Nurcan, H. A. Proper, R. Schmidt, and P. Soffer, "Enterprise, business-process and information systems modeling," in *Lecture Notes in Business Information Processing*, 2014, vol. 175.

[8] J. Y. In P. Maglio, M. Weske and and M. Fantinato, "Discovering business process similarities: An empirical study with sap best practice business processes," in Proc *International Conference on Service-Oriented Computing*, pp. 515-526, 2010.

[9] A. K. Alves de Medeiros, W. M. P. Van der Aalst, and A. J. M. M. Weijters, "Quantifying process equivalence based on observed behavior," *Data Knowl. Eng.*, vol. 64, no. 1, pp. 55–74, 2008.

[10] R. Dijkman, M. Dumas, and L. García-Bañuelos, "Graph matching algorithms for business process model similarity search," *Bus. Process Manag.*, vol. Business P, pp. 48–63, 2009.

[11] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search with feature-based similarity estimation," in Proc *OTM Confederated International Conferences*, 2010, pp. 60–77.

[12] S. Goedertier, D. Martens, J. Vanthienen, and B. Baesens, "Robust Process Discovery with Artificial Negative Events," *J. Mach. Learn. Res.*, vol. 10, pp. 1305–1340, 2009.

[13] M. Segatto, S. I. D. De Pádua, and D. P. Martinelli, "Business process management: a systemic approach?," *Bus. Process Manag. J.*, vol. 19, no. 4, pp. 698–714, 2013.

[14] J. Bae, L. Liu, J. Caverlee, L.-J. Zhang, and H. Bae, "Development of Distance Measures for Process Mining, Discovery and Integration," *Int. J. Web Serv. Res.*, vol. 4, no. 4, pp. 1–17, 2007.

[15] M. Weidlich, A. Polyvyanyy, J. Mendling, and M. Weske, "Causal behavioural profiles - Efficient computation, applications, and evaluation," in *Fundamenta Informaticae*, 2011, vol. 113, no. 3–4, pp. 399–435.

[16] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Inf. Syst.*, vol. 36, no. 2, pp. 498–516, Apr. 2011.

[17] H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun, "A workflow net similarity measure based on transition adjacency relations," *Comput. Ind.*, vol. 61, no. 5, pp. 463–471, 2010.

[18] A. Wombacher and M. Rozie, "Evaluation of workflow similarity measures in service discovery," *Serv. Oriented Electron. Commer.*, vol. 7, no. 26, pp. 51–71, 2006.

[19] K. Gerke, J. Cardoso, and A. Claus, "Measuring the compliance of processes with reference models," in *Proc OTM Confederated International Conferences*, 2009, pp. 76–93.

[20] J. Wang, T. He, L. Wen, N. Wu, A. H. M. Ter Hofstede, and J. Su, "A behavioral similarity measure between labeled Petri nets based on principal transition sequences (short paper)," in *Proc OTM Confederated International Conferences*, 2010, pp. 394–401.

[21] J. C. Corrales, C. Cobos, L. K. Wives, and L. Thom, "Collaborative Evaluation to Build Closed Repositories on Business Process Models," in Proc *ICEIS*, 2014, pp. 311–318.

[22] M. Guentert, M. Kunze, and M. Weske, "Evaluation Measures for Similarity Search Results in Process Model Repositories," pp. 214–227, 2012.

ISSN 2395-8618

Unsupervised Domain Ontology Learning from Text

V. Sree Harissh, M. Vignesh, U. Kodaikkaavirinaadan, and T. V. Geetha

***Abstract*—Construction of Ontology is indispensable with rapid increase in textual information. Much research in learning Ontology are supervised and require manually annotated resources. Also, quality of Ontology is dependent on quality of corpus which may not be readily available. To tackle these problems, we present an iterative focused web crawler for building corpus and an unsupervised framework for construction of Domain Ontology. The proposed framework consists of five phases, Corpus Collection using Iterative Focused crawling with novel weighting measure, Term Extraction using HITS algorithm, Taxonomic Relation Extraction using Hearst and Morpho-Syntactic Patterns, Non Taxonomic relation extraction using association rule mining and Domain Ontology Building. Evaluation results show that proposed crawler outweighs traditional crawling techniques, domain terms showed higher precision when compared to statistical techniques and learnt ontology has rich knowledge representation.**

***Index Terms*—Iterative focused crawling, domain ontology, domain terms extraction, taxonomy, non taxonomy.**

I. Introduction

ONTOLOGY in computer science can be viewed as formal representation of knowledge pertaining to particular domain [1]. In simpler terms ontology provides concepts and relationship among concepts in a domain. Machines perceive contents of documents(blogs, articles, web pages, forums, scientific research papers, e-books, etc.) as sequence of character. Much of the semantic information are already encoded in some form or other in these documents. There is an increasing demand to convert these unstructured information into structured information. Ontology plays a key role in representing the knowledge hidden in these texts and make it human and computer understandable.

Construction of Domain Ontology provides various semantic solutions including: (1) Knowledge Management, (2) Knowledge Sharing, (3) Knowledge Organization, and (4) Knowledge Enrichment.

It can be effectively used in semantic computing applications ranging from Expert Systems [2], Search Engines [3], Question and Answering System [4], etc. to solve day to day problems. For example, if the search engine is aware that "prokaryote" is a type of organism, better search results can be obtained and recall of the system will be improved subsequently.

Ontology is generally built under the supervision of domain experts and are time intensive process. Corpus required for building Ontology are not always readily available. Therefore, it is important to build corpus from web through crawling. Very few work is available that have incorporated crawling as a phase for collecting corpus in building Ontology. Since general crawling does not always provide domain related pages, lot of irrelevant pages are downloaded and filtering is required. Terms extracted using statistical measure or linguistic patterns are prone to noise and require additional level of filtering using machine learning techniques. Also, most systems rely on manually annotated resources for obtaining terms and also for relation discovery. These resources however mostly contain domain generic concepts and lack domain specific concepts and relations [1]. Ontology extracted using lexico-syntactic patterns are limited to certain patterns and require enrichment.

In this work we propose a framework for crawling websites relevant to the domain of interest and also build Domain Ontology without use of any annotated resource in an unsupervised manner. The crawling framework uses a novel weighting measure to rank the domain terms. The proposed framework consists of five phases Corpus Collection, Term Extraction, Taxonomic Relation Extraction, Non-taxonomic relation extraction and Domain Ontology building. Corpus is crawled using iterative focused web crawler which downloads the content which are pertinent to the domain by selectively rejecting URL's based on link, anchor text and link context. Terms are extracted by feeding graph based algorithm HITS with Shallow Semantic Relations and proposed use of adjective modifiers to obtain fine grained domain terms. Hearst pattern and Morpho-Syntactic patterns are extracted to build taxonomies. Non-taxonomic relation extraction is obtained through Association Rule Mining on Triples.

The organization of the paper is as follows: section two describes Related Work, section three describes the System Design, section four describes the Results and Evaluation, section five describes Conclusion and section six describes Future Work.

II. Related Work

In this section, we discuss the literature survey in Corpus Collection, Term Extraction, Taxonomic Relation Extraction and Non Taxonomic Relation Extraction.

Manuscript received on December 21, 2016, accepted for publication on June 18, 2017, published on June 30, 2018.

The authors are with Department of Computer Science and Engineering, College of Engineering, Guindy, India (e-mail: {vharissh14,vigneshmohanceg,naadan.uk}@gmail.com, tv_g@hotmail.com).

ISSN 2395-8618

A. Domain Corpus Collection

Domain Corpus is a coherent collection of domain text. It requires the usage of iterative focused or topical web crawler to fetch the pages that are pertinent to the domain of interest. In the work proposed by [5], a heuristic based approach is used to locate anchor text by using DOM tree instead of using the entire HTML Page. A statistical based term weighing measure based on TF-IDF called TFIPNDF (Term Frequency Inverse Positive Negative Document Frequency) was proposed for weighing anchor text and link context. The pages are classified as relevant or not relevant on the basis of trained classifier and is entirely supervised. The work however lacks iterative learning of terms to classify pages [6].

B. Domain Term Extraction

Domain Terms are the elementary components used to represent concepts of a domain. Example of domain terms pertaining to agricultural domain are "farming", "crops", "plants", "fertilizers", etc. Term Extraction is generally performed from collection of domain documents using any of the following methods: Statistical Measure, Linguistic Measure, Machine Learning and Graph-based Measure.

1) Statistical Measure: Most common Statistical Measure make use of TF (Term Frequency) and IDF (Inverse Document Frequency). Meijer et al. [7], proposed four measures namely Domain Pertinence, Lexical Cohesion, Domain Consensus and Structural Relevance to compute the importance of terms in a domain. Drymonas et al. [8], used C/NC values to calculate the relevance of multiword terms in corpus. These measures however fail to consider the context of terms and fails to capture the importance of infrequent domain terms.

2) Linguistic Measure: Linguistic Measures traditionally acquire terms by using syntactic patterns such as Noun-Noun, Adjective-Noun, etc. For example, the POS tagging of the sentence "Western Rajasthan and northern Gujarat are included in this region" tags "Western" as an adjective and "Rajasthan" as Noun. Lexico-Syntactic patterns makes use of predefined patterns such as "including", "like", "such as", etc., to extract terms. It is however tedious and time consuming to pre-define patterns.

3) Machine Learning: Machine Learning is either supervised or unsupervised. Supervised learning require the algorithm to be trained before usage and target variable is known. Some famous and commonly used supervised algorithms include Naive Bayes, Support Vector Machines and Decision Tree. In unsupervised learning training is not required and hidden patterns are found using unlabeled data. Uzun [9] work considers training features are independent and therefore used TF-IDF, distance of the word to the beginning of the paragraph, word position with respect to whole text and sentence and probability features from Naive Bayes Classifier to classify whether a term is relevant. The drawback of using machine learning is that training incurs overhead and data may not be available in abundance for training.

4) Graph Based Measure: Graph Based Measure is used to model the importance of a term and the relationship between the terms in an effective way. Survey on Graph Methods by Beliga et al. [10], suggest that graphs can be used to represent co-occurrence relations, semantic relations, syntactic relations and other relations (intersecting words from sentence, paragraph, etc.). Work by Ventura et al. [11] used novel graph based ranking method called "Terminology Ranking Based on Graph Information" to rank the terms and dice coefficient was used to measure the co-occurrence between two terms. Mukherjee et al. [12] used HITS index with hubs as Shallow Semantic Relations and authorities as nouns. Terms are filtered based on hubs and authority scores.

C. Taxonomic Relation Extraction

Taxonomy construction involves building a concept hierarchy in which broader-narrower relations are stored and can be visualized as a hierarchy of concepts. For example "rice", "wheat", "maize" come under "crop". They are commonly built using predefined patterns such as the work by Hearst [13] and Ochoa et al. [14]. Meijer et al. [7] proposed construction of taxonomy using subsumption method. This method calculates co-occurrence relations between different concepts. Knijff et al. [15], compared two methods subsumption method and hierarchical agglomerative clustering to construct taxonomy. They concluded that subsumption method is suitable for shallow taxonomies and hierarchical agglomerative clustering is suitable for building deep taxonomies.

D. Non Taxonomic Relation Extraction

Non Taxonomic Relations best describe the non-hierarchical attributes of concept. For example, in the non taxonomic relation "predators eat plants", eat is a feature of predator. Nabila et al. [16] proposed an automatic construction of non-taxonomic relation extraction by finding the non-taxonomic relations between the concepts in the same sentence and non-taxonomic relations between concepts in different sentences. Serra and Girardri [17] proposed a semi-automatic construction of non-taxonomic relations from text corpus. Association between two concepts are found by calculating the support and the confidence scores between the two concepts.

a) : To build a Domain Ontology from Text, the existing methods for Domain Term Extraction deprive from identification of low frequent terms, identification of all syntactic-patterns and require annotated re-sources for machine learning approaches. Graph based methods for identification can be used to solve the above problems as they can represent the meaning as well as composition of text. They also do not require manually annotated data unlike machine learning approaches. General Non-Taxonomic Relation Extraction methods are based on extraction of predicates between two concepts and as all predicates are not domain specific the use of Data Mining Techniques can be helpful in identifying the Domain Relations effectively.

ISSN 2395-8618

III. System Design

In this section we discuss the design of our system. Figure 1 shows the overall architecture diagram of the proposed framework. The system consists of five major phases: (1) Domain Corpus Collection, (2) Domain Term Extraction 3) Taxonomic Relation Extraction, (4) Non Taxonomic Relation Extraction, and (5) Domain Ontology Building.

A. Domain Corpus Collection

Corpus required for construction of Ontology may not be readily available for every domain. Since the quality of the corpus plays a vital role in deciding the quality of Ontology, Iterative Focused Crawling is performed to download web pages relevant to the domain. List of Seed URLs are given as input to the Iterative Focused Crawler. The web pages whose URL, anchor text or link context satisfy the relevance score are added to the URL queue. The depth of the pages to be crawled is specified. The output of the focused crawler is used as corpus for construction of Ontology. Crawling is terminated when the relevance of URL to the context vector decreases drastically. The architecture of crawler is depicted in Figure 2.

Nouns are considered as candidate terms for finding keywords in the domain. Therefore, the nouns are extracted from the corpus using the Stanford parts-of-speech tagger. The context vector of a noun is computed by using proposed weighted co-occurrence score. Weighted co-occurrence ($WCO(w_i, w_j)$) of two words w_i and w_j is given by :

$$WCO(w_i, w_j) = CO(w_i, w_j) X idf(w_i) X idf(w_j) \quad (1)$$

In Equation 1, $idf(w_i)$ and $idf(w_j)$ are the inverse document frequency of words w_i and w_j. $CO(w_i, w_j)$ is the co-occurrence frequency of the two words w_i and w_j. The proposed equation considers the inverse document frequencies of the terms in order to consider the importance of terms which occur rarely and may of importance to the domain. Unit Normalization of the context vector is performed to have a specific range of score between 0 and 1. The normalized context vector of each term is summed along the column and sorted in descending order. The top ranked terms are extracted as concepts based on percentage.

Relevance of the web pages are calculated by computing the average of the Cosine Similarity Score of the test domain vectors and each of the domain vectors of the training document. The relevance of the URL is checked without scanning the pages. It is done by computing relevance of HREF, Anchor Text and/or Link Context. Appropriate threshold are set for HREF, Anchor Text and Link Context. If HREF is not relevant (i.e Relevance Score), Anchor Text will be checked for relevance. If Anchor Text is not relevant, finally, Link Context will be checked.

B. Domain Term Extraction

Domain corpus, which contains a rich collection of text documents is pre-processed to identify the domain terms. Numbers, special characters, etc. which do not play a significant role in ontology construction are removed.

1) Shallow Semantic Relation Extraction: Domain text documents are tokenized into sentences. These sentences are parsed using Stanford Dependency Parser to identify the Shallow Semantic Relations between the words. Shallow Semantic Relations represent the syntactic contextual relations within the sentences. In addition to the Shallow Semantic Relations extracted in [12] we have also extracted and used adjective modifiers obtained through Dependency Parsing. Since, significant amount of domain terms are composed as adjective modifier, it is important to consider these dependencies. For example, in the sentence "Biological research into soil and soil organisms has proven beneficial to organic farming.", "organic farming" and "biological research" are tagged as adjective modifiers.

2) Domain Term Induction Using HITS: HITS algorithm [12], [18] is applied to identify the most important domain terms. It is composed of two major components—Hubs and Authorities. Hubs are represented by Shallow Semantic Relations and authorities are represented by nouns. Hub score is calculated as the sum of authority scores and authority score is calculated as the sum of hub scores. Hub and Authority score are calculated recursively until hub and authority score converges. The Shallow Semantic Relation which has high hub score are selected as multi-grams and nouns which has high authority score are selected as unigrams. These unigrams and multi-grams constitute the domain terms.

C. Taxonomic Relation Extraction

Taxonomic Relations represent hypernym-hyponym relation. A hypernym represents the specific semantic field of a hyponym and a hyponym represents the generic semantic field of the hyponym. The three steps involved in building a taxonomy involves (i) Hearst Pattern Extraction and (ii) Morpho-syntactic Pattern Extraction

1) Hearst Pattern Extraction: Hearst Patterns [13] are commonly used to extract taxonomic relations from text. Sentences containing the domain terms are selected for identification of Hearst Patterns. Sentences are tagged using parts-of-speech tagger to find six types of hearst patterns. Six types of hearst patterns are as listed below: NP_i is considered as a hypernym and NP_j is considered as a hyponym.

1) NP_i such as NP_j

Example : agrochemicals such as pesticides and fertilizers where agrochemicals is a hypernym and fertilizers and pesticides is a hyponym.

2) NP_j or other NP_i

Example : iron, magnesium, zinc, or other nutrients where nutrients is a hypernym and iron, magnesium and zinc are hyponyms.

ISSN 2395-8618

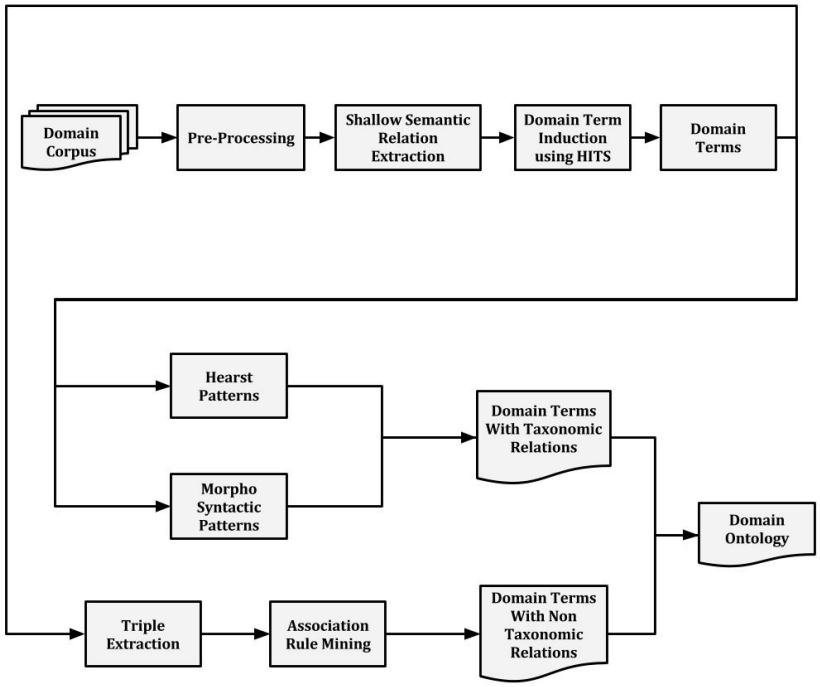


Fig. 1. Architecture of Proposed Framework: Unsupervised Domain Ontology Construction From Text

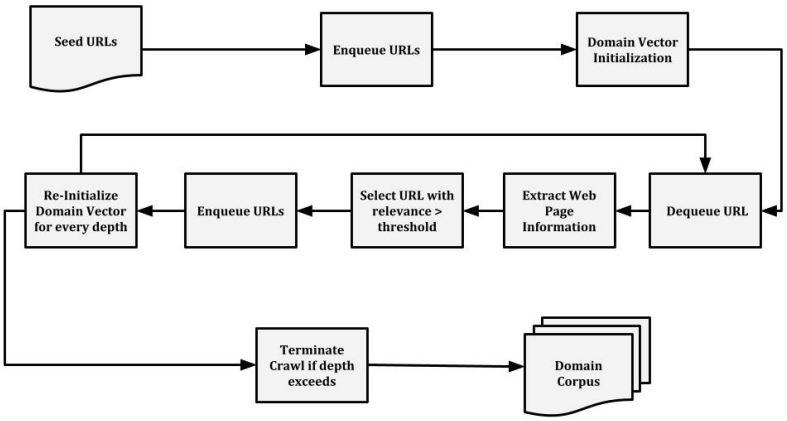


Fig. 2. Flow Diagram of Iterative Focused Crawler

3) NP_j and other NP_i
Example : barley, wheat, and other cereals where cereals is a hypernym and barley and wheat is a hyponym.
4) such NP_j as NP_j
Example : such foods as bread, porridge, crackers, biscuits where foods is a hypernym and bread, porridge, crackers and biscuits is a hyponym.
5) NP_i including NP_j
Example : political issues including water pollution where political issues is a hypernym and water pollution is a hypernym.
6) NP_i especially NP_j
Example : Tropical fruits especially bananas grows in South India where Tropical fruits is a hypernym and bananas is a hyponym.

2) Morpho Syntactic Pattern Extraction: In our work we have also extracted Morpho Syntactic Patterns [14] to extract additional Hypernym-Hyponym relations. There are two rules followed to extract morpho-syntactic patterns.

Rule 1 : If the term t_1 contains a suffix string t_0, then the term t_0 is the hypernym of the term t_1, provided the term t_0 or t_1 is a domain term. For example, “polysaccharide” is considered as the hypernym of the term “homopolysaccharide”.

Rule 2 : If the term t_0 is the head term of the term t_1 , then t_0 is considered as the hypernym of the term t_1, provided term t_0 or t_1 is the domain term. Example: “sweet corn” is the

ISSN 2395-8618

hyponym of the word “corn”.

D. Non-Taxonomic Relation Extraction

Non-Taxonomic Relations represent the properties of the object. It has no class-subclass relationship.

1) Triplet Extraction: A sentence is composed of three components - subject, predicate and object. A triplet in a sentence is defined as the relation between the subject and the object, with the relation being the predicate. Parsed documents using Stanford Parser are input to the triplet extraction process. Subject, predicate and object from the sentences is extracted using Russu’s Triple Algorithm [19].

2) Association Rule Mining: Association Rule Mining [20] is performed to find the non-taxonomic relations between the domain terms. Apriori Algorithm is used for frequent itemset generation and association rule mining. Frequent itemset whose support crosses a suitable threshold are selected for mining association rules. Association rules are filtered from frequent itemsets and association rules which satisfy a suitable confidence score are selected.

E. Domain Ontology Building

The concepts with the taxonomic and non-taxonomic relations are represented in a Resource Description Framework format. The concepts consists of a concept id, a broader relation, a narrower relation and a non-taxonomic relation associated with it. The broader/narrower relation are represented by class/subclass relations. Non-taxonomic relations consists of a property, domain and range. The domain of a property represent the subject whose predicate is that property. The range of a property represent the object whose predicate is that property. Example : “rice” is a concept with concept id “12143”, narrower relations “long-grain rice”,broader relation “crops”, “medium-grain rice”, “short-grain rice”, property “grows in”, domain “rice”, range “South India”.

IV. RESULTS AND EVALUATION

A. Domain Corpus Collection

Domain Corpus Collection consists of implementing an iterative focused web crawler that crawls pages relevant to the domain. 22 seed URLS pertaining to agriculture domain were given as input to the focused crawler. 20,632 documents were obtained at the end of crawling a depth of 3. Table I shows the number of relevant links crawled by the crawler.

TABLE I
NUMBER OF LINKS CRAWLED AT DIFFERENT DEPTHS

Depth	Number of Links Crawled
0	22
1	134
2	816
3	19732
Total	20632

Table II shows the Number of Links crawled through HREF, Anchor Text and Link Context. It is observed that most of the links were found to be relevant through HREF and Link Context. HREF usually contain the text present in the Anchor Text. So, if the relevance fails through HREF there is a high probability of checking the Link Context.

TABLE II
NUMBER OF LINKS CRAWLED THROUGH HREF, ANCHOR TEXT AND LINK CONTEXT

Mode	Count
HREF	606
Anchor Text	2256
Link Context	17842
Total	20632

Table III shows the Number of documents in different similarity range compared to SeedURL pages. It can be seen that most of the pages similarity were in the range of 0.5 to 0.6.

TABLE III
NUMBER OF LINKS CRAWLED THROUGH HREF, ANCHOR TEXT AND LINK CONTEXT

Similarity	Count
0.6 - 0.7	787
0.5 - 0.6	11624
0.4 - 0.5	5782
0.3 - 0.4	2156
0.2 - 0.3	251
0.1 - 0.2	60
0.0 - 0.1	22
Total	20632

Figure 3 shows Histogram analysis of document count to similarity of documents at various depths and Median of similarity score for a particular depth w.r.t seed documents. Histogram analysis strongly suggest that most of the documents crawled belongs to the similarity range of 0.5 to 0.6. It was also observed that the median of relevance score follows a decreasing trend and the number of irrelevant links crawled increased after a depth of 3.

In our work, Convergence Score [21] was used to evaluate the Iterative Focused Crawler. It is defined as the number of concepts present in the final crawl to the number of concepts present in initial seed page set. From Figure 4, we infer that the convergence of Focused Crawler is better than Base Line Crawler since the former crawls the page that are semantically relevant.

B. Domain Term Extraction

In our work, HITS algorithm was used to extract the top quality domain terms. The algorithm took nearly 3000 iterations to rank top quality domain terms.

The precision scores of Graph Based Domain Term Extraction using HITS algorithm used in our work is evaluated against statistical measures such as Linguistic Patterns, Inverse Document Frequency , C-value(LIDF score) and Graph Based

ISSN 2395-8618

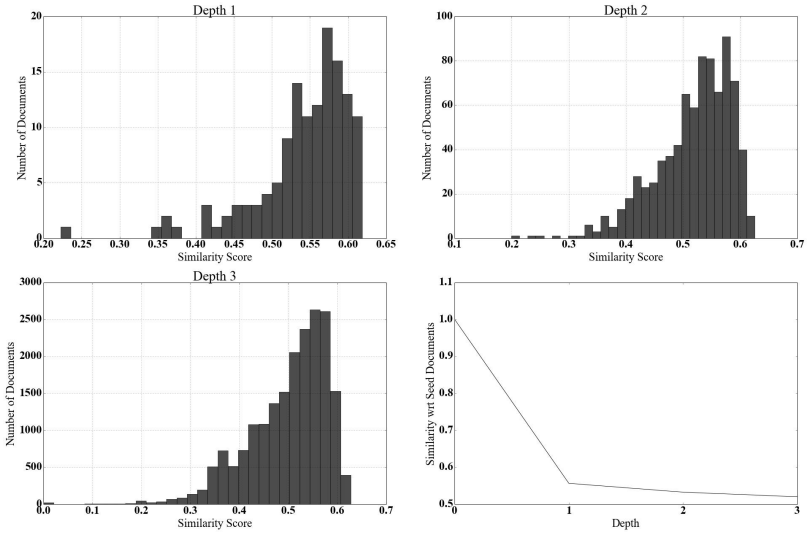


Fig. 3. Histograms analysis of Similarity Scores w.r.t Depth and Median of Similarity Score for a particular Depth w.r.t seed documents

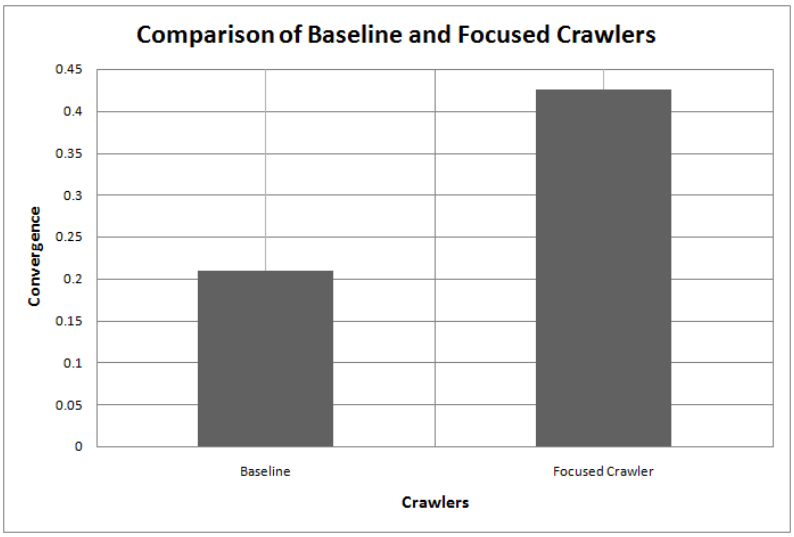


Fig. 4. Comparison of Baseline crawler to Focused Crawler using Convergence Score

TABLE IV
PRECISION SCORES OF TERM EXTRACTION USING HITS, LIDF, TERGRAPH AND DP+DC+LC+SR

Total Terms	Term Extraction using HITS	LIDF	TeRGraph	DP+DC+LC+SR
1000	0.772	0.697	0.769	0.751
2000	0.749	0.662	0.694	0.687
3000	0.733	0.627	0.644	0.657
4000	0.703	0.608	0.593	0.612
5000	0.676	0.575	0.562	0.583
6000	0.662	0.550	0.561	0.561
7000	0.651	0.547	0.552	0.550
8000	0.633	0.546	0.546	0.538

Algorithm Terminology Ranking Based on Graph Information - TeRGraph proposed by [11] and sum of statistical scores obtained from Domain Pertinence (DP), Domain Consensus (DC), Lexical Cohesion (LC) and Structural Relevance (SR) proposed in [7] is shown in Table IV. GENIA corpus used in [11] was used for evaluation purpose. The measures shows that graph based HITS algorithm shows better precision compared to statistical measures and Graph Based algorithm TeRGraph.

ISSN 2395-8618

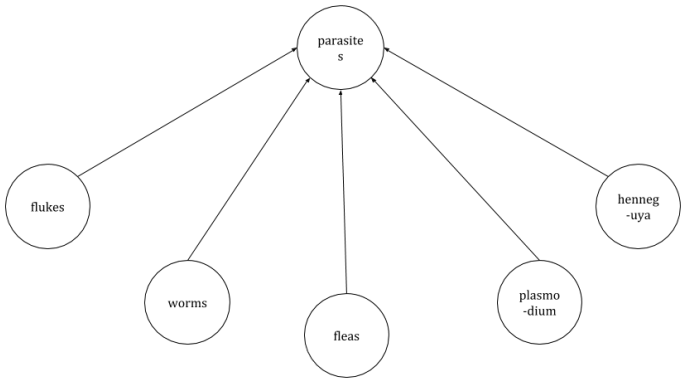


Fig. 5. Taxonomy of Parasites

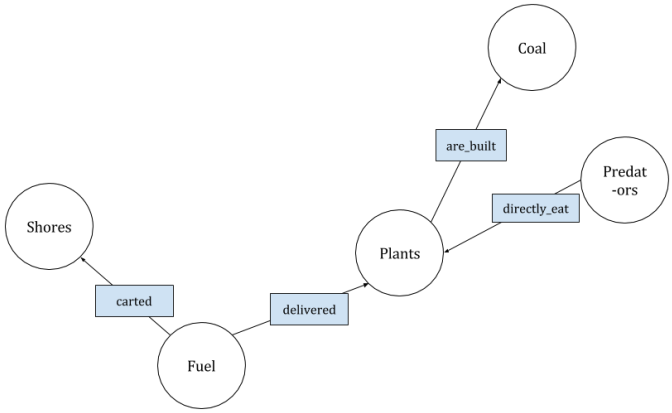


Fig. 6. Non Taxonomy of Plants

C. Domain Ontology

Hearst Patterns and Morpho-Syntactic patterns were used to induce Taxonomy. Total of 6539 Hearst Patterns and 2149 Morpho-Syntactic patterns were extracted to construct the Taxonomy. 5216 triples were extracted and 357 Non Taxonomic Relations were identified using Association Rule Mining. Figure 5 shows the snippet of Parasite Taxonomy and Figure 6 shows the snippet of Non Taxonomic Relations associated with Plants.

In our work, Domain Ontology was evaluated using Metic Based Evaluation techiques Inheritance Richness and Class Richness [22].

1) Class Richness: This metric is related to how instances are distributed across classes. The number of classes that have instances in the KB is compared with the total number of classes, giving a general idea of how well the KB utilizes the knowledge modeled by the schema classes. Thus, if the KB has a very low Class Richness, then the KB does not have data that exemplifies all the class knowledge that exists in the schema. On the other hand, a KB that has a very high CR would indicate that the data in the KB represents most of the knowledge in the schema. Table V shows the Class Richness score for Taxonomy and Non Taxonomy learning methods.

2) Inheritance Richness: Inheritance Richness measure describes the distribution of information across different levels of the ontology's inheritance tree or the fan-out of parent classes. This is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology. This measure can distinguish a horizontal ontology (where classes have a large number of direct subclasses) from a vertical ontology (where classes have a small number of direct subclasses). An ontology with a low inheritance richness would be of a deep (or vertical) ontology, which indicates that the ontology covers a specific domain in a detailed manner, while an ontology with a high IR would be a shallow (or horizontal) ontology, which indicates that the ontology represents a wide range of general knowledge with a low level of detail. Table V shows the Class Richness score for Taxonomy and Non Taxonomy learning methods.

From the results of the evaluation metrics(class richness and inheritance richness), it is evident that the constructed ontology has a good density depicting that the concepts extracted represents a wider knowledge in the domain.

ISSN 2395-8618

TABLE V
INHERITANCE AND CLASS RICHNESS SCORES

Method	Inheritance Richness	Class Richness
Hearst	4.004	0.367
Morpho-Syntactic	2.671	0.068
Hearst + Morpho-Syntactic	3.967	0.41
Non-Taxonomic Relation	1.81	0.21

V. CONCLUSION AND FUTURE WORK

In our work, we have developed an iterative focused crawler for collection of domain corpora, with each element in the co-occurrence matrix weighted as product of co-occurrence frequency and IDF of row and column. The generic terms extracted as concepts are removed using statistical measure. The relevance of the page is checked in the following levels: URL, Anchor Text and Link Context. Domain terms were extracted without any manual annotated resource unsupervised using HITS algorithm with Hubs as Shallow Semantic Relation and Authority as Nouns. The ranked terms were removed of noise using Domain Pertinence. In this work, taxonomy was induced using Hearst Patterns and Morpho-Syntactic Patterns. The Ontology was built automatically without supervision from scratch. In the future, we intend to exploit deep learning methods for building Domain Ontology to make it meaningful and useful.

REFERENCES

[1] Y. Sure, S. Staab, and R. Studer, "Ontology engineering methodology," in *Handbook on ontologies*. Springer, 2009, pp. 135–152.

[2] L.-Y. Shue, C.-W. Chen, and W. Shiue, "The development of an ontology-based expert system for corporate financial rating," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2130–2142, 2009.

[3] Y. Zhang, W. Vasconcelos, and D. Sleeman, "Ontosearch: An ontology search engine," in *Research and Development in Intelligent Systems XXI*. Springer, 2005, pp. 58–69.

[4] V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," in *European Semantic Web Conference*. Springer, 2005, pp. 546–562.

[12] S. Mukherjee, J. Ajmera, and S. Joshi, "Domain cartridge: Unsupervised framework for shallow domain ontology construction from corpus," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 929–938.

[5] L. Liu, T. Peng, and W. Zuo, "Topical web crawling for domain-specific resource discovery enhanced by selectively using link-context," *Proc. The International Arab Journal of Information Technology*, vol. 12, no. 2, 2015.

[6] R. Sheikh, "A review of focused web crawling strategies."

[7] K. Meijer, F. Frasincar, and F. Hogenboom, "A semantic approach for extracting domain taxonomies from text," *Decision Support Systems*, vol. 62, pp. 78–93, 2014.

[8] E. Drymonas, K. Zervanou, and E. G. Petrakis, "Unsupervised ontology acquisition from plain texts: The OntoGain system," in *International Conference on Application of Natural Language to Information Systems*. Springer, 2010, pp. 277–287.

[9] Y. Uzun, "Keyword extraction using naïve bayes," in *Bilkent University, Department of Computer Science, Turkey www. cs. bilkent. edu. tr/˜ guvenir/courses/CS550/Workshop/Yasin_Uzun. pdf*, 2005.

[10] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, "An overview of graph-based keyword extraction methods and approaches," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 1–20, 2015.

[11] J. A. Lossio-Ventura, C. Jonquet, M. Roche, and M. Teisseire, "Yet another ranking function for automatic multiword term extraction," in *International Conference on Natural Language Processing*. Springer, 2014, pp. 52–64.

[13] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1992, pp. 539–545.

[14] J. L. Ochoa, Á. Almela, M. L. Hernández-Alcaraz, and R. Valencia-García, "Learning morphosyntactic patterns for multiword term extraction," *Scientific Research and Essays*, vol. 6, no. 26, pp. 5563–5578, 2011.

[15] J. De Knijff, F. Frasincar, and F. Hogenboom, "Domain taxonomy learning from text: The subsumption method versus hierarchical clustering," *Data & Knowledge Engineering*, vol. 83, pp. 54–69, 2013.

[16] N. Nabila, A. Mamat, M. Azmi-Murad, and N. Mustapha, "Enriching non-taxonomic relations extracted from domain texts," in *2011 International Conference on Semantic Technology and Information Retrieval*. IEEE, 2011, pp. 99–105.

[17] I. Serra and R. Girardi, "A process for extracting non-taxonomic relationships of ontologies from text," 2011.

[18] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[19] D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proceedings of the 10th International Multiconference" Information Society-IS*, 2007, pp. 8–12.

[20] R. Srikant and R. Agrawal, *Mining generalized association rules*. IBM Research Division, 1995.

[21] S. Thenmalar and T. Geetha, "The modified concept based focused crawling using ontology," *Journal of Web Engineering*, vol. 13, no. 5-6, pp. 525–538, 2014.

[22] S. Tartir, I. B. Arpinar, M. Moore, A. P. Sheth, and B. Aleman-Meza, "OntoQA: Metric-based ontology quality analysis," 2005.

ISSN 2395-8618

GeCaP: Generador de casos de pruebas unitarias a partir del código fuente en lenguaje Java

D. Larrosa, P. Fernandez, and M. Delgado

***Resumen*—Las pruebas de software, no obstante que son costosas, aumentan considerablemente la confiabilidad y calidad de los sistemas, contribuyendo así a su posicionamiento en el mercado. Específicamente, las pruebas unitarias se encargan de probar que las unidades individuales del diseño de software, componente o módulo de software, funcionan correctamente. Aunque existen herramientas que se encargan de ejecutar pruebas unitarias de manera automática, éstas carecen de funcionalidades que proporcionen apoyo y asistencia al desarrollador en el diseño de los casos de prueba; además, las propuestas existentes para el diseño de los casos de pruebas unitarias, no se han insertado al entorno productivo y no permiten generar código de pruebas. En el presente trabajo se propone una herramienta que permite la generación automática de casos de pruebas unitarias a partir del código fuente en lenguaje Java. En la nueva propuesta se utiliza la técnica del camino básico para el diseño de los casos de prueba. De forma automática se genera el grafo de control de flujo del código fuente a probar, para luego generar los caminos independientes; por último, se generan las combinaciones de valores de prueba que satisfagan todos y cada uno de los caminos independientes. En el proceso de implementación de la nueva herramienta, se diseñó un caso de estudio para efectos de validación; se aplicaron algoritmos metaheurísticos para la generación de valores de prueba y para la generación de combinaciones de valores para cada camino, y se compararon estas combinaciones de valores con las obtenidas por otros algoritmos del estado del arte. Dado que en el caso de estudio se alcanza un 100% de cobertura de los caminos independientes la nueva herramienta exhibe resultados competitivos respecto de los resultados obtenidos por herramientas propuestas por otros autores.**

***Palabras clave*—pruebas unitarias, técnica del camino básico, generación automática de casos de prueba, algoritmos metaheurísticos.**

Manuscrito recibido el 10 de octubre de 2017, aceptado para la publicación el 17 de febrero de 2018, publicado el 30 de junio de 2018.

Danay Larrosa está con la Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), La Habana, Cuba (correo: dlarrosau@ceis.cujae.edu.cu).

Perla Beatriz Fernández Oliva está con la Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), La Habana, Cuba (correo: perla@ceis.cujae.edu.cu).

Martha Dunia Delgado Dapena está con la Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), La Habana, Cuba (correo: marta@ceis.cujae.edu.cu).

GeCaP: Unit Testing Case Generation from Java Source Code

***Abstract*—Software testing, despite its cost, considerably improves the reliability and quality of systems, contributing to their positioning in the market. Specifically, unit testing is the process by which the correct individual functioning of modules, components, and design is ensured. Even though tools that execute unit testing automatically exist, these lack the ability to provide the developer with support and assistance in the design of test cases; furthermore, the current proposals for test case design in unit testing have not been inserted into the production environment and are unable to generate testing code. The present work proposes a tool that allows developers to automatically generate test cases for unit testing from Java source code. In this new proposal the basis path testing technique is used for the design of the test cases. The control flow graph is automatically generated from the source code being tested, in order to subsequently generate the independent paths. Finally, the combinations of test values that satisfy each and every one of the linearly independent paths are generated. In the process of implementing this new tool a case study was designed for the purpose of validation; metaheuristic algorithms were applied to generate test values and value combinations for each path. These combinations were compared against the ones obtained by other state-of-the-art algorithms. Since in this case study a 100% coverage of the independent paths is reached, the proposed tool exhibits competitive results with respect to the ones reported by tools proposed by other authors.**

***Index terms*—unit tests, basic path technique, automatic generation of test cases, metaheuristics algorithms.**

I. Introducción

En la actualidad, la demanda de software ha aumentado considerablemente como consecuencia del avance de la tecnología. De esta forma, se hace necesario que los productos de software obtengan una certificación de calidad siendo la mejor manera de competir en un mercado en crecimiento que es cada vez más exigente [1].

Las pruebas de software continúan ocupando espacio en los trabajos científicos de múltiples investigadores: En particular,

ISSN 2395-8618

se mantienen como problemas abiertos la generación de caminos y valores de pruebas para apoyar el diseño de los casos de prueba [2; 3; 4; 5; 6], así como los procesos vinculados con las pruebas de software [2; 7; 8; 9; 10; 11]. Estas propuestas van desde la utilización de algoritmos de optimización e inteligencia artificial para resolver el problema de la explosión combinatoria de los caminos y valores de pruebas, hasta propuestas de *frameworks* para lograr la automatización de algunos elementos del proceso. En este último caso las propuestas son prototipos para validar la solución teórica, pero no se han incorporado a las soluciones comerciales los elementos de generación de los casos de prueba, de forma tal que puedan ser utilizados por desarrolladores y equipos de probadores, reduciendo así el esfuerzo vinculado con esta actividad de diseño que es altamente costosa. Además, las propuestas mencionadas anteriormente no llegan a la generación de código de pruebas unitarias.

Existen diferentes herramientas como JUnit [12], NUnit [13] y PHPUnit [14] que permiten ejecutar pruebas unitarias de forma automática, pero carecen de funcionalidades que asistan al desarrollador en el diseño de los casos de pruebas. Ello se debe a que, a pesar de que estas herramientas crean automáticamente una clase de prueba con un método de prueba vacío, el desarrollador debe llenar el método de prueba y crear el resto de los métodos que necesite.

Se hace necesario automatizar la generación de casos de pruebas unitarias a partir del código fuente. Para solucionar la problemática existente, el objetivo del presente trabajo es desarrollar una herramienta para la generación de casos de pruebas unitarias a partir del código fuente en lenguaje Java. La herramienta estará insertada en el propio ambiente de desarrollo, por lo que brinda apoyo al programador en cuanto al diseño de los casos de pruebas y se disminuye el tiempo y esfuerzo dedicado a esta tarea.

II. MATERIALES Y MÉTODOS

En el presente trabajo se utilizó la técnica del camino básico para el diseño de los casos de pruebas. Como se ilustra en la Figura 1, esta técnica permite ejecutar todas las instrucciones del código fuente al menos una vez.

Las actividades que aparecen sombreadas en gris corresponden a propuestas que han llegado a soluciones para los entornos productivos, y las que aparecen sombreadas en azul, corresponden a propuestas teóricas que no han sido insertadas en el entorno industrial. Como se puede observar, el diseño de casos de prueba cuenta con algunas propuestas no incorporadas al entorno productivo, por lo que el problema sigue sin resolverse en el entorno de producción; además, no cuenta con herramientas que generen el código de pruebas necesario para su posterior ejecución con las herramientas existentes.

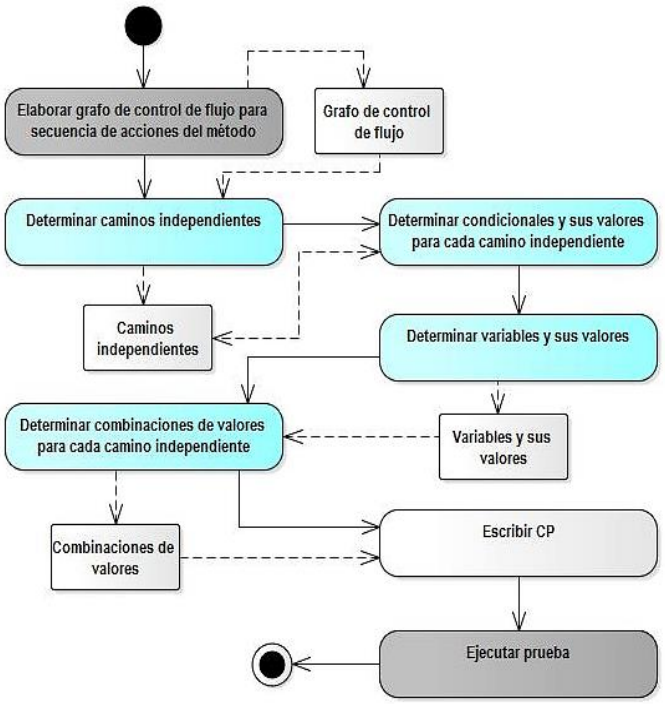


Fig. 1. Procedimiento para realizar pruebas unitarias.

Teniendo en cuenta este procedimiento, en la herramienta desarrollada en el presente trabajo se genera el grafo de control de flujo de forma automática a partir del código fuente de un método en lenguaje Java. Para la generación del grafo de control de flujo se utiliza la herramienta ANTLR (Herramienta para Reconocimiento de Lenguaje).

ANTLR es una herramienta que provee un *framework* para construir reconocedores, compiladores y traductores de descripciones gramaticales para lenguajes de dominio específico. Los lenguajes de dominio específico incluyen formatos de datos, formatos de ficheros de configuración, protocolos de red, lenguajes de procesamiento de texto, secuencia de genes, lenguajes de control de sondeo de espacio, y lenguajes de programación de dominio específico. Tiene soporte de generación de código en diferentes lenguajes de programación, tales como: Java, C#, Python, Ruby, Objective-C, C y C++. Además, permite generar un árbol de sintaxis abstracta (AST por sus siglas en inglés) con la secuencia de acciones del método [15].

Un AST es una representación de árbol de la estructura sintáctica abstracta (simplificada) del código fuente escrito en cierto lenguaje de programación. Cada nodo del árbol denota una construcción que ocurre en el código fuente. La sintaxis es abstracta en el sentido que no representa cada detalle que aparezca en la sintaxis verdadera [15].

En la figura 2 se muestra un diagrama UML con las actividades necesarias para obtener un grafo de control de flujo mediante la utilización de la herramienta ANTLR.

Con objeto de determinar los caminos independientes a partir del grafo de control de flujo, se desarrolló un algoritmo

ISSN 2395-8618

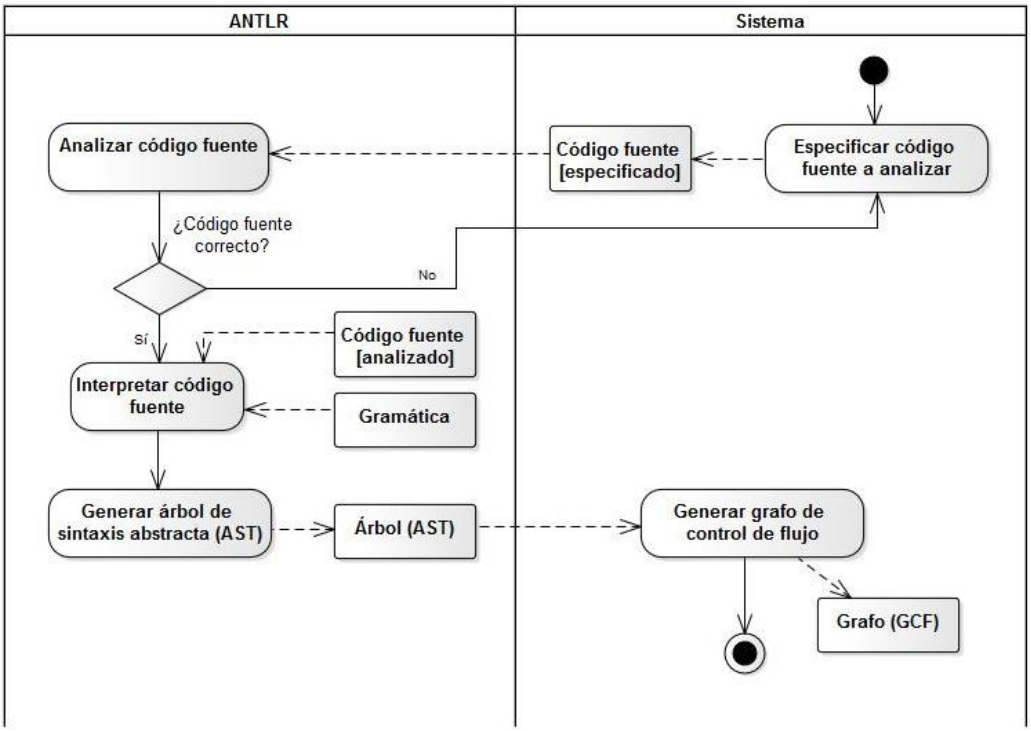


Fig. 2. Flujo de actividades para generar un grafo de control de flujo.

encargado de transformar el grafo de control de flujo a un grafo de condicionales que permite obtener los casos de prueba independientemente del lenguaje del código fuente.

Este algoritmo se describe de la siguiente manera:

1) Procesar sólo los nodos que representen una condicional.

2) En caso de que exista una sentencia *switch*, cambiar a un conjunto de condicionales, donde cada vértice tenga grado de salida dos.

3) En caso de que exista una condición compuesta (esto ocurre cuando uno o más operadores booleanos se presentan en una instrucción condicional), ésta se transforma en varias condicionales simples; para lograrlo, se tienen en cuenta los operadores booleanos asociados a cada una de las condicionales simples.

El grafo de condicionales facilita la obtención de los caminos independientes, debido a que sólo contiene los nodos que generan nuevos caminos, los cuales son precisamente los nodos condicionales. Además, la cantidad de nodos del grafo de condicionales más uno representa la cantidad de caminos independientes (caminos que poseen al menos una nueva arista), y la cantidad de casos de prueba. En la Tabla I se muestran las principales diferencias entre el grafo de control de flujo y el grafo de condicionales.

Una vez transformado el grafo de control de flujo en un grafo de condicionales, se utiliza el algoritmo de búsqueda en profundidad para obtener todos los posibles caminos del grafo, y luego se procede a eliminar los caminos redundantes. De esta forma, sólo se obtienen los caminos independientes.

A partir de los caminos independientes, se obtienen los casos de pruebas (caminos y valores asociados); para ello, se utilizaron algoritmos metaheurísticos de búsqueda para la generación de los valores y sus combinaciones para cada camino, como se indica en [1] y [2].

TABLA I
DIFERENCIAS ENTRE EL GRAFO DE CONTROL DE FLUJO Y EL GRAFO DE CONDICIONALES

Aspectos a tener en cuenta	Grafo de control de flujo	Grafo de condicionales
Vértices del grafo	Todas las instrucciones del código fuente.	Instrucciones que representan una condicional.
Grado de salida de los vértices del grafo	Si el vértice es una instrucción secuencial, grado de salida 1; si es una instrucción condicional, grado de salida 2 o más.	Grado de salida 2 a lo sumo.
Condicionales compuestas	Las instrucciones que representan condicionales compuestas se mantienen igual.	Las instrucciones que representan condicionales compuestas se transforman en condicionales simples, teniendo en cuenta el o los operadores que las relacionan.

III. PROPUESTA DE SOLUCIÓN

Para la generación de casos de pruebas unitarias a partir del código fuente en lenguaje Java, se desarrolló un *plug-in* en el

ISSN 2395-8618

entorno de desarrollo Eclipse, de forma tal que el programador puede seleccionar el método a probar y, en el propio proyecto bajo prueba, se genera una clase que contiene los métodos de prueba del método seleccionado. El método de prueba incluye el caso de prueba que responde a un determinado camino y su valor esperado.

En la Figura 3 se pueden observar, a través de un diagrama UML de casos de uso, las funcionalidades del *plug-in* desarrollado y su relación con los componentes de generación de valores y de combinaciones de valores para cada camino independiente.

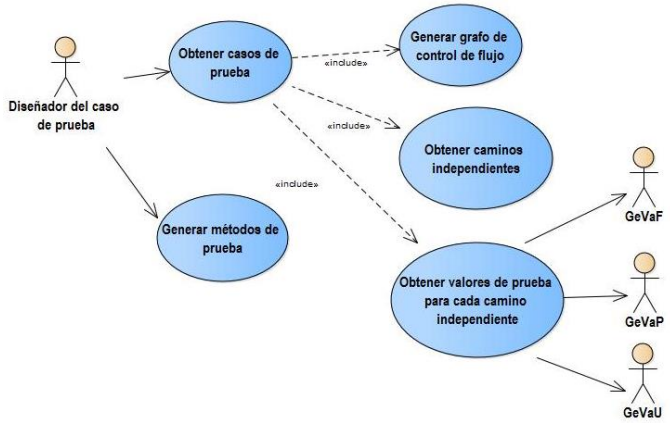


Fig. 3. Funcionalidades del plug-in para generar casos de pruebas unitarias en lenguaje Java.

El *plug-in* permite obtener los casos de pruebas unitarias a partir del código fuente de un método en lenguaje Java, mediante la generación automática de un grafo de control de flujo; luego, se obtienen los caminos independientes.

A fin de obtener las combinaciones de valores de prueba para cada camino, se utilizan tres componentes, GeVaF: encargado de generar valores de pruebas a partir de la descripción del dominio de las variables que intervienen en el grafo de control de flujo; GeVaP: encargado de generar valores de prueba teniendo en cuenta las técnicas de diseño de casos de prueba: de bucles y de condiciones; y GeVaU: encargado de generar combinaciones de valores de prueba para cada camino independiente, a partir de los valores generados por GeVaF y GeVaU.

Posteriormente, se genera un conjunto de métodos de prueba que pueden ser ejecutados con la herramienta JUnit.

IV. RESULTADOS Y DISCUSIÓN

La solución propuesta permite generar casos de pruebas unitarias mediante la utilización de técnicas de diseño de casos de prueba de Ingeniería de Software y algoritmos metaheurísticos. Se realizó una comparación de los valores de prueba generados para cada camino independiente con algoritmos propuestos por autores que trabajan el tema en la comunidad científica; paara ello, se utilizó el problema de clasificación del triángulo.

A continuación se describe el problema de clasificación del triángulo. Los resultados obtenidos se presentan en la tabla II.

TABLA II
RESULTADOS OBTENIDOS AL GENERAR COMBINACIONES DE VALORES PARA EL ALGORITMO DE CLASIFICACIÓN DEL TRIÁNGULO

Algoritmo propuesto por	Cantidad de combinaciones	Tiempo promedio (en segundos)
Jones	17789	8.4
Díaz	587	1.09
Lanzarini	51	1.03
Solución propuesta	5	0.853

El problema de clasificación del triángulo tiene tres variables de entrada (A, B, C) que representan la longitud de los lados de la figura. El programa determina, en cada caso, si la entrada corresponde, o no, a un triángulo; y en caso afirmativo, genera el tipo de triángulo: escaleno, equilátero, isósceles.

En la tabla II se muestra la cantidad de combinaciones que requirió cada uno de los algoritmos que se compararon experimentalmente, luego de realizar 2000 iteraciones en el problema de clasificación del triángulo; se incluye, además, el tiempo promedio que tardó cada propuesta en lograr el 100% de cobertura.

Al aplicar la solución propuesta en el presente trabajo de investigación al problema de clasificación de triángulo, se identificaron 5 caminos de prueba que permiten el 100% de cobertura. Considerado que en el comparativo se invirtió el menor tiempo en segundos (0.853), es evidente la superioridad de los resultados obtenidos con la nueva propuesta sobre los algoritmos respectivos de Jones, Díaz y Lanzarini: la nueva propuesta exhibe cobertura en el 100% de los caminos de prueba, genera un conjunto reducido de valores para esos caminos y, además, obtiene los resultados en menos tiempo que el propuesto en [20], [19], [18].

Adicionalmente a los resultados experimentales previos que muestran la superioridad de la nueva propuesta respecto de los algoritmos del estado del arte, en el presente trabajo de investigación se diseñó un caso de estudio para evaluar el valor práctico de la solución propuesta.

Se describe el contexto utilizado en el caso de estudio, se propone un conjunto de preguntas de estudio y se ilustra cómo la solución propuesta genera respuestas válidas y prácticas a esas preguntas de estudio. El código se incluye en la Figura 4.

Contexto: Para generar el código de prueba del caso de estudio, se utiliza el código fuente del algoritmo para la Serie de Fibonacci (ver Figura 4). El entorno de desarrollo en el que se muestra la solución es Eclipse debido a que el *plug-in* se construyó para ese entorno. Además, la herramienta de ejecución de pruebas unitarias que se seleccionó fue JUnit porque permite realizar pruebas para el lenguaje de programación Java.

ISSN 2395-8618

```java
public int SerieFibonacci (int limiteSerie) {
    int result=-1, a, b;
    if(limiteSerie==0 || limiteSerie==1) {
        result=limiteSerie;
    }
    else {
        a = -1;
        b = 1;
        for(int i = 0; i <= limiteSerie; i++) {
            result = a + b;
            a = b;
            b = result;
        }
    }
    return result;
}
```

Fig. 4. Código fuente del caso de estudio.

Preguntas de estudio y proposiciones:

1) ¿Cómo obtener caminos independientes a partir de código fuente en Java?

Para obtener caminos independientes a partir de código fuente en Java es necesario contar con un grafo de control de flujo del código fuente en Java, el cual es generado por el *plug-in* diseñado en el presente trabajo de investigación; luego, se generan los caminos independientes, a partir del grafo de control de flujo. En la Figura 5 se pueden observar los caminos generados.

Caminos independientes:

Caminos/Condiciones	limiteSerie==0	limiteSerie==1	i<=limiteSerie
C1	T	-	-
C2	F	T	-
C3	F	F	T
C4	F	F	F

Fig. 5. Caminos independientes generados por el *plug-in*.

2) ¿Cómo obtener casos de pruebas unitarias de forma automática para ejecutar todas las instrucciones del código fuente?

Una vez generados los caminos independientes, se generan los valores interesantes con los componentes GeVaF y GeVaP. Posteriormente, a partir de los caminos, las condiciones y los valores interesantes, se generan las combinaciones de valores de prueba para cada camino independiente mediante el uso del componente GeVaU. En la Figura 6 se muestran los casos de prueba generados (caminos y valores asociados).

Como se puede observar en la figura anterior, el diseñador del caso de prueba debe especificar el resultado esperado en cada caso de prueba, teniendo en cuenta la combinación de valores de prueba generada para cada camino independiente.

Casos de prueba:

Caminos/Variables	limiteSerie	Valor esperado
1	0	0
2	1	1
3	5	5
4	-1	-1

Fig. 6. Casos de pruebas (caminos y valores asociados) generados por la herramienta.

3) ¿Cómo insertar los casos de pruebas unitarias en un entorno de prueba específico?

Para insertar los casos de pruebas unitarias en un entorno de prueba específico se utiliza el *plug-in* desarrollado en el presente trabajo, lo cual se llevó a cabo en el entorno Eclipse para el lenguaje Java.

En caso que se quieran insertar los casos de pruebas en otros entornos, bastaría con desarrollar un *plug-in* para el entorno requerido, siempre y cuando lo permita. Se recomienda el desarrollo de un *plug-in* debido a que facilita el trabajo del diseñador del caso de prueba; se debe, además, seleccionar la herramienta a utilizar para realizar las pruebas unitarias.

A diferencia de la herramienta JUnit, que solamente genera un método de prueba vacío (ver Figura 7), el *plug-in* desarrollado ofrece la implementación de los métodos de prueba necesarios para satisfacer cada camino independiente.

```java
package pruebas_unitarias;

import static org.junit.Assert.*;

import org.junit.Test;

public class TestEjemploTesisJUnit {

    @Test
    public void testIsPrime() {
        fail("Not yet implemented");
    }

}
```

Fig. 7. Método de prueba creado por JUnit.

En la Figura 8 se muestran los métodos de prueba generados por el *plug-in*, a partir de los casos de pruebas obtenidos previamente para ejecutarlos con JUnit.

Como se puede observar en la figura anterior, el diseñador del caso de prueba debe especificar el resultado esperado en cada caso de prueba, teniendo en cuenta la combinación de valores de prueba generada para cada camino independiente.

Como se puede observar, se genera un método de prueba por cada caso de prueba obtenido, teniendo en cuenta la combinación de valores generada para cada camino, además del resultado esperado especificado previamente.

ISSN 2395-8618

```java
package pruebas_unitarias;

import org.junit.Assert;
import org.junit.Test;
import ejemploTesis.EjemploTesis1;

public class TestEjemploTesis1  {

    //Este método de prueba hace referencia al camino C1: limiteSerie==0,
    @Test
    public void test_SerieFibonacci_CP1() {
        EjemploTesis1 ejemploTesis1 = new EjemploTesis1();
        int expected = 0;
        Assert.assertEquals(expected, ejemploTesis1.SerieFibonacci(0));
    }
    //Este método de prueba hace referencia al camino C2: limiteSerie==0,
    @Test
    public void test_SerieFibonacci_CP2() {
        EjemploTesis1 ejemploTesis1 = new EjemploTesis1();
        int expected = 1;
        Assert.assertEquals(expected, ejemploTesis1.SerieFibonacci(1));
    }
    //Este método de prueba hace referencia al camino C3: limiteSerie==0,
    @Test
    public void test_SerieFibonacci_CP3() {
        EjemploTesis1 ejemploTesis1 = new EjemploTesis1();
        int expected = 5;
        Assert.assertEquals(expected, ejemploTesis1.SerieFibonacci(5));
    }
    //Este método de prueba hace referencia al camino C4: limiteSerie==0,
    @Test
    public void test_SerieFibonacci_CP4() {
        EjemploTesis1 ejemploTesis1 = new EjemploTesis1();
        int expected = -1;
        Assert.assertEquals(expected, ejemploTesis1.SerieFibonacci(-1));
    }
}
```

Fig. 8. Código de pruebas generado por la herramienta JUnit.

4) ¿Cómo ejecutar pruebas unitarias en un entorno específico?

Para ejecutar pruebas unitarias en un entorno específico, se hace necesario utilizar una herramienta para realizar pruebas unitarias para un determinado lenguaje. En la solución propuesta se utiliza la herramienta JUnit porque ejecuta pruebas unitarias para el lenguaje de programación Java, lenguaje para el cual fue creado el plug-in. En la Figura 9 se muestran los resultados de una ejecución de los métodos de prueba generados por el *plug-in*.

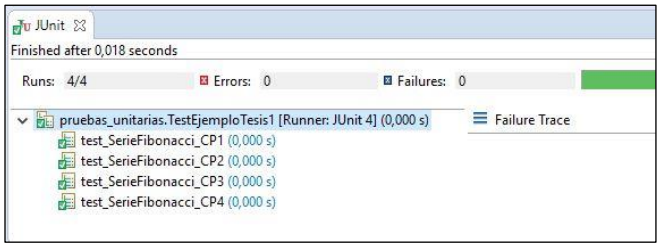


Fig. 9. Ejecución del código de prueba generado con la herramienta JUnit.

V. CONCLUSIONES

Los resultados del presente trabajo de investigación facilitan el proceso de pruebas unitarias durante el desarrollo de productos de software, debido a que permiten automatizar el diseño de casos de pruebas unitarias y la generación de código de pruebas unitarias en lenguaje Java. Adicionalmente, estos resultados permiten reducir el tiempo y esfuerzo dedicados por desarrolladores, probadores y diseñadores de casos de prueba en el diseño y ejecución de pruebas unitarias. Debido a que los valores de prueba generados tienen en cuenta las técnicas de bucles y condicionales, las combinaciones de valores generados satisfacen todos los caminos independientes. De esta forma, se alcanza un 100% de cobertura de caminos independientes permitiendo ejecutar cada instrucción del código fuente del método a probar, con el objetivo de detectar errores.

REFERENCIAS

[1] Equipo del Producto CMMI, "CMMI para Desarrollo, Versión 1.3," CMMI-DEV, V1.3. Software Engineering Institute, Hanscom AFB, Massachusetts, Tech. Rep. ESC-TR-2010-033, Nov. 2010.

[2] M. B. Chrissis, M. Konrad, and S. Shrum, *CMMI for Development. Guidelines for Process Integration and Product Improvement.* 3rd. ed., USA: Pearson Education, 2011, pp. 123–135.

[3] S. Sekhara, M. L. Hary, U. Kiran, et al. (2012, marzo). Automated Generation of Independent Paths and Test Suite Optimization Using Artificial Bee Colony. *Procedia Engineering.* [Online]. *30*, pp. 191-200. Available: isiarticles.com/bundles/Article/pre/pdf/7408.pdf

[4] A. Pachauri and G. Srivastava. (2013, enero). Automated test data generation for branch testing using genetic algorithm: An improved approach using branch ordering, memory and elitism. *The Journal of Systems and Software.* [Online]. *86(5)*, pp. 1191-1208. Available: https://www.researchgate.net/publication/256991955_Automated_test_data_generation_for_branch_testing_using_genetic_algorithm_An_improved_approach_using_branch_ordering_memory_and_elitism

[5] P. Ranjan, B. Mallikarjun and X. Yang. (2012, septiembre). Optimal test sequence generation using firefly algorithm. *Swarm and Evolutionary Computation.* [Online]. *8*, pp. 44-53. Available: https://pdfs.semanticscholar.org/bbdc/692a58b3517d66b4b0e000a7e0fcb8cc9e3d.pdf

[6] G. Carvalho, D. Falcão, F. Barros, et al. (2014, junio). NAT2TESTSCR: Test case generation from natural language requirements based on SCR specifications. *Science of Computer Programming.* [Online]. *95(3)*, pp. 275-297.

[7] I. Hermadi, C. Lokan and R. Sarker. (2014, enero). Dynamic stopping criteria for search-based test data generation for path testing. *Information and Software Technology.* [Online]. *56(4)*, pp. 395-407. Available: https://www.researchgate.net/publication/260009614_Dynamic_Stopping_Criteria_for_Search-based_Test_Data_Generation_for_Path_Testing

[8] F. Elberzhager, A. Rosbach, J. Münch and R. Eschbach. (2012, mayo). Reducing test effort: A systematic mapping study on existing approaches. *Information and Software Technology.* [Online]. *54(10)*, pp. 1092-1106. Available: http://www.juergenmuench.com/publications/uploads/cd60a34afaa5f50e7422113e7d8941ef4ce84456.pdf

[9] T. Rongfa, "Adaptive Software Test Management System Based on Software Agents," in Advanced Technology in Teaching - Proceedings of the 2009 3rd International Conference on Teaching and Computational Science (WTCS 2009), vol. 117, Y. Wu, E d . Berlin: Springer Berlin Heidelberg, 2012, pp. 1–9.

[10] T. Chen, X. Zhang, S. Guo, et al. (2012, marzo). State of the art: Dynamic symbolic execution for automated test generation. *Future Generation Computer Systems.* [Online]. *29(7)*, pp. 1758-1773. Available: https://pdfs.semanticscholar.org/02af/b1a4a45cfea0c1aeffca4a441e6541a5d34b.pdf

[11] X. Ying, G. Yun-zhan, W. Ya-wen and Z. Xu-zhou. (2014, abril). Intelligent test case generation based on branch and bound. *The Journal of China Universities of Posts and Telecommunications.* [Online]. *21(2)*, pp. 91-97. Available:

ISSN 2395-8618

http://www.juergenmuench.com/publications/uploads/cd60a34afaa5f50e7422113e7d8941ef4ce84456.pdf

[12] JUnit. (2017, Enero). Sitio official de JUnit. [Online]. Available: www.junit.org

[13] NUnit. (2015). Sitio official de NUnit. [Online]. Available: www.nunit.org

[14] S. Bergmann. (2017). Sitio official de PHPUnit. [Online]. Available: http://phpunit.de/

[15] T. Parr, "The Definitive ANTLR Reference," Texas, USA: Pragmatic Bookshelf, 2007, pp. 15–17.

[16] A. Macías, M. D. Delgado, J. Fajardo and D. Larrosa. (2016, enero-junio). Generador de valores de casos de pruebas funcionales. *Lámpsakos.* [Online]. *15,* pp. 51-58. Available: https://dialnet.unirioja.es/descarga/articulo/5403332.pdf

[17] P. B. Fernández, W. Cantillo, M. D. Delgado, et al. (2016, mayo-agosto). Generación de combinaciones de valores de pruebas utilizando metaheurísticas. *Ingeniería Industrial.* [Online]. *36(2),* pp. 200-207. Available: http://www.redalyc.org/pdf/3604/360446197009.pdf

[18] B. F. Jones, H. -H. Sthamer and D. E. Eyres. (1996, septiembre). Automatic structural testing using genetic algorithms. *Software Engineering Journal.* [Online]. *11(5),* pp. 299-306. Available: ieeexplore.ieee.org/iel1/2225/11679/00533215.pdf

[19] E. Díaz, J. Tuya, R. Blanco and J. J. Dolado. (2008, octubre). A tabu search algorithm for structural software testing. *Computers & Operations Research.* [Online]. *35(10),* pp. 3052-3072. Available: giis.uniovi.es/testing/papers/caor-2007-tabustesting.pdf

[20] L. C. Lanzarini, and P. E. Battaiotto. (2010, junio). Dynamic generation of test cases with metaheuristics. *Journal of Computer Science & Technology.* [Online]. *10(2),* pp. 91-96. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/21338/Documento_completo.pdf?sequence=1

ISSN 2395-8618

Naïve Screw Nut Classifier Based on Hu's Moment Invariants and Minimum Distance

Antonio Alarcón-Paredes, Roberto Contreras-Garibay,
Gustavo Adolfo Alonso-Silverio and Eric Rodríguez-Peralta

***Abstract*—In this paper, an algorithm for classification of screw nuts by means of digital image processing is presented. This work is part of a project where a production line was built, and is focused on the quality assessment section. The algorithm presented classifies among good and poor quality screw nuts passing by a conveyor belt, by computing Hu's moment invariants of its picture. Those moment invariants are the input of a minimum distance classifier, obtaining very competitive results compared with some other classification algorithms of the WEKA plattform.**

***Index Terms*—Classification algorithms, Computer vision, Manufacturing automation, Pattern recognition.**

I. Introduction

THE computer vision techniques have been developed since 1960's, and continued growing as in theory and applications [1], [2]. Nowadays, these techniques are used in a wide range of applications, such as medical imaging [3]-[5], industry automation [6], [7], monitoring [8], food quality [9]-[11], quality assessment [12], [13], among others [14], [15]. The product quality depends on how the industry processes are performed. A systematic inspection of these tasks have been usually done by humans, however, in some cases they could incur in errors due to fatigue and psychological or health factors; these errors make a computer vision system more attractive [16], [17].

With the increasing volume production, it is necessary to create strategies to achieve quality products at large scale in less time. For that reason, manufacturing companies have chosen computer vision automation as the solution of the problem established above [18].

Manuscript received on December 30, 2014, accepted on December 2, 2017, published on June 30, 2018.

A. Alarcón-Paredes is with the Universidad Autónoma de Guerrero, Chilpancingo, Guerrero. 39060 México (e-mail: aalarcon@uagro.mx).

J. R. Contreras-Garibay, is Computer Engineer from the Universidad Autónoma de Guerrero, Chilpancingo, Guerrero. 39060, México (e-mail: jroberto@hotmail.com).

G. A. Alonso-Silverio is with the Universidad Autónoma de Guerrero, Chilpancingo, Guerrero. 39060 México (corresponding author to provide phone: +5217471122838; e-mail: (gsilverio@uagro.mx).

E. Rodríguez-Peralta is with the Universidad Autónoma de Guerrero, Chilpancingo, Guerrero, 39060 México (e-mail: erodriguez@uagro.mx).

Automation is a key factor to improve the production lines in order to stay alive in the competitive production market. For this reason, an automatic system for object inspection could be implemented on the quality assessment in industry.

The main idea is that a computer vision system performs the following whole process: from image acquisition, feature extraction and image analysis, to finally classify among good and poor quality objects [19].

In this paper, a computer vision system for nut quality control in industry is presented. This system is former part of a project where a conveyor belt built with Lego Mindstorms NXT kit is used [20], however, this work focuses only in the quality assessment part.

The proposed algorithm classifies between good and bad quality nuts. For its classification, the system takes a picture of a nut passing by a conveyor belt, applies some preprocessing to the image, and then computes the Hu's moment invariants [21]. The values of the seven moments constitute the input data to the algorithm, which is a minimum distance classifier, and uses the Euclidian distance [22].

The results obtained with the proposed algorithm are very competitive with a variety of classification algorithms included in the WEKA open source platform [23].

II. Theoretical Support

A. Image acquisition

One of the main problems when acquiring an image is the different lighting and brightness condition of the environment, since a proper light permits to obtain a good quality image [24]. Here, that issue is simply solved by using a light-controlled chamber, with infrared sensors and a webcam placed within.

In order to take the pictures of the nuts, the system uses the infrared sensors placed beside the conveyor belt; once the sensors detect and object passing by, the webcam automatically receive a signal to take an RGB color picture.

Once the image is stored, it is cropped into a specific area; the boundaries of this area were chosen in an experimental way, and the nut is always into this area. The cropped image is now converted to a gray scale image. The latter steps help

ISSN 2395-8618

the image processing to be more efficient.

B. Spatial filtering

In general, spatial filtering of an image $f(x,y)$ of size $M \times N$ with a filter mask $h(s,t)$ of size $m \times n$ is given by the expression:

$$g(x,y) = \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} f(x+s, y+t) h(s,t) \qquad (1)$$

where $x = 0,1,\ldots,M-1$ and $y = 0,1,\ldots,N-1$.

In this work, the expression (1) and a Gaussian mask are used to smooth the image and for noise reduction. The next step consists in obtain the Sobel gradient ∇f of the image by means of the components *Gx* and *Gy*, computed using (1) and the *x*-direction and *y*-direction Sobel masks.

C. Feature extraction and Hu's moment invariants

In [3]-[16], the feature extraction process is divided into different steps, and a variety of algorithms for this purpose and some other transformations are used, such as the conversion to other color spaces than RGB, the FFT, PSO, PCA, Markov Chains, Kalman filters, Canny edge detector, among others. The work presented in this paper is called a *naïve classifier*, since instead of using complex algorithms for feature extraction, uses a simple preprocessing for noise reduction, the image gradient, and then, Hu's moment invariants are obtained. These moments could be computed as follows:

The two-dimensional (*p*+*q*)th order moment are given by:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x,y) \qquad (2)$$

where $p, q = 0,1,2,3,\ldots$

Some invariant features can be achieved using the central moments, which are computed with the following equation:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad (3)$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$, and the point $(\bar{x}, \bar{y})$ is the centroid of the image $f(x,y)$.

Scale invariance could be obtained by normalization. Thus, the normalized moments are described by

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}} \qquad (4)$$

Finally, by means of (2), (3), and (4), the seven Hu's moment invariants are computed as follows:

$$\phi_1 = \eta_{20} + \eta_{02} \qquad (5)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \qquad (6)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \qquad (7)$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \qquad (8)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12}) - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})\right] \qquad (9)$$

$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03}) \qquad (10)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \qquad (11)$$

The Hu's moment of all the training images are obtained and used to train the algorithm, such that each image can be represented by a seven-dimensional vector, *i.e.*, the value of the seven moment invariants. The patterns which represent the training images are stored into a *comma separated values* (.CSV) text file in order to be used by the proposed algorithm; these patterns are also stored in an .ARFF file, this file is used by the WEKA data mining software.

D. Fundamental set of patterns

In this stage it becomes necessary to obtain the algorithm training set. Let the input patterns be represented by column vectors **x** of size *n*, and the associated class is represented by c, with $c \in \{0,1\}$ since there are only two output classes: good quality nuts or poor quality nuts. Each input pattern $\mathbf{x}^k$ is corresponded to one and only one output class c^k forming thus the association of the ordered pair: $(\mathbf{x}^k, c^k)$. The set of *p* associations of input patterns $\{(\mathbf{x}^1, c^1), (\mathbf{x}^2, c^2), ..., (\mathbf{x}^p, c^p)\}$ is called the fundamental set, and is represented as

$$\{(\mathbf{x}^k, c^k) \mid k = 1, 2, ..., p\} \qquad (12)$$

E. Euclidian distance

For measuring similarity between patterns, the Euclidian

ISSN 2395-8618

distance is used as follows:

$$d\left(\mathbf{x}^{\omega},\mathbf{x}^{k}\right)=\left[\sum_{i=1}^{n}\left|x_i^{\omega}-x_i^{k}\right|^{2}\right]^{\frac{1}{2}} \tag{13}$$

Note that to obtain the minimum distance, the use of the squared Euclidian distance is sufficient, thus:

$$d^{2}\left(\mathbf{x}^{\omega},\mathbf{x}^{k}\right)=\sum_{i=1}^{n}\left|x_i^{\omega}-x_i^{k}\right|^{2} \tag{14}$$

III. Classification Algorithm

The proposed algorithm is described as follows:

1) Obtain an RGB picture of the nut in the conveyor belt (Fig. 1).
2) Crop the image for efficiency and obtain the gray level image of the nut, as shown in Fig 2.
3) In Fig. 3, the noise reduction of the image by means of a Gaussian filter is shown.
4) Obtain the Sobel gradient of the filtered image resulting in previous step (See Fig. 4).
5) Get the seven Hu's moment invariants as established in equations (5) to (11). From now, these moments may be referred as patterns, or image patterns.

Fig. 1. RGB image acquired with a conventional webcam.

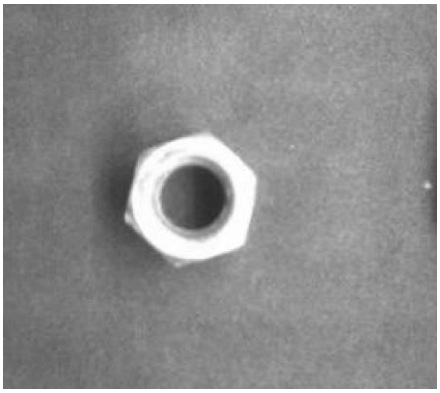


Fig. 2. Cropped grayscale image.

6) Compute the distance vector dv of size p (the same as the cardinality of the fundamental set) with the Euclidian distance between the test image patterns and each of the training patterns:

$$dv_k = d^{2}\left(\mathbf{x}^{\omega},\mathbf{x}^{k}\right) \tag{15}$$

$$dv_k = \sum_{i=1}^{n}\left|x_i^{\omega}-x_i^{k}\right|^{2} \tag{16}$$

where $\mathbf{x}^{\omega}$ is the pattern of an unknown image, $\mathbf{x}^{k}$ are the p patterns in the fundamental set, and $k=1,2,...,p$. The distance vector is then normalized.

7) It is necessary to choose a classification threshold denoted by θ, *i.e.*, the greater distance that could exist between two patterns of the same class. This value, which can go from 0 to 1 due to the normalization of vector dv, was obtained by experimentation and varies when the cardinality of the fundamental set changes.

8) Look for the smallest value in the distance vector dv:

$$\varepsilon = \min_{i}\left(dv_i\right) \tag{17}$$

9) Obtain the class c^{ω} for the correspondent pattern $\mathbf{x}^{\omega}$:

$$c^{\omega}=\begin{cases}1 & \text{if } \varepsilon \le \theta \\ 0 & \text{other case}\end{cases} \tag{18}$$

where a value of $c^{\omega}=1$ represents that the nut is a good quality one; otherwise, means that the object in the image could be a poor quality nut or even a strange object.

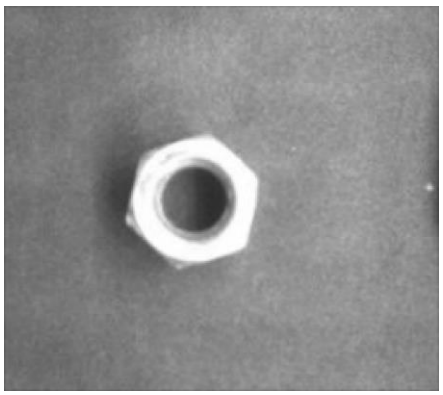


Fig. 3. The image in the Fig. 2 filtered with a gaussian.

ISSN 2395-8618

TABLE I
CLASSIFICATION OF THE WHOLE FUNDAMENTAL SET

Classifier	% Performance
Naïve Bayes	84.67%
Bayes Net	87.33%
Logistic Regression	94.67%
Simple Logistic	94.00%
1-NN	**100.00%**
3-NN	96.67%
5-NN	96.67%
AdaBoostM1	99.33%
LogitBoost	97.33%
Bagging	96.67%
PART	99.33%
C4.5	98.67%
Proposed	**100.00**%
AVERAGE	95.80 %

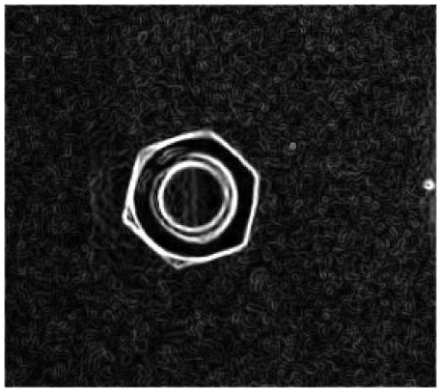


Fig. 4. Image gradient obtained by means of Sobel filter.

IV. RESULTS AND DISCUSSION

A. Preliminars

The proposed algorithm was developed using MATLAB R2013b. A total of 100 pictures of good quality nuts, and 50 poor quality nuts were obtained; all of them were taken on the conveyor belt running.

When speaking of distance function-based classification, it is inevitable to talk about the k-Nearest Neghbours [25], which is a non-parametric *lazy* algorithm. In this paper, the proposed algorithm depends on two principal factors: the first is the cardinality p of fundamental set, *i.e.*, the number of good quality nuts to be compared with nuts to classify, and secondly the similarity threshold θ between the training patterns and the unknown ones. For this reason, a comparative study with different number of p is presented. Although, we run an exhaustive procedure to find the optimal threshold θ for each different value of p. This procedure is carried on only once just when the value of p changes, and can be seen as a system *calibration* process.

TABLE II
RESULTS OF PROPOSED ALGORITHM

p (% hold out)	θ	% Performance
5 (3.33%)	0.85	89.65%
10 (6.67%)	0.68	90.71%
20 (13.33%)	0.59	90.77%
30 (20.00%)	0.46	90.00%
40 (26.67%)	0.40	89.10%
50 (33.33%)	0.35	91.00%

Please note that the distances computed and stored in the distance vector dv are normalized, so the exhaustive procedure must only find values of θ between 0 and 1, which makes this process run faster than it seems.

B. Results classifying the whole fundamental set

The first estimate of the algorithm performance was made by learning and classifying the whole fundamental set entirely.

The WEKA platform was chosen to compare the proposed algorithm with some other classifiers on the state of the art. The classifiers selected were: Naïve Bayes, Bayes Net, Logistic Regression, Simple Logistic, k-NN ($k = 1, 3, 5$), AdaBoostM1, LogitBoost, Bagging, PART, and C4.5.

The process was applied to the proposed algorithm and to 10 classifiers included in WEKA. Results show that only the proposed classifier and the 1-NN can classify the whole fundamental set without ambiguity, *i.e.*, they classify the 100% of patterns.

Table I shows the results of classification with proposed algorithm and the other 10 selected algorithms.

C. Comparison between algorithm proposed and WEKA algorithms.

The second estimation of the performance was carried out by learning 5 patterns and classifying 145, then learn 10 and classify 140, learn 20 and classify 130, learn 30 and classify 120, learn 40 and classify 110, and finally learn 50 and classify the other 100.

Since the proposed algorithm was tested with different values of p: 5, 10, 20, 30, 40 and 50, also needs different values of the threshold θ which were: 0.85, 0.68, 0.59, 0.46, 0.4 and 0.35, respectively.

The election of the p nut pictures was done randomly, and takes only good quality ones. Notice that the value of p is inversely proportional to the threshold. It means that if there are few nuts to compare with, the algorithm must give a greater margin of similarity between patterns; but if there are many nuts to compare, should give a lower threshold value.

Remember that there are 150 nuts pictures in total, the first 100 are good quality nuts, and the other 50 are poor quality

ISSN 2395-8618

TABLE III
COMPARISON BETWEEN PROPOSED ALGORITHM AND SOME OF THE WEKA CLASSIFIERS, USING HOLD-OUT

Classifier	Performance of the classifier using different values for p (% of hold-out partitions)					
	p=5 (3.33%)	p=10 (6.67%)	p=20 (13.33%)	p=30 (20.00%)	p=40 (26.67%)	p=50 (33.33%)
Naïve Bayes	**90.34%**	81.43%	*90.77%*	80.00%	89.09%	76.00%
Bayes Net	82.07%	**92.14%**	88.46%	**90.00%**	88.18%	***94.00%***
Logistic Regression	85.52%	**92.86%**	***93.85%***	88.33%	**91.82%**	87.00%
Simple Logistic	85.52%	86.43%	**92.31%**	85.00%	**91.82%**	***93.00%***
1-NN	86.21%	***96.43%***	**96.15%**	**91.67%**	**91.82%**	91.00%
3-NN	**90.34%**	86.43%	84.62%	84.17%	**92.73%**	***95.00%***
5-NN	66.90%	78.57%	86.15%	84.17%	**92.73%**	***95.00%***
AdaBoostM1	**86.90%**	88.57%	87.69%	86.67%	***90.91%***	88.00%
LogitBoost	**86.90%**	88.57%	87.69%	86.67%	***90.91%***	90.00%
Bagging	66.90%	90.71%	90.00%	**90.83%**	88.18%	*91.00%*
PART	77.93%	87.14%	86.15%	85.00%	90.00%	*92.00%*
C4.5	77.93%	87.14%	86.15%	85.00%	90.00%	*92.00%*
Proposed	**89.65%**	90.71%	90.77%	**90.00%**	89.10%	*91.00%*
AVERAGE	82.55 %	88.24%	89.29%	86.73%	*90.56%*	90.38%

ones. Nevertheless, the algorithm is trained with the p patterns in fundamental set, and tries to classify the other 150-p patterns; this constitutes a hold out cross-validation algorithm. The performance of the algorithm can be seen in Table II.

The same process was applied to 10 other algorithms in the WEKA platform to be compared with the proposed algorithm. The classifiers selected were: Naïve Bayes, Bayes Net, Logistic Regression, Simple Logistic, k-NN ($k = 1, 3, 5$), AdaBoostM1, LogitBoost, Bagging, PART, and C4.5. In order to make a good comparison, the classifiers selected were tested using the same hold-out partitions used with the proposed algorithm. The results of this experiment are shown in Table III, in which the values with italic style represent the best performance for the classifier in that row, and the bold style values are the top five performances for that value of p.

Notice that the proposed algorithm outperforms the average performance for all cases except the case where p=40.

V. CONCLUSION

A naïve classifier based on Hu's moments invariants and minimum distance has been presented.

This classifier was tested and compared to other 10 algorithms in the state of the art which are included in WEKA. The results from Table I to Table III shows that the proposed algorithm overcomes the performance of some well-known algorithms in the literature, such as Naïve Bayes, Logistic Regression, k-NN, C4.5 trees, among others. It is clear that some classifiers present a better performance in some cases, although, the proposed algorithm comes close to their results. This behavior is not strange, since the No Free Luch Theorem [26] shows that when a classifier is very good with some family of problems, may be not so good to others.

However, this drawback can be overcome with the simplicity of the algorithm presented in this work.

It is worth to mention that the proposed algorithm is very competitive among state of the art classifiers, and it is possible to be implemented on industry since it has the possibility to be implemented on a single board computer, such as the Raspberry Pi.

ACKNOWLEDGEMENT

This work was supported in part by the Engineering Faculty of the Universidad Autónoma de Guerrero.

REFERENCES

[1] G. A. Baxes, "*Digital image processing: Principles and applications*" New York: John Wiley & Sons, Inc. 1994.

[2] B. Zhang, *et al*, "Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review" *Food Research International*, vol. 62, pp. 326–343, Aug. 2014.

[3] L. Wen, X. Wang, Z. Wu, M. Zhou, and J. S. Jin. "A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization" *Neurocomputing*, vol. 148, pp. 569-577, Available online July 2014, to be published in 2015.

[4] M. Veta, *et al*. "Assesment of algorithms for mitosis detection in breast cancer histopathology images" *Medical Image Analysis*, Available online Nov 2014, to be published.

[5] C. Petitjean, *et al*. "Right ventricle segmentation from cardiac MRI: A collation study" *Medical Image Analysis*, vol. 19, pp. 187-202, Available online Oct. 2014, to be published in 2015.

[6] M. A. Montironi, P. Castellini, L. Stroppa, and N. Paone "Adaptive autonomous positioning of a robot vision system: Application to quality control on production lines" *Robotics and Computer-Integrated Manufacturing*, vol. 30, pp. 489-498, Apr. 2014.

[7] Y. Liu, S. Q. Li, J. F. Wang, H. Zeng, and J. P. Lu. "A computer vision-based assistant system for the assembly of narrow cabin products" *The International Journal of Advanced Manufacturing Technology*, Aug. 2014.

[8] A. T. Fleury, F. C. Trigo, and F. P. R. Martins. "A new approach based on computer vision and non-linear Kalman filtering to monitor the nebulization quality of oil flames" *Expert Systems With Applications*, vol. 40, no. 12, pp. 4760-4769, 2013.

[9] T. Pérez-Palacios, D. Caballero, A. Caro, P. G. Rodríguez, and T. Antequera. "Applying data mining and Computer Vision Techniques to MRI to estimate quality traits in Iberian hams" *Journal of Food Engineering*, vol. 131, pp. 82-88, Jan. 2014.

ISSN 2395-8618

[10] B. Pace, M. Cefola, P. Da Pelo, F. Renna, and G. Attolico. "Non-destructive evaluation of quality and ammonia content in whole and fresh-cut lettuce by computer vision system" *Food Research International*, vol. 64, pp. 647-655, Aug. 2014.

[11] H. Hong, X. Yang, Z. You, and F. Cheng. "Visual quality detection of aquatic products using machine vision" *Aquacultural Engineering*, vol. 63, pp. 62-71, Oct. 2014.

[12] P. Bellini, I. Bruno, and P. Nesi. "A distributed system for computer vision quality control of clinched boards" *Real-Time Imaging*, vol. 10, pp. 161-176, 2004.

[13] E. Asoudegi, and Z. Pan. "Computer vision for quality control in automated manufacturing systems" *Computers and Industrial Engineering*, vol. 21, pp. 141-145, 1991.

[14] H. Zhang, and D. Li. "Applications of computer vision techniques to cotton foreign matter inspection: A review" *Computers and Electronics in Agriculture*, vol. 109, pp. 59-70, Sept. 2014.

[15] K. L. Lin, and J. L. Fang. "Applications of computer vision on tile alignment inspection" *Automation in Construction*, vol. 35, pp. 562-567, Feb. 2013.

[16] H. Golnabi and A. Asadpour, "Design and application of industrial machine vision systems" *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 6, pp. 630–637, Dec. 2007.

[17] C. Anagnostopoulos, D. Vergados, E. Kayafas, V. Loumos and G. Stassinopoulos. "A computer vision approach for textile quality control" *The Journal of Visualization and Computer Animation*, vol. 12, no. 1, pp. 31–44, Feb. 2001.

[18] M. A. Jimenez, *et al*. "Automation and parameters optimization in production line: a case of study" *The International Journal of Advanced Manufacturing Technology*, vol. 66, pp. 1315-1628, Jun. 2013.

[19] G. J. Awcock and R. Thomas. "*Applied image processing*". London: MacMillan New Press Ltd. 1995.

[20] *Mindstorms NXT 2.0*, LEGO Group TM. 1999 http://shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547

[21] M. K. Hu. "Visual Pattern Recognition by Moment Invariants" *IRE Transactions on Information Theory*, pp. 179-187, Feb. 1961.

[22] C. Yáñez-Márquez, and J. L. Díaz-de-León. "*Minkowski's Norms and Metrics*" (in Spanish: "*Normas y métricas de Minkowski*"), México: technical report, Center for Computing Research, 2003.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. "The WEKA Data Mining Software: An Update; SIGKDD Explorations", vol. 11, no. 1, 2009. Available: www.cs.waikato.ac.nz/ml/weka/

[24] R. C. González, and R. E. Woods. "*Digital Image Processing*", New Jersey: Prentice Hall, 2002.

[25] E. Fix, and J. L. Hodges Jr. "Discriminatory analysis. Nonparametric discrimination: Consistency properties" *No. Project 21-49-004, Report Number 4*, pp. 261-279, 1951.

[26] D. H. Wolpert, and W. G. Macready. "No Free Lunch Theorems for optimization *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 67-82. 1997.

ISSN 2395-8618

Recommender System for Tourist Itineraries Based on Aspects Extraction from Reviews Corpora

Liliya Volkova, Elena Yagunova, Ekaterina Pronoza, Alexandra Maslennikova, Danil Bliznuk, Margarita Tokareva, and Ali Abdullaev

***Abstract*—In this paper a recommender system is described which takes a set of venue categories of user's interest into account to form a tourist itinerary throughout a city. The system is focused on user preferences in venue aspects. Techniques of such aspects extraction are developed in this paper, in particular from reviews corpora. User preferences are used to weigh aspects associated with particular sights and restaurants. These filtered venues along with time restrictions are subject to submit into the recommender system. A lightweight ontology is discussed which describes the domains of restaurants and sightseeing knowledge and allows venues comparative analysis to enhance the search for relevant venues. The system designed performs automated planning of tourist itineraries, flexible sights searching, and analysis of venues aspects extracted from reviews in Russian.**

***Index Terms*—Information extraction, lightweight ontology, natural language processing, recommender systems.**

I. Introduction

SUBJECT area of this research is a recommender system for tourist itineraries planning. Provided with venue reviews corpora, the analyzer component extracts aspects defined for museums and restaurants. A lightweight ontology is described, which serves as a semantic resource for estimating venues for a narrow search and further thematic planning. With support of the lightweight ontology, the recommender system forms a route over a selected set of venue categories (e.g., visiting two museums, then a restaurant, then another museum) basing on user preferences for previously extracted aspects. A flexible recommender engine is designed which generates relevant itineraries throughout the city, each accompanied with a map-based route.

The rating approach is the one avoided in this research: generalized ratings are widespread, but apparently not always exact they are. The task of verifying and attributing ratings itself requires a separate study. Moreover, user preferences differ, which is not reflected by ratings on the whole. Therefore, the thematic recommender system is in the focus of this article, including techniques of venue analogs selection.

One of the most popular web search queries is for tourism and trip planning; hence an automated trip planner is in need of. Most existing resources have a number of limitations. According to a survey undertaken, they contain static information only [36], either no thematic routes planning [38], or fixed routes [18]. The most complete solution provides routes flexible planning [37]. No solution including restaurants into agenda was discovered, nor analogues selection in case of absence of venues, exactly matching the query (as stated above, the rating approach is not under consideration).

The system of interest shares with the above-mentioned sites the goal of providing a tourist trip planner. In this research, the recommender system is designed which is focused on users preferences consideration. For flexibility and thematic search of restaurants and sights, the recommender system under design includes the following subsystems, which will be discussed below.

1) Reviews analyzer with aspects extraction for sights and restaurants.
2) Knowledge base for venues (with a lightweight ontology-supported schema [15]).
3) Recommender system:
   a) content-based recommender strategies;
   b) flexible parameterization with user filters;
   c) lightweight-ontology-driven heuristics (apart from route-forming heuristics).
4) Itinerary building, conjugated with route planning and maps API.

Manuscript received on September 2, 2016, accepted on December 7, 2017, published on June 30, 2018.

Liliya Volkova is with the Moscow Institute for Electronics and Mathematics, National Research University Higher School of Economics, Moscow, 101000, Myasnitskaya ul., 20, Russia, and with the Bauman Moscow State Technical University, Moscow, 105005, 2-ya Baumanskaya ul., d. 5, str. 1, Russia (e-mail: liliya@bmstu.ru).

Elena Yagunova, Ekaterina Pronoza, Alexandra Maslennikova, and Danil Bliznuk are with the Saint Petersburg State University, Saint Petersburg, 199034, Universitetskaya nab., d. 7-9, Russia (e-mail: {iagounova.elena, katpronoza, msasha1996}@gmail.com, blizda@outlook.com).

Margarita Tokareva is with the Moscow Institute for Electronics and Mathematics, National Research University Higher School of Economics, Moscow, 101000, Myasnitskaya ul., 20, Russia (e-mail: rit1336@yandex.ru).

Ali Abdullaev is with the Bauman Moscow State Technical University, Moscow, 105005, 2-ya Baumanskaya ul., d. 5, str. 1, Russia (e-mail: klim_sychev@mail.ru).

ISSN 2395-8618

Aspects extraction techniques are developed for further venues automatic estimation, the detailed description is given in chapter II. The first two subsystems require a specific knowledge organization, which is a lightweight ontology [39], see chapter III. The latter two subsystems are described in chapter IV.

The recommender system based on user preferences implies a technique of evaluating venues aspects in terms of natural language. For ex., a sample user likes art, but not modern art, and his tastes are limited to authentic Italian cuisine. The easiest solution can be found when all of the venues of specified kinds are present in the vicinity. The question is what strategy should be built into the recommender system to search for similar venues in case of absence of the exact match. If there is no authentic Italian cuisine, some substitution should be mined with similarity heuristics (be it French cuisine or a café with pizza). In this research, ontological reasoning is considered to be the solution of this problem. Two sets of aspects are defined for sights and restaurants respectfully (see chapter 2), and the recommender system comprises rules deriving from lightweight ontology relations over the aspects mentioned.

II. Aspect-based Restaurant and Museum Information Extraction

A method for Russian reviews corpora analysis (as part of information extraction (IE)) is discussed, which gathers and structures restaurants and museums parameters from users' reviews, and feeds the recommendation system with the data collected. The focus of this chapter is on extracting aspects (so to be referred to).

IE methods, as well as NLP methods in general, are classified into rule-based, statistical and hybrid. The first approach implies using templates and semantic resources (e.g. WordNet-Affect, SentiWordNet, SenticNet), while statistical methods allow solving the task without such resources [27]. For recommender systems, in particular for museums and conterminal fields, three approaches are mostly combined: (1) content-based, (2) aspect-based, and (3) user-based [17], [21], [30], [32]. The only considered traits of the latter approach in this work are the review language and the informant's homeland. Content-based approach involves full consideration of official museums data from different resources; the aspect-based approach comprises analysis of aspects retrieved by automatic and semi-automatic reviews processing. The goal of the IE task in general is to retrieve most aspects extractable within the two approaches, while the focus of this work is on aspects extraction from reviews corpora, in particular on research for key aspects and analysis of their realization types.

The approach towards corpora analysis presented in this paper is based on non-contiguous bigrams and part of speech (POS) distribution analysis [28]. Trigger words dictionaries are obtained by means of the bootstrapping method. The venues can be described with a set of characteristics, for instance, service quality, food quality, cuisine type, price level, noise level, etc. The key aspects are selected below. All of the aspects to be extracted from the reviews are experts-predefined. No techniques of automatic aspects identification were employed, for these would inevitably introduce noise into the IE model. Most examples are dedicated to restaurants IE.

It should be stated that our corpora consist of Russian colloquial texts, and Russian is known for its rich morphology and free word order which complicate its automatic processing. Another complicating factor is that the practice of data adjusting to common recommender systems standards is not yet widespread in Russia, and therefore users' reviews are often not what one would expect them to be (e.g., free narratives are quite common, with no point of reviewing, as opposite to expected). However, according to the results, an information extraction system for Russian can still be successful, especially when based on the ideas obtained from corpora analysis.

A. Restaurant Information Extraction

The hypothesis is that the most important characteristics of a restaurant are service and food quality along with cuisine type, so the analysis is so far focused on these three (and on the extraction of their aspects). This assumption is proved by the distribution of the aspects in the data. These main aspects are discussed in this section, though more aspects can be aggregated within further research for fine-grained detail.

The next assumption is that the proposed IE system can be highly effective despite the difficulties imposed by the structure of a typical Russian restaurant review. The fact is that, when such a review is concerned, the key information about restaurant characteristics does not always lie on the surface. However, tuning models with respect to the results gained during corpus analysis can increase IE system performance.

The corpus analyzed consists of 32525 users' reviews (colloquial texts) about restaurants (4.2 millions of words). The reviews are provided by tulp.ru and dated 2013. A part of the corpus is annotated in a semi-supervised way (first, automatically using a simple keywords-based algorithm, and then manually corrected by two experts). It includes 1025 reviews about 206 restaurants located in the centre of Saint-Petersburg. The list of aspects is given in Table I (the most important aspects related to food quality, cuisine type and service quality frames, are given in bold).

TABLE I
Restaurant Aspects (Examples)

Restaurant Aspects			
Cuisine type	**Service Quality**	Company	Children menu
Food quality	**Staff politeness**	Audience	Kids area
Noise level	**Staff amiability**	Average cheque	Bar
Service speed	Cosiness	Price level	Parking place

ISSN 2395-8618

The task is actually a classification problem, but the classes differ from aspect to aspect. For example, for kids' area and bar aspects there are 2 classes: available and unavailable; and for the aspects related to food and service quality (service speed, food quality, etc.) we define 5 sentiment classes: -2, -1, 0, 1, 2. For each aspect the system should either label a review with one of the possible classes or reject it as irrelevant with respect to the given aspect. As most restaurants characteristics are never mentioned in the reviews, an empirical threshold frequency value of 10% is defined in this research, and aspects mentioned in at least 10% of reviews are considered. Classifiers were only trained for the frequent aspects (they are divided into groups in Table II).

TABLE II
FREQUENT RESTAURANT ASPECTS DISTRIBUTION IN THE CORPUS

Occurrence Percentage	List of Aspects
[85%; 100%]	Food quality (86%)
[55%; 85%)	Service quality (55%)
[25%; 55%)	Staff politeness and amiability, service speed, price level, cosiness
[10%; 25%]	Noise level, crampedness, romantic atmosphere, company

The information extraction task related to food and service quality can be reformulated as sentiment analysis with respect to the restaurant aspects of interest. For the aspects chosen as the most frequent ones, the following classifiers were considered: Naive Bayes (NB), Logistic Regression (LogReg), and Support Vector Machines (SVM) as implemented in scikit-learn [31]. In this paper an illustration of machine learning is given with respect to food and service quality criteria. Since the cuisine type aspect suggests a multilabeling task, in this section machine learning models are only considered with respect to food and service quality.

Since the annotated corpus includes a large amount of missing values, the classification task is divided into two parts: first, a classifier is trained to tell between missing and present values, and then, if the value is present, the classifier is to predict its class. The latter is discussed in detail in this section.

Our baseline feature set consists of unigrams and bigrams (on the lemma-level, only contiguous ones). Trigrams were also considered, but since they did not improve performance much while increasing feature space dimensions, trigrams were excluded from the feature set. The experiments were conducted with two extended features sets. First, only non-contiguous bigrams were added (with window size equal to 3 as it appeared to perform best). In the second set, emoticons and exclamations, predicative-attributive words and key words and expressions were added instead.

To evaluate the models, shuffle 10-fold cross-validation was conducted. Average weighted F1 scores for food and service quality are given in Table III. The weights are calculated as relative frequencies of the classes in the annotated subcorpus.

TABLE III
FOOD AND SERVICE QUALITY F1 SCORES
(BEST AVERAGE WEIGHTED F1 SCORE GIVEN IN BOLD)

Restaurant aspects	Model	Baseline, %	Extended (1), %	Extended (2), %
Food quality	NB	**69.45**	**70.08**	**70.26**
	LogReg	64.24	68.77	68.64
	SVM	63.99	65.57	66.21
Service quality	NB	**64.37**	**68.77**	**65.33**
	LogReg	56.14	65.05	57.90
	SVM	54.30	63.80	56.27

NB appears to be the best among the three classifiers for both aspects, but its basic and extended versions show similar scores while SVM and LogReg extended versions show improvement compared to corresponding basic versions.

In further phases of research other restaurant aspects were also considered (apart from food and service quality and cuisine type described in this paper), and experiments were conducted with different classifiers, such as Multinomial NB, Decision Trees, Random Forests, and Perceptron-based. Optimal combinations of feature and classifier were selected for each frequent aspect [26].

Basing on the experimental data, the suggestion is to recommend LogReg for the classification of informal unstructured Russian texts into those which contain information or opinion about the specific aspect and those which do not.

At sentiment classification task, NB is best for all the aspects. It can be explained by both the nature of the classifier and the data: NB, having high bias, usually behaves better on the small amount of training data, and for food and service quality aspects there are 5 classes of sentiment which makes the amount of training data inside each of the classes rather small. Therefore it might be suggested that NB is good at classifying sentiment in the informal texts on the small training set.

It should be also stated that including emoticons and exclamations into the feature set is not a good idea unless the aspect is service quality. For the other aspects it does not improve F1 or even impairs it [28].

For the service frame, dictionaries do improve the results. But food quality, one of the most important aspects, is best extracted using non-contiguous bigrams which cover a wide variety of the expressions of opinion. Thus, a more elaborate lexicon and dictionaries construction could be one of the promising work areas.

A thorough corpus analysis was conducted based on non-contiguous bigrams and POS-distribution of the trigger words context. Experiments with several classifiers showed that their performance can be improved with the results and ideas de-

ISSN 2395-8618

rived from corpus analysis, thus proving the importance of the latter. In particular, it has been shown that using trigger words and predicative-attributive words dictionaries is an effective approach for food quality extraction while service quality aspect, which is harder to deal with, demands a wider range of features.

B. Museums Information Extraction

As the recommender system at its origin is dedicated to cultural journeys, the museum topic requires corresponding aspects extraction as well. The implementation of an aspects extraction module necessitates reviews corpus analysis, patterns construction (including development of the methodology for such construction) and evaluation. The approach for patterns construction presented in this paper is based on n-grams (n ranges from 1 to 8) and POS-distribution analysis. Trigger words dictionary and predicative-attributive dictionaries are obtained by means of the bootstrapping method, targeted at the aspects of interest [27], [28].

The key distinctions for museum IE are the vast repertoire for aspects and the main focus on estimating trigger words and patterns coverage of users' reviews. This leads to combining information extraction, opinion mining and sentiment analysis procedures. The implemented approach is based on foresaid results for restaurant IE. But in this paper there is no results discussion for the evaluation stage is ongoing.

At this point the system is based on the following reviews corpus: The State Hermitage – 2 100 reviews, The Museo del Prado – 1 000, The Louvre Museum – 1 525, The Uffizy Gallery – 450, The Rijksmuseum – 425, The National Gallery – 350.

The approach being as for restaurants, the procedure of analysis comprises the following stages: (1) corpus preprocessing (tokenization, lemmatization, normalization, splitting into sentences, filling frequency and n-grams dictionaries), (2) filling nominations and predicative-attributive dictionaries, (3) filling keywords and keyphrases dictionaries, (4) filling modifiers dictionaries, (5) titles analysis for generalized description.

The predicative-attributive dictionaries were chosen, in particular for adjectives and full and short participles, which refer to nominations of the key frames. This is conditioned by the POS distribution analysis within corpus n-grams showed dominating of noun phrases in most aspects description [28].

Aspects are objective (for ex., tickets e-booking, student prices), subjective (for ex., queues for tickets, crowds inside museums), and mixed. The first category requires IE, the second – opinion mining and sentiment analysis, while the last category requires the composition of both approaches. The latter triade is solved in this research: subjective aspects are of interest, these are represented in 5-degree scale (ranged from -2 to 2), namely ticket prices, queues and crowds in museums.

The important point is to make sure that enough cases for aspects under consideration are present in the subcorpus dedicated to one aspect. The threshold is empirical and is 10 %; it would be also actual for the next stage of pre-processing based on machine learning with different classifiers. To compensate weak accessibility of semantic resources for Russian, semi-automatic dictionaries filling is used basing on the reviews corpus, while thorough syntactic analysis is substituted by n-grams analysis (n ranges from 1 to 8). The latter is supplemented by POS tags, in particular, by POS-filters applied to n-grams components [24]. Different types of negation for Russian might also be covered by the same n-grams.

The data on the above mentioned aspects is present for all of the museums named. Basic museums information is extracted, as well as masterpieces (by name and author) and different services. The worst results are obtained for mining exhibitions, even the long-term ones. Reviews in English have such advantage as their uniform structure compared to reviews in Russian. For the latter the problem is in their rather essay character, for example, these sometimes contain compulsive comparisons to the homeland museums (Hermitage, Tretiakov Gallery, Russian Museum, etc.). Reviews in several corpora are non-uniform and vague, as it is stated in [24]. Using semantic dictionaries and hierarchies thesauri [20], which were semi-automatically or manually filled from reviews corpus, allows improving the quality of most aspects extraction.

The repertoire for topics and aspects is vast: general information, masterpieces, exhibitions, service, tickets prices, e-booking for museums tickets, tickets queues, payment by credit card, opening time, etc. All of these aspects imply thorough IE techniques, and the repertoire allows different kinds of routes: from trips for students with low budget to wealthy tourists, from family tourism with children to big youngsters companies. Considering all of the aspects in the recommender system allows covering a wide range of tourist types, so that the system in production would gain success for its detailed search (with blocking or non-clocking aspects, e.g., no 18+ bars, or preferably parks and family leisure). The aspects extracted are provided with interrelations, which form the lightweight ontology, the latter serving not only as dictionary for aspects extraction, but also for estimating objects within venue categories for thematic itineraries recommending described further.

III. The Lightweight Ontology

In order to describe semantics lying behind data, ontologies can be used in an information integration task to make the content explicit [40]. Addressed to the bottleneck of combining domain experts with ontology engineers in order to build a full-sized ontology, a lightweight ontology is intended to meet the expectations of people who argue in favor of

ISSN 2395-8618

powerful, knowledge-intensive applications based on ontological reasoning [7]. It is presumed that lightweight ontologies are limited in their expressiveness and are mostly focused on a hierarchy of concepts [22], but still they have proven useful, this resonates with the so-called Hendler hypothesis [16]: "A little semantics goes a long way." Besides, the problem of unsupervised ontology learning is still unsolved [23] and is most crucial for languages which still do not have semantic resources thorough enough, e.g. for Russian (though several projects exist [20]).

On the base of analysis conducted, it should be stated that applying and developing a taxonomic model and further a lightweight ontology is a perspective approach towards determining venues similarity (through similarity relation [33]) and solving the problems derived from data insufficiency and incompleteness. A pictorial example of a lightweight ontology employment is as follows: if there is no direct "whisky bar" category match available around user's current location, the system should use rule-based analysis and advise alternatives, e.g. a restaurant with an excellent selection of whisky. The approach allows (1) searching within a database with further application of lightweight-ontology-driven rules of venues extraction in case of absence of match, and (2) pre-mining the data to provide more substitutes (with fuzzy estimation). Additionally, the information on a venue might be processed to match the description of venues satisfactory to the query, but not reachable, e.g., in a given time period [10].

Two domains are covered for further referring to their concepts and interrelations as in corresponding domain of knowledge with agreed meanings and properties [14]: restaurants and sightseeing. Besides, intersections of domain vocabularies can slightly disfigure the results [2]. "Good ontology design, especially for larger projects, does require a degree of modularity. An architecture of multiple ontologies often work together to isolate different work tasks so as to aid better ontology management. Ontology architecture and modularization is a separate topic in its own right" [3], [4].

Though several approaches exist towards automatic converting of classifications into lightweight ontologies [11], still initial expert estimation is of big value and is chosen as the path for this research. Three data sources were considered.

(1) The Foursquare [9] classification which is quite fulfilled but does not contain relations nor all of the parameters necessary (e.g., there is no strict cuisine types classification, and one can find a bakery and restaurants with different types of Chinese cuisine on the same level of abstracts). Such hierarchy requires thorough correctives.

(2) An experts-composed taxonomic model which comprises a thorough classification (designated for this research) and is on the relations adjustment stage.

(3) The set of aspects extracted for museums and restaurants for further lightweight ontology filling (extracted with the above discussed techniques).

The advantages of all of the three items are considered for creating a hybrid model. The characteristics from the latter set of aspects are necessary to complete the first two items, refined and modified. The diverse relations are necessary for various tasks requirements: vertical and horizontal, different types of them (for ex., the "differ" relation for classes might reflect music genre, target audience age, average bill). This allows a more detailed and flexible search oriented on refining the output according to user's query parameters. At this stage the lightweight ontology is as stated in Table IV. The OWL [25] is used, the lightweight ontology is under further development.

TABLE IV
CURRENT LIGHTWEIGHT ONTOLOGY VOLUME

Category	Restaurants domain	Sightseeing domain
Classes and subclasses	32	22
Instances	55	25
Object relations	12	6
Properties or type relations	30	10

IV. THE RECOMMENDER SYSTEM

Accumulating a relevant dataset itself being a research and engineer task (e.g., getting venues basic data from Foursquare), its processing requires thorough development of techniques for all recommender system components available to process a huge amount of information on the fly. Tourist agenda composer meets the requirement of providing real-time services. For routing, it is necessary to find solutions to the problem of the aggregated dataset processing interfaced with map APIs. Aforesaid resulted in several project decisions discussed in this chapter, in particular a recommender function taking submitted preferences into account to provide relevant content.

With the dataset collected by means of reviews analysis, the system should weigh the venue alternatives with user preferences to compose itineraries satisfying the restrictions imposed, and to advise the most optimal according to the recommender function. Recommender strategies could be implemented as follows.

Content-based systems deal with user tastes profiles based on one's ratings. Generally, when creating a profile, a survey is urgent for getting initial information in order to avoid the new-user problem [6].

Case-based systems implement a particular style of content-based ones, undermining the apparent inability of most systems to consider preferences varying over time. New problems are solved by retrieving a case whose specification is similar to the current target problem and then adapting its solution to fit the target [34], [5].

ISSN 2395-8618

Collaborative filtering attributes users to groups with similar preferences within: user-based approach or item-based approach [29].

Hybrid recommender approaches [1].

The rating approach is the one avoided in this research: ratings are widely spread over sites, but apparently not always exact they are, the task of verifying and attributing ratings requires a separate study. Moreover, user preferences differ, which is not reflected by ratings on the whole. Hence, the recommender system is venues-oriented.

A hybrid of the first two strategies is of interest with content-based filtering and implementing some predefined cases (e.g., the must-see sights for first-time visitors to the city). The cold start problem [35] is solved with current preferences indicated for each route query (blocking/non-blocking filtering), with few additional cases possible (extracted from check-ins frequency or manually by experts). With aggregating initial user behavior, detecting and further specifying this very user's modus operandi is subject for the collaborative filter extension of the strategy chosen, subject to design.

The recommender system developed generates a set of routes (itineraries); its inputs for content-based filtering are as follows: (1) an ordered set of venue categories of user's interest; (2) filters for each category (by aspect); (3) an overall time filter.

With venues represented as a graph, a combinatorial optimization problem is solved by means of ant colony optimization technique, which results in suboptimal solution finding (actually, a set of solutions) in a finite time and allows embedding local search. A heuristic is proposed for estimating found routes' costs. As the prototype developed solves a problem of finding the shortest path through categories of objects of interest (e.g., museum + museum + restaurant + museum) with filtering by categories' aspects and considering time restrictions (lower and/or upper bounds), a recommendatory function (RF) is an important part of it. RF penalizes a route for every filter-parameters transgressing by a value between (0, 1]. While some aspects might be absent for a specific object, their penalties are subject to customize in each query. A heuristic of path cost between two nodes is a sum of transfer time and time spent in the node, divided by all the penalties (by filter and by time restriction) multiplication. The overall route cost is a sum of such node-to-node costs, divided by all of the penalties multiplication; then the minimization problem is solved. Dijkstra algorithm is used for routes finding from current node to the next category objects. It is optimized algorithmically to increase performance. Additionally, reducing map nodes, which contain no objects, resulted in x50 acceleration of the well-optimized system.

Beyond the recommender engine, its user interface is subject to implement. The query interface should contain numerical and categorical restrictions for aspects and time, as well as thematic tags (instead of trackbars pattern implemented in [37]) necessary for thematic-focused itineraries forming task. Visual representation of an itinerary might use Gant diagrams in addition to map-based route with links to extracted venues description and/or sites, or otherwise follow the schedule pattern (fully designed in [37]). The engine implemented allows fast creating 10 itineraries per query, and it is easy to provide such feature as recalculating from the current location in case the tourist has changed plans with a time shift. Taking the location factor into consideration is promising for tourist recommender systems, in particular to obtain updated sightseeing information [30] for fast in-place replanning. Location might also be useful for developing an extra widget for creating situational hints (e.g., when a historical building is approached) [8], [19] according to one's tastes/query.

These foresaid project decisions allowed developing a real-time recommender system which forms itineraries with routes satisfying the imposed restrictions, arising from user queries. Optimizations resulted in reduction of this system's recall time, which shifts the system towards production and, in particular, makes it big data-ready, which is actual for big cities and, furthermore, regions (e.g. Provence).

Let the sample input data include 5 categories of venues, the time restriction given not less than 5 hours, starting at 10:00. In Table 5 two routes are provided for this query, each venue accompanied with a timestamp, approximate visit duration and options affected by the query. For this sample, two restaurant positions require simple parameters matching, while the third one and the cultural sites are provided with hashtags marking desired thematic. In case of absence of the 'Lights of Moscow' museum, the recommender system selects a substitute also dedicated to lighting: 'The Ray' cultural center (not a museum, the category is changed). This further implies changes in adjacent positions (the café and the restaurant) to fulfill the route.

TABLE V
SAMPLE QUERY AND ROUTES

Query	Route A	Route B
A café with lunch	10:38 Belucci café (20 min): brunch, dinner, lunch	10:17 Emelya café (15 min): dinner, lunch
A museum (#lighting)	11:01 'Lights of Moscow' museum (56 min)	11:15 'The Ray' cultural center (60 min)
A restaurant with dinner, full bar, wi-fi, outdoor seating, live music	12:38 'The Birch Chalet' restaurant (30 min): full bar, cocktails, live music, outdoor seating, serving lunch, brunch, dinner, with wi-fi	12:37 'Spices and Pleasures' restaurant (35 min): full bar, cocktails, live music, outdoor seating, serving lunch, brunch, dinner, with wi-fi
A cafeteria (#donuts)	13:38 (B: 13:40) The Mega Foods canteen (45 min): breakfasts, desserts, dinner and lunch	
A historic site (#photography)	15:00 (B: 15:02) The Young Photographer Memorial (20 min)	

ISSN 2395-8618

V. Conclusion

The lightweight ontology is described, which covers the domains of restaurants and museums. With this basis the aspects mining method is discussed in detail. Annotated venues are source for automated routes planning and recommending methods, with lightweight ontology refining given heuristics of user's interest in objects. Heuristics for the recommender system and those for estimating objects' relevance to the query consist of rules deriving from lightweight ontology relations over the aspects mentioned. A recommender system for thematic itineraries is designed, which is big data-ready and optimized to allow real-time advising.

Venue aspects aggregated are processed along with user preferences by the flexible route recommending system which generates thematic itineraries throughout the city. A number of such itineraries are generated to user selection, each consisting of particular venues selected by means of recommender techniques, and accompanied with a schedule and a map-based route.

The further development will include constructing different types of relations in order to allow detailed venues analysis. For instance, semantic matching methods should be useful [12], [13] for the "match" relation in case of implementing different overlapping hierarchies, intended for processing detailed information while selecting venues to fulfill the itinerary suiting the request.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, pp. 734–749, June 2005. Washington: IEEE Computer Society.

[2] M. Alexandrov, A. Gelbukh and P. Rosso, "An Approach to Clustering Abstracts", *LNCS* 3513, pp. 275–285, 2005. Berlin: Springer.

[3] M. K. Bergman. (2009, November 23). A reference guide to ontology best practices. In: AI3: Adaptive Information [Online]. Available: http://www.mkbergman.com

[4] M. K. Bergman. (2010, September 13). A new methodology for building lightweight, domain ontologies. In: AI3: Adaptive Information [Online]. Available: http://www.mkbergman.com

[5] D. Bridge, M. Goker, L. McGinty and B. Smyth, "Case–based recommender systems," *Knowledge Engineering Review*, vol. 20 (3), pp. 315–320, 2006. New York: Cambridge University Press.

[6] L. Candillier, K. Jack, F. Fessant and F. Meyer, "State-of-the-art recommender systems," *Collaborative and Social Information Retrieval and Access-Techniques for Improved User Modeling*, pp. 1–22, 2009. Hershey: IGI Global.

[7] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Berlin: Springer (2006)

[8] A. A. Economides, "Requirements of mobile learning applications," *Int. J. of Innovation and Learning*, vol. 5 (5), pp. 457–479, 2009. Geneva: Inderscience Publishers.

[9] Foursquare [Online]. Available: http://www.foursquare.com

[10] A. Gelbukh, G. Sidorov and A. Guzmán-Arenas, "A method of describing document contents through topic selection," in *Proc. of the String Processing and Information Retrieval Symposium and International Workshop on Groupware*, pp. 73–80, 1999. Los Alamitos: IEEE.

[11] F. Giunchiglia, M. Marchese and I. Zaihrayeu, "Encoding classifications into lightweight ontologies", University of Trento, Italy, Technical Report DIT-06-016, March 2006.

[12] F. Giunchiglia and P. Shvaiko, "Semantic Matching," *The Knowledge Engineering Review Journal*, vol. 18 (3), pp. 265–280, 2004. New York: Cambridge University Press.

[13] F. Giunchiglia, P. Shvaiko and M. Yatskevich, "S-match: An algorithm and an implementation of semantic matching," in *Proceedings of ESWS'04, LNCS* 3053, pp. 61–75, 2004. Heidelberg: Springer-Verlag.

[14] M. Gruninger and J. Lee, "Ontology Applications and Design," *Communications of the ACM*, vol. 45 (2), pp. 39–41, 2002. New York: ACM.

[15] N. Guarino, "Formal Ontology and Information Systems," in *Proceedings of Formal Ontologies in Information Systems*, pp. 3–15, 1998. Amsterdam: IOS Press.

[16] J. Hendler, "On beyond ontology", Keynote talk, Second International Semantic Web Conference, Sanibel Island, Florida, USA, 2003, unpublished.

[17] Y.-M. Huang, C.-H. Liu, C.-Y. Lee and Y.-M. Huang, "Designing a Personalized Guide Recommendation System to Mitigate Information Overload in Museum Learning," *Journal of Educational Technology & Society*, vol. 15 (4), pp. 150–166, 2012. International Forum of Educational Technology & Society.

[18] Iknow.travel [Online]. Available: http://www.iknow.travel

[19] I. Keller and E. Viennet, "Recommender Systems for Museums: Evaluation on a Real Dataset," in *IMMM 2015: The Fifth International Conference on Advances in Information Mining and Management*, pp. 65–71, 2015. IARIA.

[20] Y. Kiselev, A. Krizhanovsky, P. Braslavski, I. Menshikov, M. Mukhin and N. Krizhanovskaya, "Russian Lexicographic Landscape: a Tale of 12 Dictionaries," in *Computational Linguistics and Intellectual Technologies: papers from the Annual International Conference "Dialogue" (Moscow, 27-30 May 2015)*. Issue 14, vol. 1, pp. 254–271, 2015. Moscow: RSUH.

[21] T. Kuflik, E. Minkov and K. Kahanov, "Graph-based Recommendation in the Museum," in *CEUR Workshop Proceedings*, vol. 1278. Proceedings of the First International Workshop on Decision Making and Recommender Systems (DMRS2014, Bolzano, Italy, September 18–19, 2014), pp. 46–48, 2014.

[22] D. Lande, A. Snarskii, E. Yagunova, E. Pronoza and S. Volskaya, "Network of Natural Terms Hierarchy as a Lightweight Ontology," in *Thirteenth Mexican International Conference on Artificial Intelligence MICAI 2014*, Tuxtla Gutiérrez, Mexico, 16–22 November 2014. Special session. Revised papers. Gelbukh, A., Espinoza, F. C., Galicia-Haro, S. N. (Eds.), pp. 16–23, 2014. Los Alamitos: IEEE.

[23] N. V. Lukashevich, B. V. Dobrov and D. S. Chuyko, "Selecting word phrases for an automatic text processing system dictionary" (in Russian), in *Computational linguistics and intellectual technologies: Proceedings of Int. Conf. «Dialog–2008»*, pp. 339–344, 2008. Moscow: RSUH.

[24] A. Maslennikova and E. Yagunova, "Information extraction and opinion mining for reviews on the most prominent museums in Russian and English. Methodic and preliminary results," in *New information technologies in automated systems: proceedings of 19th scientific and practical seminar* (in Russian), pp. 68–74, 2016. Moscow: V. M. Keldysh Institute for Applied Mathematics Press.

[25] *OWL Web Ontology Language Guide, W3C Recommendation*, M. K. Smith, C. Welty and D. L. McGuinness (Eds.), 10 February 2004 [Online]. Available: http://www.w3.org/TR/2004/REC-owl-guide-20040210/

[26] E. Pronoza, S. Volskaya and E. Yagunova, "Corpus-based Information Extraction and Opinion Mining for the Restaurant Recommendation System," in *Proceedings of the 2nd Statistical Language and Speech Processing*. L. Besacier et al. (Eds.): SLSP 2014, LNAI, vol. 8791, pp. 272–284, 2014. Berlin: Springer.

ISSN 2395-8618

[27] E. Pronoza, E. Yagunova and A. Lyashin, "Restaurant Information Extraction for the Recommendation System," in *Proceedings of the 6th Language Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, 2nd Workshop on Social and Algorithmic Issues in Business Support: "Knowledge Hidden in Text"*, 2013. Berlin: Springer.

[28] E. Pronoza, E. Yagunova, S. Volskaya and A. Lyashin, "Restaurant Information Extraction (Including Opinion Mining Elements) for the Recommendation System," in *13th Mexican International Conference on Artificial Intelligence, MICAI2014*, Tuxtla Gutiérrez, Mexico, November 16–22, 2014. Gelbukh, A., Espinoza, F. C., Galicia-Haro, S. N. (Eds.). Proc., part I, pp. 201–220, 2014. New York: Springer.

[29] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40 (3), pp. 56–58, March 1997. New York: ACM.

[30] M. K. Sarkaleh, M. Mahdavi and M. Baniardalan, "Designing a tourism recommender system based on location, mobile device and user features in museum," *Int. J. of Managing Information Technology*, vol. 4 (2), pp. 12–21, 2012. Geneva: Inderscience Publishers.

[31] SciKit machine learning library for Python [Online]. Available: http://scikit-learn.org

[32] L. Sebastia, I. Garcia, E. Onaindia and C. Guzman, "E-Tourism: a tourist recommendation and planning application," *International Journal on Artificial Intelligence Tools*, vol. 18 (05), pp. 717–738, 2009. Singapore: World Scientific Publishing Co. Pte. Ltd.

[33] G. Sidorov, A. Gelbukh, H. Gómez-Adorno and D. Pinto, "Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model". *Computación y Sistemas*, vol. 18 (3), pp. 491–504, 2014.

[34] B. Smyth, "Case-based recommender," *The adaptive web*, pp. 342–376, 2007. Berlin, Heidelberg: Springer-Verlag

[35] M. M. Tokareva, L. L. Volkova and A. P. o. Abdullaev, "On a recommender system for itineraries based on user preferences evaluation," in *New information technologies in automated systems: proceedings of 19th scientific and practical seminar* (in Russian), pp. 75–80, 2016. Moscow: V. M. Keldysh Institute for Applied Mathematics Press.

[36] Travel2Moscow [Online]. Available: http://www.travel2moscow.com

[37] Triplantica [Online]. Available: http://www.triplantica.com

[38] Triptomatic [Online]. Available:, http://www.triptomatic.com

[39] M. Uschold and M. Gruninger, "Ontologies and semantics for seamless connectivity," in *SIGMOD Rec.*, 33(4), pp. 58–64, 2004. New York: ACM.

[40] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, "Ontology-Based Integration of Information – A Survey of Existing Approaches," in *Gómez Pérez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.). Proc. of IJCAI-01 Workshop: Ontologies and Information Sharing*, Seattle, WA, pp. 108–117, 2001.

ISSN 2395-8618

Building an Information Extraction and Question Answering Model for Text Based on the Human Brain Process

F. A. K. Hemant

***Abstract*—An information extraction and question answering model for text, which is based loosely on the human brain process, is showcased in this paper. The ideology used is based on how humans perceive and interact with text, and the process of storing the text for future reference. Each word of each sentence is cross referenced and linked with all available information and the answer is given based on matching information found. The model is basic, but the future applications and scope of improvement is also shown.**

***Index Terms*—Question Answering, Linguistics, Information Extraction, Text Analysis**

I. Introduction

The majority of data and information in the world is transmitted and stored in the form of text. It could even be said that language and text are the backbones of the human civilization. Without communication of knowledge and ideas, humans could not have progressed to the state they are in now.[5][6]

The first use of language was to communicate, whether it be ideas or information. With this motivation in mind, the goal was to build a simplistic computer model for analyzing and storing information in text, for easy retrieval [7].

This document is divided into four sections. The first section addresses the ideology used to build the model. The second section displays the model built so far. The third section shows the results achieved. The last section talks about the future scope and applications of this project.

II. Ideology

A very simple adaptation of the human brain's process of perceiving information is used. Consider a simple sentence:

"Bob went to Jim's house last weekend."

The first thing that is addressed is the identity of the entity "Bob". The brain first goes about remembering information regarding the entity, and the latest known instances when the entity was referenced. This is followed by the same process regarding the second entity, "Jim".

The information regarding the other entities, i.e "house" and the time period "last weekend" are also recalled. Then, the information is stored, and this information is added to the list of instances recalled.

The process is only one way that a human brain might perceive information, and the existence of other processes is disregarded for now. This process is used because it functions at the most basic level, and is thus easier to implement, while also following the norms established i.e. to emulate the brain in at least the most rudimentary way [6].

III. Implementation

A system is built which tries to emulate the process shown in the last section. Each aspect is explained in detail in the following subsections.

A. Data Used and Depiction of Process

The flowcharts Fig. 1 and Fig. 2 depict the salient features of the process. The data sources are listed below:

1. Pang & Lee Data Set: A collection of movie reviews [1]
2. Newspaper Data: Local newspaper data was collected and used. About 400 English news articles were collected.
3. Wikipedia Data: Around 200 Wikipedia pages were used.

B. Text Parsing Module

Stop words in the input text are removed and the remaining words are considered to be the entities in the text.

Input text is parsed using the Stanford Parser. The dependency parsing module of the Stanford parser is most useful and efficient in this present endeavor, as we can obtain the relations between entities in data, which is most useful for storing data for easy retrieval. For example, for the sample sentence used in the last section, "Bob went to Jim's house last weekend.", the dependencies output is:

Manuscript received on November 18, 2016, accepted on October 12, 2017, published on June 30, 2018.

F. A. K. Hemant is with the International Institute of Information Technology, India (+919494868838; e-mail: kancharla.hemant@gmail.com).

ISSN 2395-8618

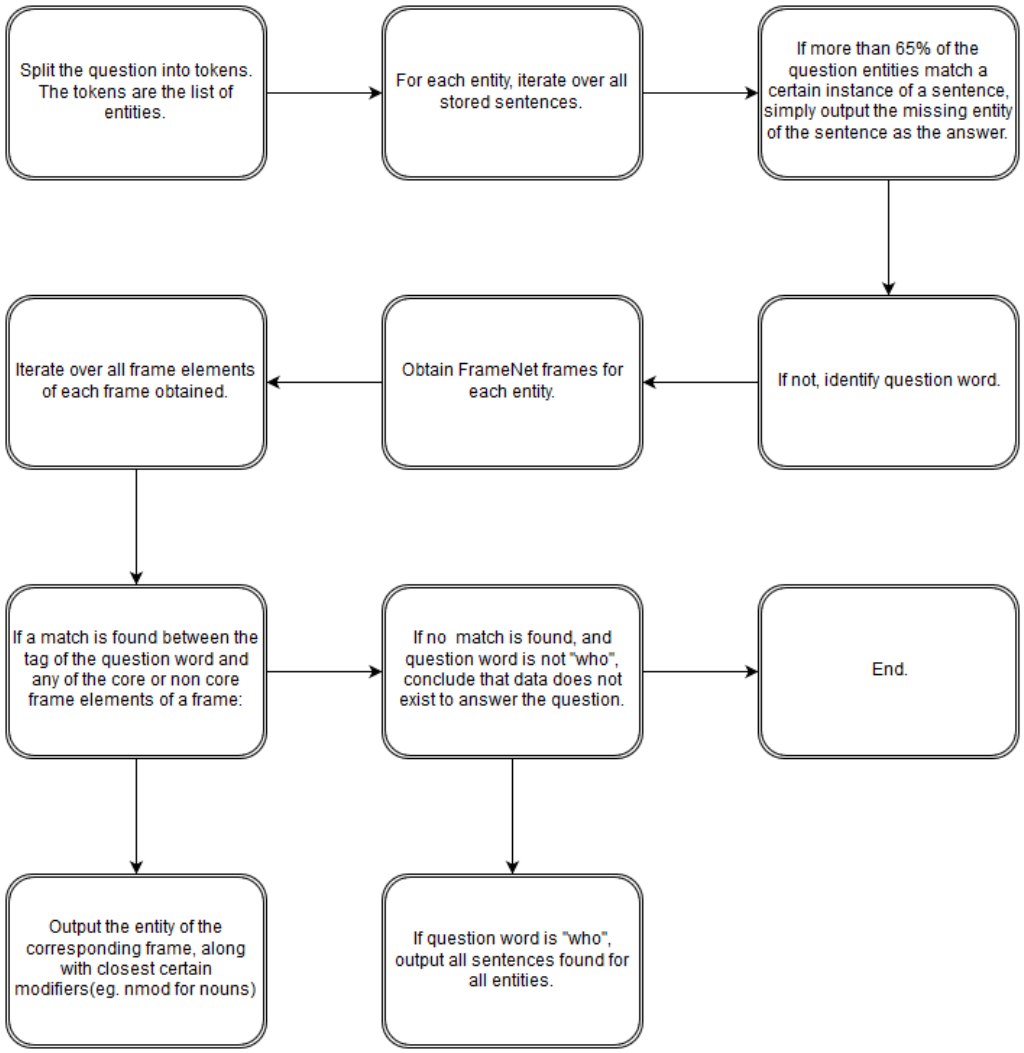


Figure 1. General flowchart of the process.

nsubj(went-2, Bob-1)
root(ROOT-0, went-2)
case(house-6, to-3)
nmod:poss(house-6, Jim-4)
case(Jim-4, 's-5)
nmod(went-2, house-6)
amod(weekend-8, last-7)
dobj(went-2, weekend-8)

Thus, we have a set of the dependencies between the words.

C. Text Storage Module

Text is stored in the form of a dictionary of lists in python. All the dependency tags associated are also stored. For the entity Bob, the resulting information stored would be:

Bob : {1{nsubj(went-2, Bob-1),
root(ROOT-0, went-2),
case(house-6, to-3),
nmod:poss(house-6, Jim-4),
case(Jim-4, 's-5),
nmod(went-2, house-6),
amod(weekend-8, last-7),
dobj(went-2, weekend-8)}}

The same is done for all the entities in the sentence, including "house". For decreasing space taken, the string of tags is stored once, and subsequently referenced in each entity.

This is done because as each entity is referenced, for effective question answering, it is relevant to have the information of all occasions the reference was instanced [8][9].

D. Question Answering Module

Question answering is the main test through which the system can be assessed and tested. A sample question:

"To whose house did Bob go to last weekend?"

ISSN 2395-8618

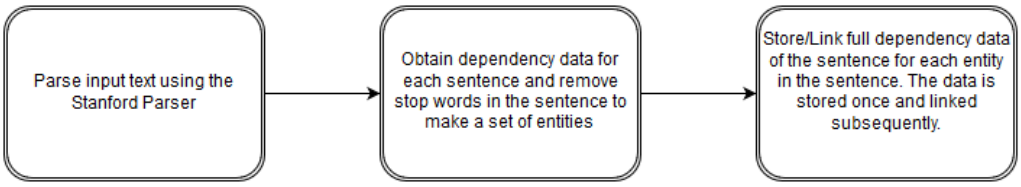


Figure 2. Flowchart of one step of the process.

The steps in which simple questions are answered are:

1. Identify entities (in this case, tokens) using a tokenizer i.e. house, Bob, last weekend
2. Iterate over all entries of each entity of the question sentence in the database.
3. Output the instance entry with the maximum matching (at least above 65 percent, this figure established by manual testing)
4. Output the missing entity in the instance as the answer

This is a simple and naive approach for basic questions. In the case that this fails, the case is either that the question is complex, or that no data exists to answer the question. In the case of a complex question like:

Where is Bob?

In this case, the matching approach wouldn't work. Thus, a different approach is used:

1. All question words are hard coded to tags, for instance, "where" to the "spatial location" tag and "when" to the "temporal collocation" tag (except the "who" tag).
2. FrameNet is used. The FrameNet project is building a lexical database of English that is both human- and machine- readable, based on annotating examples of how words are used in actual texts. It is a dictionary of more than 10,000 word senses, most of them with annotated examples that show the meaning and usage.

Frame elements are frame-specific defined semantic roles that are the basic units of a frame.

In the case of "where", which has a spatial tag, all entities in instances of the entity in the question are searched in framenet. If a "spatial" tag is located in the core or non-core frame elements of the frame of the entity searched, then the entity, and its closest modifiers (from the stored Stanford dependency tags) are outputted.

For example, "Where" has the "spatial" tag. Each word in the sentence is searched in framenet. For the entity "house", which is in the frame "buildings", there is a "spatial" tag in the non-core frame element. Thus, the entity, and its closest modifier, which is "Jim's"(only certain tags are considered, like nmod) is chosen as the answer.

The tags for each question word are:

- When: Temporal_Collocation
- What: Entity
- Why: Reason
- Which: Entity
- Where: Spatial_Co-location
- How: Means

The approaches combined give nominal results for all question words except "who". For "who", all the instances themselves are outputted. In the following section, the types of questions used and observations seen are displayed [2][3][4].

E. Results & Observations

This approach was taken after first manually checking the viability of using such an approach. About 100 sentences from newspaper data were taken and checked manually. As an accuracy of more than 60 percent was obtained, the work was continued. Accuracy in this case is simply meant to be whether any frame element matched with the tag of the question word. The tags of the question word were also decided upon after tweaking with other alternatives. The best results were obtained when using these tags, which are the tags that the question words have themselves in framenet.

The results obtained for each data set are displayed in Table 1.

TABLE 1.
Obtained results.

Dataset	Accuracy
Pang & Lee	67%
Newspaper	69%
Wikipedia	53%

Accuracy was checked manually for 200 sentences from each data set. Questions were 50% simple questions, and the rest complex.

It is to be noted that in the case of newspaper data, the highest accuracy was achieved. This can be attributed to the style and general format of sentences in the data. As the majority of the sentences are used to state facts, it is easier to answer questions. In the cases where answers spanned a phrase, the first approach gives answers accurately. It was

ISSN 2395-8618

also noted that most of the questions of such a variety did fall under the category of the first approach. This could be because in the cases where specific answers are needed, the questions also need to be specific. E.g., for the specific date of a certain event, some other information like the location of the event must also be given the question. And the presence of such information enables the first approach to work.

The lowest accuracy was in the case of the Wikipedia dataset. This is because of the high presence of data which might not be directly related to the topic of the text itself, at least in a way that can be identified by the present system. Answers which had too many sentences were also regarded as false, as such the accuracy is lower.

The performance was uniform on the Pang & Lee data sets, as the data itself was fairly uniform. Not many abnormalities were noticed, but the abundance of data for the "who" question was noted, as a very naive approach was used in that case.

IV. Conclusion

The future development of this model lies in using also the context of the text to answer questions regarding a sentence. Question answering has a wide variety of uses. Although for now the model cannot contend either in the scope of accuracy or complexity, it can compete in the area of varying domains. Because of the general implementation, which is not restricted to a certain domain, the model can be used in any domain for reasonable results.

A time-based model can also be built based on this model, which can record the states in the text. For example, if there are two sentences-"Bob went to Jim's house last weekend", and, "Bob is in Harry's house now", and the question regarding the location of Bob is asked, the system, which keeps a track of the temporal implications of sentences, should be able to give the correct answer, i.e. Harry's house. This is only one state, and other states can also be recorded and used to answer questions. This would increase accuracy by quite a fair bit.

Another improvement would be adding the capability of taking social media and chat data as the input. This would further increase the number of areas in which the model can be used.

Overall, this was only the first step in building a model which can even be referred to as being based on the human brain process. Future work will revolve around first increasing the viability and accuracy of the system, before again focusing on replicating the human brain process, to any extent.

Acknowledgment

I would like to thank Dr. Radhika Mamidi of the International Institute of Information Technology, Hyderabad for her steadfast guidance and belief in my idea.

References

[1] Bo Pang, Lilian Lee: “Opinion Mining and Sentiment Analysis”- Foundations and Trends in Information Retrieval archive, Volume 2 Issue 1-2 (January 2008), Pages 1-135

[2] Deepak Ravichandran, Eduard Hovy: “Learning surface text patterns for a Question Answering system”- ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Pages 41-47

[3] David L. Waltz: “An English language question answering system for a large relational database” – Magazine Communications of the ACM, Volume 21 Issue 7(July 1978), Pages 526-539

[4] Dan Moldovan, Sanda Harabagiu , Marius Pasca , Rada Mihalcea , Roxana Girju , Richard Goodrum, Vasile Rus: “The structure and performance of an open-domain question answering system”- Proceeding ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics,Pages 563-570

[5] Hecht-Nielsen, R.: “Neurocomputing: Picking the human brain”- IEEE Spectrum (United States) Volume: 25:3(1988-03-01)

[6] John C. Mazziotta, Arthur W. Toga, Alan Evans, Peter Fox, Jack Lancaster: “A Probabilistic Atlas of the Human Brain: Theory and Rationale for Its Development”- The International Consortium for Brain Mapping (ICBM), Volume 2, Issue 2, Part A (June 1995), Pages 89-101

[7] Steven Pinker: “How the Mind Works”- Annals of the New York Academy of Sciences, Volume 882, Great Issues For Medicine In The Twenty-First Century: Ethical And Social Issues Arising Out Of Advances In The Biomedical Sciences (June 1999), Pages 119–127

[8] Stephen Soderland: “Learning Information Extraction Rules for Semi-Structured and Free Text”- Soderland, S. Machine Learning (1999) 34: 233

[9] Ellen Riloff , Wendy Lehnert: “Information extraction as a basis for high-precision text classification”- ACM Transactions on Information Systems (TOIS), Volume 12 Issue 3(July 1994), Pages 296-333

ISSN 2395-8618

Journal Information and Instructions for Authors

I. Journal Information

Polibits is a half-yearly open-access research journal published since 1989 by the *Centro de Innovación y Desarrollo Tecnológico en Cómputo* (CIDETEC: Center of Innovation and Technological Development in Computing) of the *Instituto Politécnico Nacional* (IPN: National Polytechnic Institute), Mexico City, Mexico.

The journal has double-blind review procedure. It publishes papers in English and Spanish (with abstract in English). Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research. The main topics of interest include, but are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary Algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces
- Multimedia and Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Robotics
- Geo-processing
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Virtual Instrumentation
- Computer Architectures

B. Indexing

The journal is listed in the list of excellence of the CONACYT (Mexican Ministry of Science) and indexed in the following international indices: Web of Science (via SciELO citation index), LatIndex, SciELO, Redalyc, Periódica, e-revistas, and Cabell's Directories.

There are currently only two Mexican computer science journals recognized by the CONACYT in its list of excellence, *Polibits* being one of them.

II. Instructions for Authors

A. Submission

Papers ready for peer review are received through the Web submission system on www.easychair.org/conferences/?conf=polibits1; see also updated information on the web page of the journal, www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish. In case of Spanish, author names, abstract, and keywords must be provided in both Spanish and English; in recent issues of the journal you can find examples of how they are formatted.

The papers should be structures in a way traditional for scientific paper. Only full papers are reviewed; abstracts are not considered as submissions. The review procedure is double-blind. Therefore, papers should be submitted without names and affiliations of the authors and without any other data that reveal the authors' identity.

For review, a PDF file is to be submitted. In case of acceptance, the authors will need to upload the source code of the paper, either Microsoft Word or LaTeX with all supplementary files necessary for compilation. Upon acceptance notification, the authors receive further instructions on uploading the camera-ready source files.

Papers can be submitted at any moment; if accepted, the paper will be scheduled for inclusion in one of forthcoming issues, according to availability and the size of backlog.

See more detailed information at the website of the journal.

B. Format

The papers should be submitted in the format of the IEEE Transactions 8x11 2-column format, see http://www.ieee.org/publications_standards/publications/authors/author_templates.html. (while the journal uses this format for submissions, it is in no way affiliated with, or endorsed by, IEEE). The actual publication format differs from the one mentioned above; the papers will be adjusted by the editorial team.

There is no specific page limit: we welcome both short and long papers, provided that the quality and novelty of the paper adequately justifies its length. Usually the papers are between 10 and 20 pages; much shorter papers often do not offer sufficient detail to justify publication.

The editors keep the right to copyedit or modify the format and style of the final version of the paper if necessary.

See more detailed information at the website of the journal.