Retrieving Targeted Results from a Static File Repository using a Keyword matching Mechanism relying on a Cluster-based Algorithm

Mădălina ZURINI, Diana BUTUCEA
University of Economic Studies, Bucharest, Romania
madalina.zurini@gmail.com, dianabutucea@gmail.com

Due to the fact that the web based solutions designed for learning contain, among the compulsory functions, the uploading of didactic materials of the person doing the examining (on a regular basis this being the professor) and the possibility of accessing these by the examinee (on a regular basis this being the student), within this paper we have chosen to set the goal of finding a resolution that will enable the access of content relevant to the person being examined. Hereby we have suggested a prototype which will capacitate the singling out and grouping of documents depending on the keywords, which will be followed by a visual search depending on the distance between two documents, by the recurrence of the closest k documents to the one being the element of interest – reaching the optimal alternative in case of a performance oriented point of view. The algorithm needed for the extraction of this data is presented within the paper. An optimization model is proposed in order to reduce the time consuming component in regards to the minimization of differences in the quality of the documents resulted in the automatic search using a k Nearest Neighbour grid search engine.

Keywords: *e-learning, grid-based algorithm, keyword, kNN, automatic grid search optimization*

1 Introduction

The places and faces of information have changed in dramatic ways since the birth of the Web. Over the past ten years, library-based research has been replaced by web-based research, just as libraries have shifted from book and journal repositories to learning commons and large-scale computer labs. Also, the rapid growth of information on the Web has clearly had major implications for the ways students identify the need for, locate, evaluate, use, and create information. The evolution of the Web, particularly the maturation of search engines and development of Web 2.0 technologies, has forever changed the information landscape. [1]

Adopting advanced forms of information dissemination and information technology solutions has been the main driving force of vigorous development of e-learning. Nowadays, the popularization and application of Internet provides a broad implementation space to distance education. From the point of view of development process of information technology, each step of Internet will bring new impact and positive effect to distance education model. [2]

Introducing these as a starting point, we have developed an algorithm for automatic generation of documents' keywods and a framework for determining the first k documents similar in terms of keywords to a given input document using and optimized searching algorithm based on k Nearest Neighbour algorithm with an integration of space reduction.

In chapter 2 there is presented the e-learning application and the general activities flow. There is also described an automated algorithm for choosing keywords in each uploaded document.

Chapter 3 contains the description of k Nearest Neighbour (KNN) general algorithm, the input data, the output form and the main steps. All these information is gathered in a pseudocod that reveals the complexity level of the automatic search.

For the optimization of this algorithm, in chapter 4 a grid based model is proposed in order to reduce the complexity, generating a lower time consuming component by reducing the searching space recursively within the grid cells neighbour to the cell from which the input document is part of.

The empirical results are shown in Chapter 5, where a set of 900 bi-dimensional points are used for comparing the results obtained by kNN general algorithm and Grid based kNN proposed model. The comparison is done in terms of time consuming generated by the number of documents analysed using a metric of evaluation defined.

The conclusions are drawn in Chapter 6, where the need of integration the proposed model is highlighted in the context of e-learning platforms.

2. E-learning application

The learning application has been developed with the help of framework Laravel 4, based on PHP 5.3.7. language. We took into consideration the assurance of a high quality platform, this being one of the fastest working PHP full-featured framework. Using the head off offered advantages – expandable for a large number of users, ensures the stateful inspections and server cache for the Web – we have managed to integrate a product which presents a high level of security and the acceleration of network traffic.

The documents will be uploaded on to the platform by the professor with the especially elaborated module. He can also upload the 5 representative keywords for the document in question.

The general activities flow is shown in Figure 1.

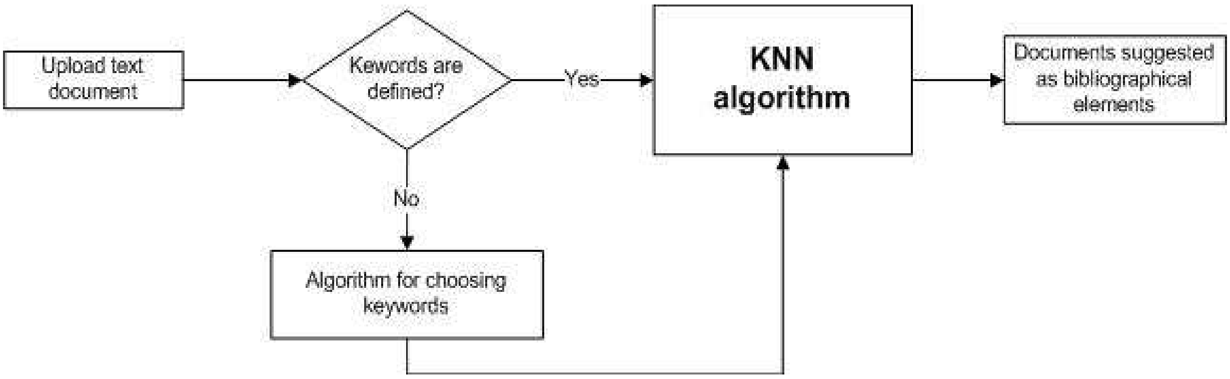

Upload text document
Kewords are defined?
Yes
No
KNN algorithm
Documents suggested as bibliographical elements
Algorithm for choosing keywords

Fig. 1. General Activities Flow

Since the application is used in an university, containing information for all the disciplines, the number of uploaded documents will easily reach a few thousands.

In case the professor chooses not to set the keywords, we have introduced a possible algorithm that will define them in an authomathic manner. This is presented in Figure 2. All the words in the documents are counted and depending on the length and frequency found within the text, the 5 mentioned keywords will be witheld to be used further on, in this paper in the way it has been mentioned. In order to establish a good classification, we took into consideration the length of the text and we set a different minimum number of appearances in each case. In case a word appears in the title, the system places it in a superior position, since it is considered of higher importance. Also, in case of an equal number of appearances, alphabetical classification is applied.

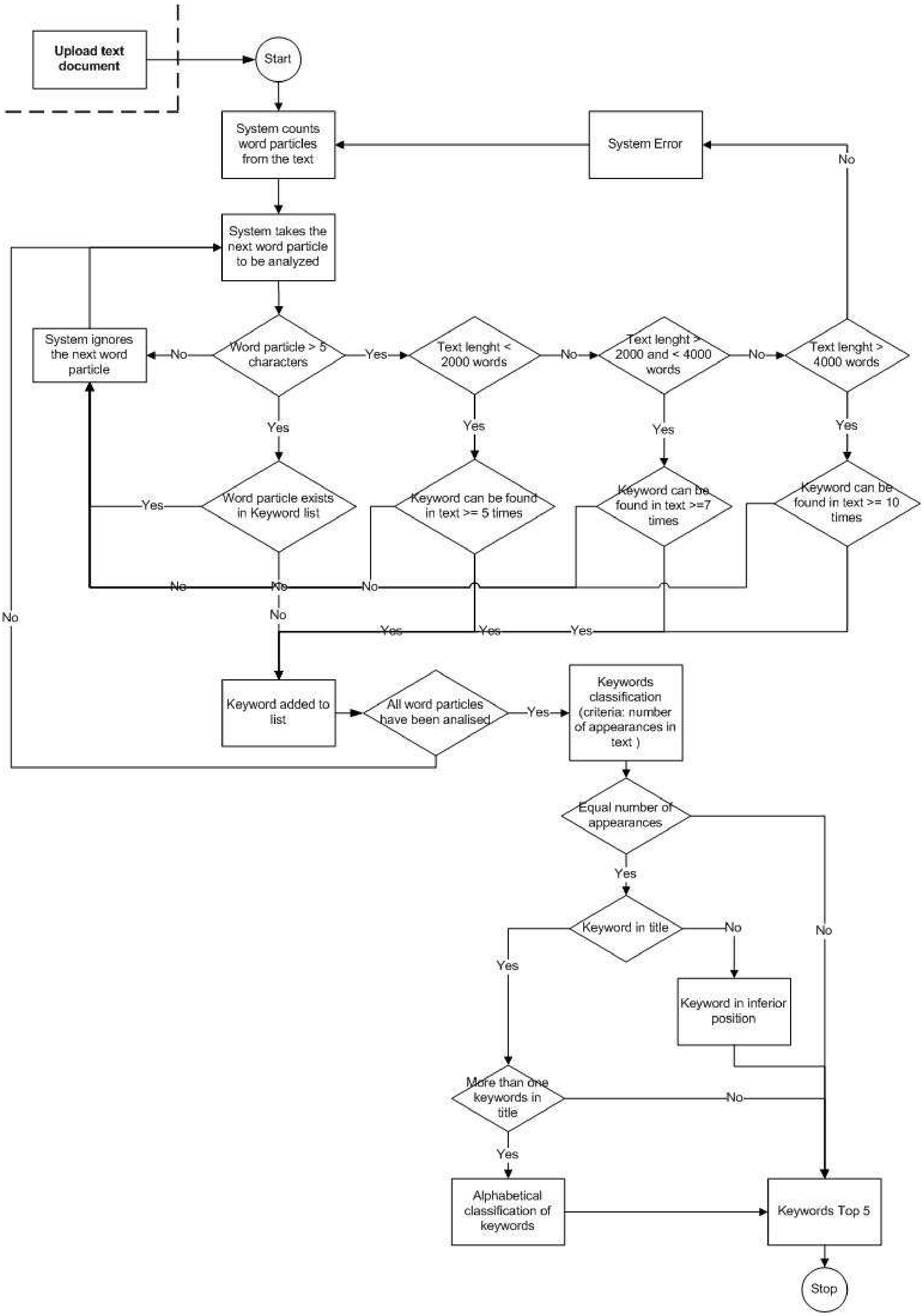

Upload text document
Start
System counts word particles from the text
System Error
No
System takes the next word particle to be analyzed
System ignores the next word particle
No
Word particle > 5 characters
Yes
Text lenght < 2000 words
No
Text lenght > 2000 and < 4000 words
No
Text lenght > 4000 words
Yes
Yes
Yes
Yes
Yes
Word particle exists in Keyword list
Keyword can be found in text >= 5 times
Keyword can be found in text >=7 times
Keyword can be found in text >= 10 times
No
No
No
No
No
Yes
Yes
Yes
Keyword added to list
All word particles have been analised
Yes
Keywords classification (criteria: number of appearances in text)
Equal number of appearances
Yes
No
Keyword in title
Yes
No
Keyword in inferior position
More than one keywords in title
No
Yes
Alphabetical classification of keywords
Keywords Top 5
Stop

Fig. 2. Algorithm for choosing keywords

3. KNN automatic search

K Nearest Neighbour algorithm, also known as Closest Pair of Points Algorithm in computational geometry is the algorithm used for searching the closest k neighbours of an n-dimensional given point, neighbours that are part of an initial set of n-dimensional a priori known objects. This algorithm is used in the process of supervised classification and for automatic search.

In the context of supervised classification, as a knowledge base a set of n-dimensional points a priori classified is used. In addition, a new object is used, also represented in the n-dimensional space, object that is used as input data in the classification process.

K Nearest Neighbour algorithm is run with the input parameter k, the number of neighbours used in the analyses, the set of objects, X, and the new object unassigned to a class. Using a distance function, the algorithm returns the closest k objects in the context of distance function minimization from each object to the input object. According to the defined objective, the returned objects are then aggregated using an aggregation function based on vote majority with constant or variable weights.

In the process of automatic search, the k returned objects are not aggregated, but used as output values. Also, the initial set of objects isn't a priori classified into membership classes.

In the context of metric space, the notation used for the initial set of n-dimensional objects is X:

$$X = \{x_1, x_2, \dots, x_m\}$$

where:

- *m* represents the cardinality of the set of objects X.

Each object is represented by the values of the analyzed characteristics that are considered coordinate axes of the feature space.

$$x_i = (x_{i1}, x_{i2}, \dots, x_{in}), \forall i \in \{1, 2, \dots, m\}$$

where:

- x_{ij} is the value of the *j* characteristic for the object *i*;
- n is the total number of analyzed characteristics.

Let x' be the new n-dimensional object given as input for which the algorithm returns its closest objects. Let f_d be the distance function used for the evaluation of similarity between two objects, with $f_d: \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$.

For the distance function, different distances can be used, such as: Euclidian distance, Manhattan, Canberra or Cosine.

The general model of searching and identification of the objects that are closest to a specified object is visually presented in figure 3. The objects' representation is done in the bi-dimensional space generated by two characteristics in which the red point represents the input point for which the neighbors are searched and the green objects are the results of the automatic search algorithm.

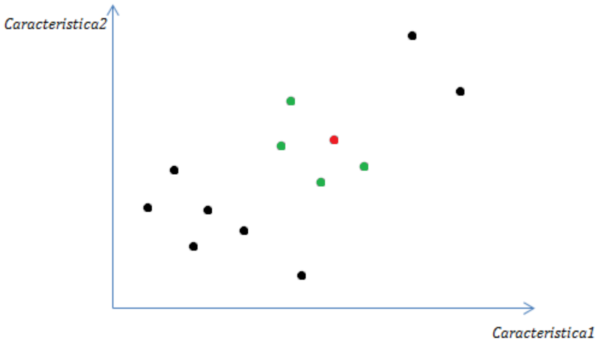


Fig. 3. General model of automatic search using kNN

The complexity level of the searching algorithm is equal to $O(n \times m + m^2)$. The disadvantages are given by the parameter k and distance function choosing. Also, the complexity level can be reduced by reducing the dimension of the feature space, using Principal Component Analysis, which generates a lower number of uncorrelated features.

4. Grid based optimized search

Different optimization algorithms are proposed in [3], [4], [5], [6], using the reduction of searching area, with the help of searching, representing and computing in an optimized manner. The problem of optimization is applied in the context of high dimensional space having a large number of objects.

Given the high complexity level of the searching algorithm using kNN, which depends of the dimension of the feature space, n, and the cardinality of the set of objects, m, new methods of comparing objects are proposed.

The process of optimization of the searching using kNN is done by reducing the complexity level of the algorithm through the cardinality diminishing of the features and the space searching reducing by:

- dividing the causal space into cells and sequentially search within the neighbour cells from which object x' is part of;
- the use of object clustering results of the initial objects from the X set and sequentially searching within the closest clusters.

The separation of the causal space within a matrix of cells starts from the concept of cell that is defined as being that zone from the feature space in which the objects assigned to it have the values of the characteristics in a predefined segment. For that, each value of the characteristics are retained and integrated in a transformation function for generating the cell of membership using the formula:

$$g_f(o_f) = \frac{o_f}{\dim_f} + 1, \forall o \in X$$

where:

- $g_f(o_f)$ represents the result of the assignation function of the f feature for the o object;
- $\dim_f$ represents the dimension for f characteristic.

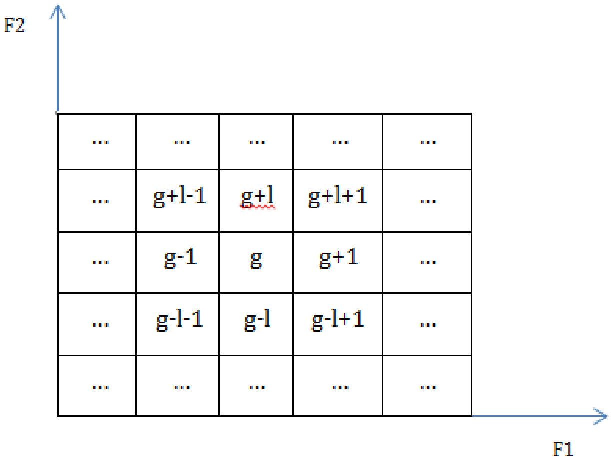
The dimension for each characteristic represents the size of the separation cells for the coordination axes of feature f, using the amplitude reported to the number of desired cells

$$dim_f = \frac{\max\limits_{i=\overline{1,m}} x_{if}}{l}$$

where:

- l represents the total number of segments in which the values are separated in.

Figure 4 reveals the manner of separation of a bi-dimensional feature space into cells. The numbering of the cells is done from left to right, from an inferior to a superior layer.



...	...	...	...	...
...	g+l-1	g+l	g+l+1	...
...	g-1	g	g+1	...
...	g-l-1	g-l	g-l+1	...
...	...	...	...	...

Fig. 4. Bi-dimensional space separation

The visual model of selecting the closest k neighbours using the optimization based on grid partitioning into cells is presented in figure 5. For the red object, the search is done only within the neighbour cells. The green points are the ones closest to the red object. Not all the points are used in the sorting algorithm.

After applying the formulas for each existing object, an assignation vector is loaded, $grid(x) : \mathbb{R}^n \to \mathbb{N}$. The manner of aggregation of the results of assignation of each feature is done using a linearization function, transforming a matrix into an array. For the bi-dimensional case, in which each object is characterized by two features, $g_f(o_1)=i$ and $g_f(o_2)=j$, the assignation result of the o object into a cell is generated by$grid(o) = (j-1) \times l + i$. The objects being assigned, the input object x' is also allocated, $grid(x') = g$.

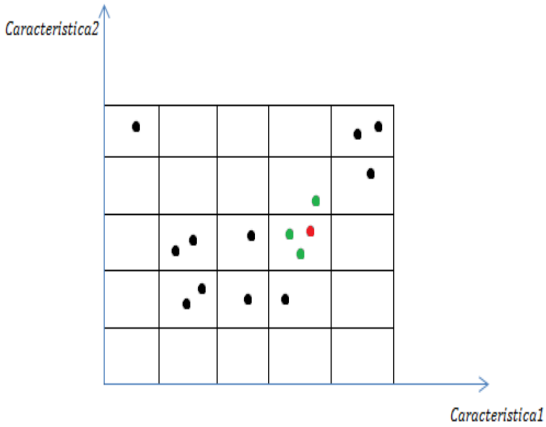


Fig. 5. Visual model of selecting using grid partitioning

The optimized kNN algorithm uses only the objects that are part of the same cell with x' or from the neighbour cells with g cell. A vector of visited cells is initialized with 0, along with a parameter *nr_obj*, representing the number of objects found among the neighbour cells. An empty list of objects is initialized, that will contain all the candidate neighbours to the input object.

The pseudocode proposed for the optimized automatic search contains the following steps.

P1. Determining the cell from which object x' is part of, *g*.

P2. If the cell isn't visited yet, the number of objects within it is counted. The visited objects are introduced in the list of neighbour objects.

P3. If the number of objects is less than k parameter, the search moves the one of the neighbour cells. For the bi-dimensional space, the labels of the neighbour cells to g one are *g-1, g+1, g+l+1, g+l-1, g-l+1* and *g-l-1*.

P4. Steps 2 and 3 are repeated until the number of identified objects is equal or greater than k.

P5. The identified objects allocated as nodes within the list of objects are further used for calculating the distances between them and x' object. After the sorting of the list, the first k objects are given as output.

The exit point from the algorithm is reached when the number of neighbour points is equal or greater than k. The optimization is done with the help of the algorithm proposed, by the manipulation of a lower number of objects, using a sequentially search.

Thereby, the new complexity level is$O(n \times m' + m'^2)$, where *m'* represents the size of the list of neighbor objects used in the sorting process.

5. Evaluation of the results

For the evaluation of kNN algorithm in general version used as compare base with all the objects from the initial set of objects, a set formed out of 900 bi-dimensional randomly generated objects is used.

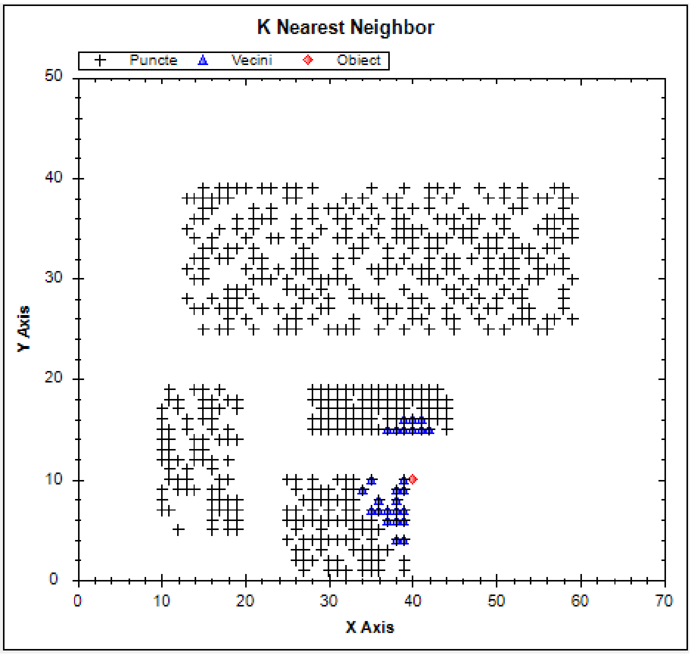


Fig. 6. Results of kNN algorithm

In figure 6 are presented the results obtained using the searching algorithm for k=50 bi-dimensional objects found closest to object $x'(40,10)$ from the initial set of $n=900$ objects. The distance function used the evaluation of the similarity between two objects is the Euclidian distance. From the complexity point of view, after calculating the distances of each 900 objects to the input object, the sorting is applied and the first 50 objects are given as output.

In the situation of optimized searching algorithm using grid based separation, the results are presented in figure 7.

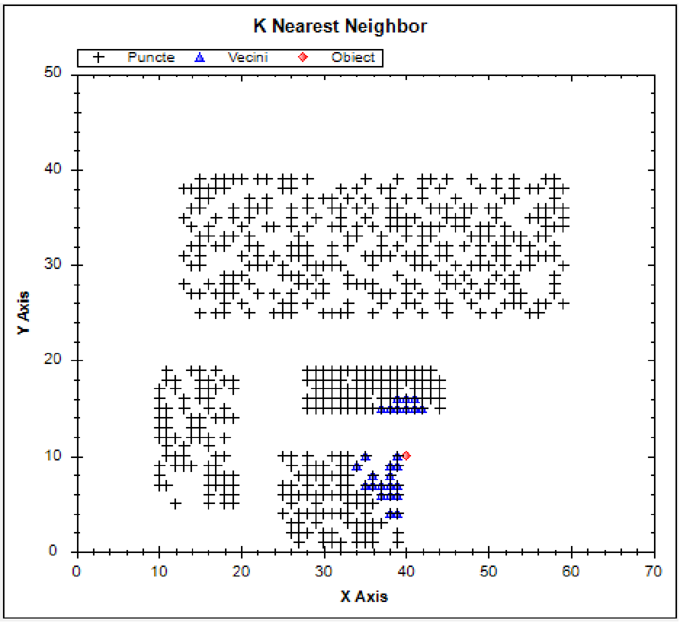


Fig. 7. The search results using kNN grid based algorithm

The number of resulted objects after running the algorithm is equal to *nr_obj = 97*, objects that are found around the cell from which object $x^{'}$ is part of. The 97 objects are sorted and the first 50 objects are returned.

The improvement of the algorithm by reducing the complexity level is found in the time consuming indicator reduced to 99% from the time consumed using the general automatic search algorithm in the context of maintaining the same searching results.

This percentage is resulted from the comparison of complexity levels, using the formula:

$$p = \frac{n \times m + n^2}{n' \times m + n'^2}$$

where:

- p represents the percentage of searching space reduction using an optimized searching algorithm compared to the searching algorithm among the total space of objects.

The cell labelled with 1 from figure 8 is the first cell for which the objects are retained, following this the cell 2 and 3. The process of automatic search ends when, introducing the last searching cell, the number of objects identified is at least equal to parameter k.

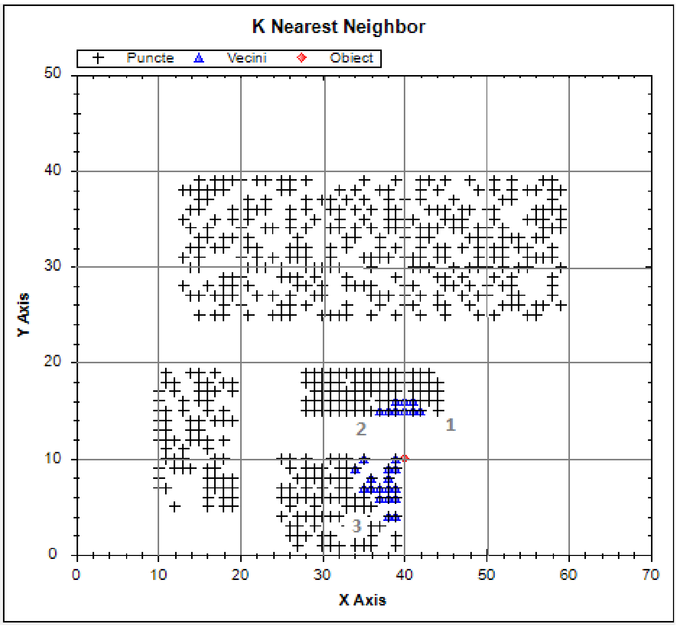


Fig. 8. The selection order of the cells in the process of automatic search

6. Conclusions

K Nearest Neighbour is an automatic search algorithm that generates the closest k objects in terms of distance measure of similarity between two objects from a given object as input data. The disadvantage of the general algorithm of automatic search is given by the high complexity level of comparing each object from the initial set of objects to the input searching object. This problem results in a high time consuming component. The proposed optimized model of automatic search divides the feature space into grid cells, and each object is assigned to the cell from which it is part of. The searching input object is assigned also to its cell and the algorithm of kNN is run on the objects found in the cells neighbour to the input cell until a stopping point is reached, representing the number of objects found in the cells analysed. The objects are then sorted in terms of distances between them and the searching object and the first k objects are returned as output.

The algorithm is suitable for document searching, when for each document a set of keywords is assigned manually or automatically using the proposed model for keywords generator. The feature space is formed out of the unique keywords found for each document from the initial set of documents.

The optimized algorithm reduces the time consuming, being analysed only the documents closest to the initial document using the grid cell analyser. For the experiment conducted, the time consuming is reduced with 99% from the time consumed using the general algorithm.

Future work is related to different optimization models for generating the closest documents.

Acknowledgments

This work was cofinanced from the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards”.

References

[1] McClure, R., „Writing Research Writing: The Semantic Web and the Future of the Research Project”, *Computers and Composition*, 28, 2011, 315–326

[2] Shi, Y., Wang, M., Qiao, Z., Mao, L., “Effect of Semantic Web technologies on Distance Education”, *Procedia Engineering*, 15, 2011, 4295 – 4299

[3] Kolbe, D., Zhu, Q., Pramanik, S., “Reducing non-determinism of k-NN searching in non-ordered discrete data space”, *Information Processing Letters*, 2010, 420-423

[4] Chen, Y.S., Hung, Y.P., Yen, T.F., Fuh, C.S., “Fast and versatile algorithm for nearest neighbor search based on a lower bound tree”, *Pattern Recognition*, 2007, 360-375

[5] Plaku, E., Kavraki, L.E., “Distributed computation of the knn graph for large high-dimensional point sets”, *Journal of Parallel Distributed Computation*, 2007, 346-359

[6] Lopez, V.F., Prieta, F., Ogihara, M., Wong, D.D., “A model for multi-label classification and ranking of learning objects”, *Expert Systems with Applications*, 2012, 8878-8884

Mădălina ZURINI is currently a PhD candidate in the field of Economic Informatics. She graduated the Faculty of Cybernetics, Statistics and Economic Informatics (2008) and a master in Computer Science, having her dissertation given in *Implications of Bayesian classifications for optimizing spam filters* (2010). She is also engaged in Pedagogical Program as part of the Department of Pedagogical Studies. Her fields of interest are data classification, artificial intelligence, data quality, algorithm analysis and optimizations. She wants to pursue a pedagogical career.

Diana BUTUCEA graduated in 2008 from the Economic International Relations Department of the Academy of Economic Studies in Bucharest, in 2009 from the Faculty of Automatics and Applied Informatics of the Transylvania University of Brasov and in 2010 from the Economic Informatics masters at the Academy of Economic Studies in Bucharest. Her parallel interests, in economy and software engineering, are now merging into her studies and researches since she is PhD student at the Academy of Economic Studies, Bucharest, studying integrated software systems, web technologies and e-learning platforms.