AN IDIOSYNCRATIC TOOL FOR CLUSTERING LEGAL DOCUMENTS USING PRINCIPAL DIRECTION DIVISIVE PARTITIONING ALGORITHM (PDDP) FROM WEBMINING

Mrs.V.Annapoorani MCA.,MPhil.,(PhD) [1] and Dr.A.Vijaya MCA.,MPHIL.,PhD [2]

[1]Lecturer in MCA DEPT, PAAVAI ENGINEERING COLLEGE
NAMAKKAL, TAMILNADU
annapoorani504@gmail.com

[2]Professor in Govt Arts and Science College SALEM, TAMILNADU
vijayakathiravan@gmail.com

ABSTRACT

This paper depicts an idiosyncratic tool for clustering legal documents for assisting lawyers and legal groups during document review from web mining. This tool clusters the data collection and generates a set of distinctive word labels for each cluster of documents, all entirely autonomous. This tool is based on the use of PDDP and K-Means, applied to both the document and the words, using a bag of words model. All this processing by the tool occurs without input from a human user, except to specify the original document set. In this thesis, we address the name of civil cases in orderly fashion so as to retrieve rapid comprehension of the retrieved document from the web mining.

Clustering is a technique used to group and divide underlying data based on their similarities and dissimilarities. It discovers both the dense and the sparse regions in a data set. Both the academic and document researchers rely on this technique. This method is unusual. It operates by repeatedly splitting clusters into smaller clusters. Though it's frequently clustering methods, we have little knowledge about the characteristics of available clustering methods or how clustering methods should be deployed. Cluster analysis is referred to as segmentation method in document research. It means segmenting or separating a particular data from the given set of data available. In this paper, we discuss the most classical problems that occur while clustering underlying data. The first problem is which cluster must be split and the second problem is how to split the selected cluster. This paper clusters the data collection and generates a set of distinctive word labels for each cluster of documents, all entirely autonomous.

KEYWORDS: *PDDP, FORGY'S ALGORITHM, LATENT SEMANTIC INDEXING ALGORITHM(LSI), LINEAR LEASE SQUARE FIT(LLSF)ALGORITHM.*

1. INTRODUCTION

In this idiosyncratic tool spreads into a large database. This tool is of sorting all the civil cases documents in alphabetical order from the web mining. This tool used the bisecting divisive clustering approach for sorting the civil case documents automatically. Unsupervised clustering of documents is a critical component for the exploration of large unstructured document sets.

Natarajan Meghanathan, et al. (Eds): SIPM, FCST, ITCA, WSE, ACSIT, CS & IT 06, pp. 425–436, 2012.
 DOI : 10.5121/csit.2012.2342

Unsupervised clustering is an important because the volume of data available on the web is enormous.

-the problem of selecting which cluster must be split.

-the problem of how to split the selected cluster.

This paper focuses on the first sub-problem. In particular, in this paper used a new method for the selection of the cluster to split is proposed. This paper discusses a scalable unsupervised algorithm that is capable of automatically generating automated clusters, while identifying the attributes or words that are the most important in distinguishing between the variety of legal documents in a given collection. This method is here presented with reference to two specific bisecting divisive clustering algorithms.

-The bisecting k-means algorithm.

-The PDDP.

This paper is organized as follows : Section2 the bisecting k-means and PDDP algorithms are concisely recalled, whereas in section 3 selecting the cluster to split(a quick review), section 4 discussing the theoretical results for Infinite data sets, section 5 display the Numerical results for finite data sets and section 6 will compare the clustering performance of both algorithm. Empirical evaluation and performance are presented in section 7.

2. BISECTING K-MEANS AND PDDP

K-means is the most celebrated and widely used clustering technique (see eg., [1],[2],[3]) hence, it is the best representative of the class of iterative centroid-based divisive algorithm. On the other hand PDDP is a recently proposed technique[4],[5],[6],[7]. It is representative of the non-iterative technique based upon the Singular Value Decomposition (SVD) of a matrix built from the data set.

The clustering approach considered here in bisecting divisive clustering. Namely, we want to solve the problem of splitting the data-matrix. $M=[X_1,X_2,\ldots\ldots X_N)€R^{P*N}$(where each column of M_1 X_1 € R^P, is a single data point) into two sub-matrices (or sub – clusters) M_L € R^{P*NL} and M_R € $R^{P*N}{}_R$, $N_L + N_R = N$.

In such algorithms, the definition of “centroid” will be used extensively, specifically, the centroid of M, say w is given by,

$w = 1/N \sum_{j=1}^{N} xj$ -------------------- [1]

where x_j is the j^{th} column of M. similarly, the centroids of the sub-clusters M_L and M_R say W_L and W_R, are computed as the average value of their columns.

$\{W_L = \frac{1}{N_L} \sum_{j=1}^{N_L} M_{Lj}$

$W_R = \frac{1}{N_R} \sum_{j=1}^{N_R} M_{Rj}\}$------------------[2]

BISECTING K-MEANS:

Step1: (Initialization) Randomly select a point as $C_L€R^{P,}$ then compute the centroid of M, and compute $C_R€R^P$ as $C_R = w-(C_L-w)$.

Step 2: Divide M = $[X_1,X_2,\ldots\ldots X_N]$ into two sub-clusters M_L and M_R, according to the following rule.

$X_i € M_L$ if $|| x_i – C_L || \le || x_i – C_R||$

$X_i € M_R$ if $|| x_i – C_L || \le || x_i – C_R||$

Step 3: compute the centroids of M_L and M_R, W_L and W_R as in Eq[2].

Step 4: If $W_L = C_L$ and $W_R = C_R$ stop. Otherwise, let $C_L := W_L$, $C_R := W_R$ and go back to step 2.

The algorithm above presented is the bisecting version of the general k-means algorithm. This algorithm is claimed to be very effective in document processing and content-retrieving problems.

The very classical and basic version of K-means also known as Forgy's algorithm. Many variations of this basic version of the algorithm have been proposed, aiming to reduce the computational demand at the price of (hopefully little) sub-optimality.

PDDP(Principal Direction Divisive Partitioning Algorithm)

Step 1: Compute the centroid w of M as in Eq.[1]

$$W = \frac{1}{N}\sum_{j=1}^{N} Xj \qquad \text{------- [1]}$$

Step 2: compute the auxiliary matrix $\tilde{M}$ = M-we, where e is a N-dimensional row vector of ones, namely e = [1,1,1,1,1…….1].

Step 3: compute the singular value decomposition(SVD) of $\tilde{M}$, $\tilde{M} = U\sum V^T$, where $\sum$ is a diagonal PXN matrix and U and V are orthogonal unitary square matrices having dimension PXP and NXN respectively.

Step 4: Take the first column vector of U, say u = U and divide M = $[x_1, x_2,\ldots\ldots x_n]$ into two sub clusters M_L and M_R according to the following rule.

$X_i € M_L$ if $U^T (x – w) \leq 0$

$X_i € M_R$ if $U^T (x – w) \leq 0$

This PDDP algorithm proposed SVD based data processing algorithms among them, the most popular and widely known as the Latest Semantic Indexing Algorithm(LSI) and LSI – related Linear Lease Square Fit(LLSF) algorithm.

PDDP and LSI mainly differ in the fact that the PDDP splits the matrix with a hyper plane passing through its centroid LSI through the origin. Another major features of PDDP is that the SVD of $\tilde{M}$ (Step 3) can be stopped at the first singular value/vector. This makes PDDP significantly less computationally demanding than LSI. Especially, if the data matrix is sparse and the principal singular vector is calculated by resorting to the Lanczos Technique.

The main difference between k-means and PDDP is that k-means is based upon an iterative procedure, which in general, provides different results for different initializations, whereas PDDP is a "One – Shot" algorithm, which provides a unique solution. In order to understand better, how k-means and PDDP work, in Fig1(a) and Fig1(b) the partition of a matrix of dimension provided by k-means and PDDP respectively is displayed.

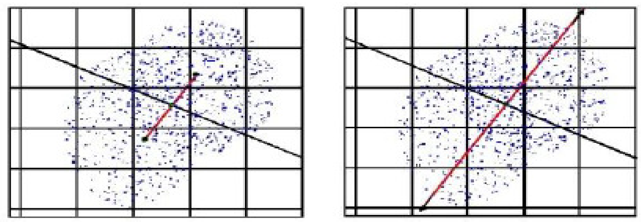

Fig 1(a)K-Means and Fig 1(b)

From Fig1, it is easy to see how k-means and PDDP work.

- The bisecting k-means algorithm splits M with a hyperplane which passes through the centroid w of M, and is perpendicular to the line passing through the centroids of w_L and w_R of the sub-clusters M_L and M_R.

- PDDP splits M with a hyperplane which passes through the centroid w of M, and is perpendicular to the principal direction of the "unbiased" matrix $\tilde{M}$ ($\tilde{M}$ is the translated version of M, having the origin as centroid).

At first glance, the two clusters provided by k-means and PDDP look almost indistinguishable. A more careful analysis reveals that the two partitions differ by a few points. Note that this is somewhat unexpected, since two algorithms differ substantially.

3. SELECTING THE CLUSTER TO SPLIT: a quick review

The problem of selecting the cluster to split in divisive clustering techniques may have a remarkable impact on the overall clustering results.

The following three classes of approaches are typically used for the selection of the cluster to split[14]:

(A) complete partition: every cluster is split, so obtaining a complete binary tree.

(B) The cluster having the largest number of elements is split.

(C) The cluster with the highest variance with respect to its centroid.

$\alpha(M) = \frac{1}{N_R} \sum_{j=1}^{N_R} M_{R_j} \|X_j - w\|^2\}$-----------------[2]is split(w is the cetroid of data matrix of the cluster, X_j its j-th column, $\| \|$is the Euclidean norm).

The above criteria are extremely simple and raw. Criterian (A) is indeed a "non-choice", since every cluster is split: it has the advantage of providing a complete tree, but it completely ignores the issue of the quality of the clusters. Criterian (B) is also very simple. It does not provide a complete tree, but it has the advantage of yielding a "balanced" tree, namely a tree where the leaves are (approximately) of the same size. Criterian(C) is the most "sophisticated", since it is based upon a simple but meaningful property(the "scatter") of a cluster. This is the reason why (C) is the most commonly used criterion clusterselection.

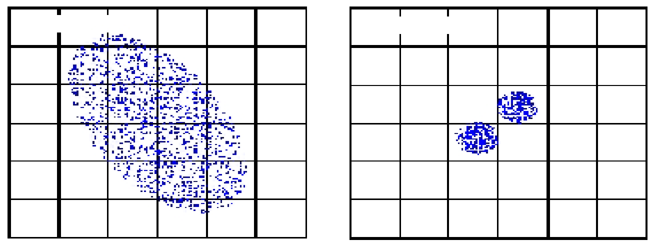

Fig.2(a) size 2X2000 Fig.2(b) 2 x 1000

The main limit of the above criteria can be pictorially described with a naïve example. In Fig.2 two data sets are displayed: the first is a matrix of size 2X2000(Fig.2(a); the second is a matrix of size 2 x 1000(Fig.2(b). by inspecting the two data sets, it is apparent that the best cluster to split is the second one; it is inherently structured into two sub-clusters.

Both criterion (B) and (C), however, would suggest the first one as the best cluster to split. It has the largest number of data points, and the largest variance ($\alpha(M) = 0.53$ for the first data- set and $\alpha(M) = 0.10$ for the second set).

It is interesting to observe that the main limit of criteria (A-C) is that they completely ignore the "shape" of the cluster, which is known to be a key indicator of the extent to which a cluster is well-suited to be partitioned into two sub-clusters. This simple but crucial observation, however, deserves some additional comments:

Taking into account the "shape" of the cluster is a difficult and slippery task, which inherently requires more computational power than the computation of the simple criterion(2). Henceforth, in computationally-intensive applications the simplicity of criteria (A-C) can be attractive; however, in many applications characterized by a comparative small number of data-points(N) and features(p), a better criterion than (A-C) would be appealing.

Taking into account the "shape" of the cluster requires a wise balancing between an application-specific approach and a multi-purpose approach. If the criterion is too application-specific it can only be helpful for that application. If too generate(as A-Care), it cannot provide high clustering performance.

4. THEORETICAL RESULTS FOR INFINIT DATA SETS:

In this section, the "asymptotic" behavior of bisecting k-means and PDDP will be analysed. Asymptotic here means that the data set has an infinite an infinite number of points, namely $N \rightarrow \infty$ difference between a finite and an infinite set of points is naively depicted.

In the first part of this section, we will focus on the 2-dimensional case, specifically, it is assumed that each point $X = [X_1 , X_2]^T$ of the data set belongs to an ellipsoid centered in the origin and referred to the axes. $X = [X_1 , X_2]^T$ belongs to the data set

$$\frac{x_1^2}{a^2} + x_2^2 \leq 1 \text{ -----------------[3]}$$

The semi-axes lengths of the ellipsoid in Eq[3] are a ($0 < a \leq 1$) and 1 respectively. The following two prepositions, generated to P dimensions.

Preposition 1: if the data points of a data set are uniformly distributed in a 2-dimensional ellipsoid, the semi-axes of the ellipsoid have lengths equal to 1 and a, ($0 < a \leq 1$) and N --- ∞, then the dynamic discrete – time system which models the k-means itereative algorithm is characterized by 2 equilibrium points, I point is locally unstable and 1 is locally stable.

In particular , the dynamic model has the form: $a_t + 1 = \text{atan}(a^2 \tan(\alpha_t))$, $0 < a < 1$, $0 \leq a_t \leq \pi/2$. The splitting hyper plane corresponding to the stable equilibrium points is orthogonal to the main axis of the ellipsoid.

The splitting hyper plane corresponding to the stable equilibrium points is orthogonal to the largest axis of the ellipsoid.

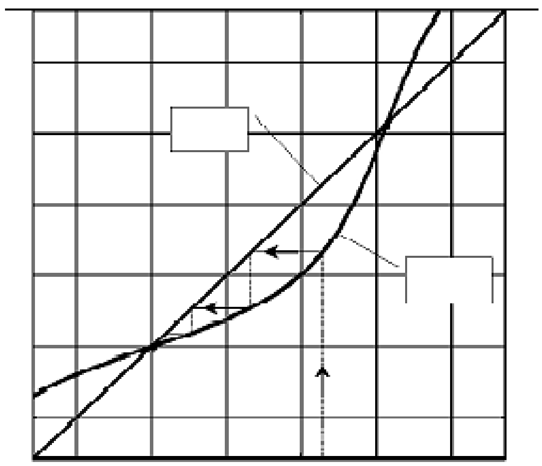

Fig 3 Proof: the proof of this result is given in items:

Preposition 2: If the data points of a data set are uniformly distributed in a 2-dimensional ellipsoid, the semi-axes of the ellipsoid have lengths equal to 1 and a_1($0 < a_1 \leq 1$) and $N \longrightarrow \infty$, then the PDDP algorithm splits the ellipsoid with a hyper plane passing through the origin and orthogonal to the largest axis of the ellipsoid.

Proof: The result is a direct implication of the properties of the SVD. Indeed the 2 singular vectors of a set of points uniformly distributed within an ellipsoid coincide with the direction of the principal axes of the ellipsoid.

Prepositions1 and 2 show that bisecting k-means and PDDP provide the same solution, except in the case when the initialization of k-means exactly corresponds to an unstable equilibrium point of the k-means dynamic model. However, if the initialization is made randomly, this event occurs with probability zero.

5. NUMERICAL RESULTS FOR FINITE DATA SET:

In this section, the bisecting k-means and PDDP will be analyzed, when the data set has a finite number of data points. The analysis will be done empirically using simulated data.

The purpose of this section is two fold:

Validate the theoretical results can change when the data set is finite.
Understand the pros and cons of k-means and PDDP.

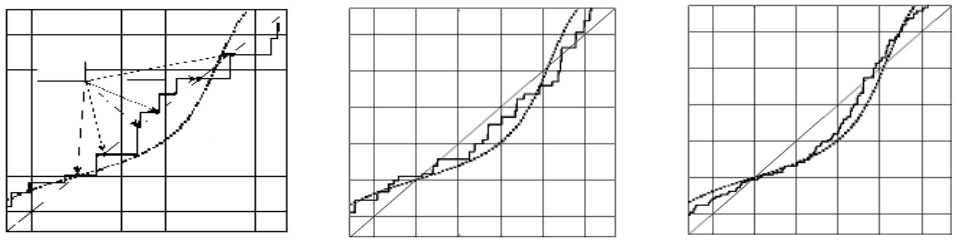

FIG:4 RECURSIVE FUNCTION

Four sets of data have been considered, characterized by 15, 30, 100 and 2000 data points uniformly distributed within a 2-dimensional ellipsoid with a=0.6. the recursive function $\alpha_t + 1 = f(\alpha_t)$ has been numerically computed for these data sets.

A major consequence of this function being step-like is that every equilibrium point (namely, every point where the function crosses the line $\alpha_t + 1 = \alpha_t$ is locally asymptotically stable, since the local slope of the function about the equilibrium is smaller than 1. This explains why, in the case of finite data sets, the k-means algorithm is affected by bad local minima problem.

When the number of data points grows, the finite data set function converges towards the asymptotic function. (see fig 5d). This explains why, when the number of data is sufficiently large, it is the common experience that the problem of local minima tends to vanish.

These experiments suggest that the problem of local minima for bisecting k-means is expected to:

- Decrease when the number of data grows.
- Increase when the size of the "short" semi-axes(a_{ij} ….a_{p-1}) approaches the largest axis.

In order to validate these conjectures, the bi-secting k-means algorithm has been extensively tested for different values of a (α = 0.6, 0.7, 0.8, 0.9) and for different sizes of the data set(N ranging from 10 to 5000). The average dispersion of the centroids, we have obtained (which is directly related to the problem of local minima) is displayed in Fig.5 In particular, for each value of N, 20 different data sets have been randomly generated for each dataset, 100 different runs of k-means have been done(starting from different initial conditions), so obtaining 100 "dispersed" centroids. The dispersion of these 100 centroids has been computed for each of the 20 data sets and averaged. The conjuctures above outlined are fully confirmed by the data, the centroids dispersion increases with and decreases with N. Estimated standard deviation of the centroids.

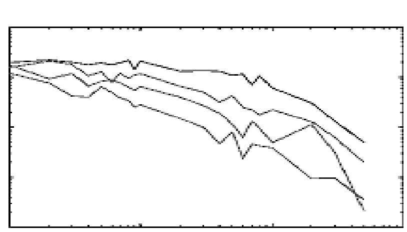

Fig 5: Average Dispersion of Centroids

6. COMPARING THE CLUSTERING PERFORMANCE OF BI-SECTING K-MEANS AND PDDP:

In order to compare the performance of k-means and PDDP, we have considered two sets of data, uniformly distributed in a 100-dimensional ellipsoid. The main semi-axis of the ellipsoid is 1. The size of the remaining 99 semi-axes is in the range[0.05, 0.95]. The first data set has 1000 points and the second data set 5000.

Note that, this number of data point is comparatively small for a 100-dimensional vector space. For each data set, the following clustering techniques have been used.

(a) Bi-Secting k-means initialized randomly, specifically 1000 different initializations have been tested for each data set.

(b) PDDP

(c) Bi-Secting k-means, initialized with the result provided by PDDP.

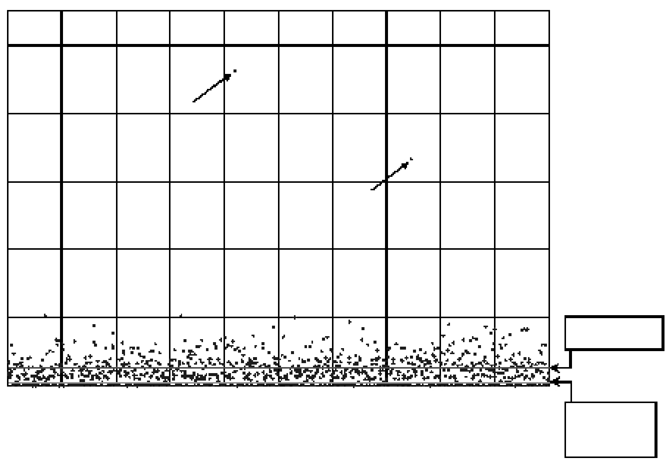

Fig.6 Measure of quality of bisecting partition of a data set of N= 5000 points.

At the end of each clustering experiment the so-obtained partition has been evaluated using Eq.[1]. The results are displayed in Fig6(N=1000) and in Fig10(N=5000). The measure of quality is a normalized version of Eq.[1]specifically, 0 corresponds to the best clustering performance, 1 corresponds to the "worst-case" situation of non-partitioned cluster(namely $M_L = M$ and $M_R = \theta$). The 1000 dots show the clustering performance of k-means randomly initialized, the two horizontal lines show the performance of PDDP and the performance of k-means initiailized via PDDP.

7. EVALUTION AND PERFORMANCE:

To assess the performance of our clustering approach , we clustered the set of headnoted case law documents on Indian law. This includes 1 million civil case law documents and collectively contains more than 10 million headnotes classified to about 1,00,000 key numbers.

Table1: A sample set of clusters and their metadata.

ClusteriD	Label	Noun Phrases	Key Numbers
12799254	Eminent Domain/ Compensation/ Nature of Property and Neighborhood and Estimated Replacement Value	"fair cash" "right way" "land owner right access"	148k221 148k14(1) 148k107
12436744	Opinion evidence/challenged portion of Expert Witness Opinion	"admissible expert testimony" "quality expert witness" "product liable action"	157k536 157k546 157k539
12957372	Automobiles/Arrest, stop or Inquiry; Bail or deposit/ stop and arrest by police officers	"Police officers""valid trafficstop" "defendant drive license"	48Ak34(4) 35k63.5(6) 48Ak349(2)
13080456	Child custody/ removal from jurisdication/ best interests and welfare of children and Right of other parent	"admissibleexperttestimony"modify child custody""non custodial parent child"	76Dk261 76Dk76 76Dk921(1)

8. LEGAL ISSUE DETECTION QUALITY, LEGAL ISSUE CLUSTERING QUALITY:

The objective of the assessment in this evaluation was two fold:

First fold is the clustering algorithm able to discover all the legal clusters in these documents.

Second fold is the clustering algorithm able to find the common legal issue in each of the reports.

We used precision P and recall R to measure the performance for the first objective, in which:

-P is defined by the number of correctly identified clusters of a document(compare to the manually identified clusters)divided by the total number of clusters and

-R is defined by the number of correctly identified clusters of a document divided by the total number of manually identified clusters of a document.

For the second objective, we used precision P for evaluation, which is defined by the number of common legal issues identified among documents in all reports divided by the number of common legal issues manually identified among documents in all reports by experts.

Regarding the first objective, the precision and recall of the system on 10 reports across different document type is shown in the Figure 7. Overall, we achieved reasonably high precision, but the recall was quite low especially for case law documents. The main reason for this is the aggressive filtering, by adopting much higher thresholds, in the past processing of the system to achieve high precision.

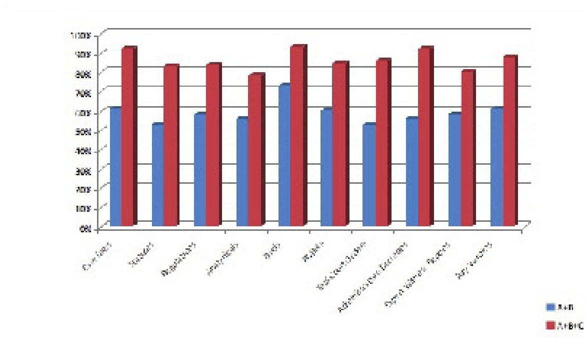

Fig.7 10 reports across different document type

For the second objective, Table2 shows the performance of the system in each individual reports as well as overall.

Table 2: Precision of common Legal Issues

Report IDs	Total No.of Documents	Documents with common theme Editorially Created Associations	Documents with common theme algorithmically created associations
AE_2	6	6	6
AE_20	6	6	6
AE_22	4	4	4
AE_24	5	3	3
AE_31	6	6	6
AE-32	7	5	5
AE_35	6	6	5
AE_36	6	6	6
AE_37	5	3	3
AE_39	7	7	7
TOTAL	58	52	51
PRECISION		89.7%	87.9%

In this analysis, each of the 10 reports has a common legal issue running through it. However, the common 'thread' did not appear in each document in each of the reports.

In summary, the experts manually created clusters by identifying a common thread through all documents in 7 of the 10 reports, our system identified a common thread in one of the documents in that report, and is thus considered as a failure.

Across the entire set, experts manually created clusters and identified the common thread in 52 of the 58 documents(89.7%). Our system created clusters and identified the "common thread" in 51 of the 58 documents (87.9%).

9. CONCLUSION:

This paper examined how clustering technique is acquired successfully by using k-means algorithm and PDDP algorithm by applying them to document clustering in order to yield more useful results. This case study shows the efficacy of this method in document clustering. In this case study PDDP algorithmic methods were applied in addition to k-means and this yielded a significant improvement in the results obtained. This research tool is very efficient and it effectively stores, sorts and retrieves the legal documents in an alphabetical order automatically with the structure of a concentric circle.

Area of future study could be the expansion of the scope to consider other algorithms such as PMPDDP, partitioning algorithm, BIRCH Algorithm etc, in this type of sorting and retrieving tools makes it much easier for lawyers and judges to do any reference. By storing the law and order files by this way also provides a great assistance to them.

REFERENCES:

1. E. Forgy, Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification, *Biometrics* (1965), 768-780.
2. A.K. Jain, M.N. Murty and P.J. Flynn, Data Clustering: a Review, *ACM Computing Surveys* **31**(3) (1999), 264-323.
3. M.W. Berry, Z. Drmac, E.R. Jessup, Matrices, Vector spaces, and Information Retrieval, *SIAM Review* **41** (1999), 335-362.
4. D.L. Boley, Principal Direction Divisive Partitioning, *Data Mining and Knowledge Discovery* **2**(4) (1998), 325-344.
5. D.L. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher and J. Moore, Document Categorization and Query Generation on the World Wide Web Using WebACE, *AI Review* **11** (2000), 365-391.
6. Bottou, Y. Bengio, *Convergence properties of the k-means algorithm*, Advances in Neural Information Processing Systems, G.H. Golub and C.F. van Loan, *Matrix Computations*, (3rd Ed.), The Johns Hopkins University Press, 1996. E. Gose,
7. document clustering. In *Proc. of International Joint Con- ference on Neural Networks IJCNN*, pages 2545-2549, 2002. H. Kozima. Text segmentation based on similarity between words full text. In *Proc. of the ACL*, pages 286-288, 1993.
8. H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the ACL*, pages 232-239, 1993.
9. J. Lin and C. Dyer. Data-intensive text processing with mapreduce. *Synthesis Lectures on Human Language Technologies*, 2010. Apache mahout overview. http://lucene.apache.org/mahout/, 2010.
10. A. McCallum, K. Nigam, and L. Ungar. Efcient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD00)*, pages 169-178, 2000.
11. D. Merkl and E. Schweighofer. En route to data mining in legal text corpora: Clustering, neural computation, and international treaties. In *Proceedings of the 8th International Workshop on Database and Expert Systems Applications (DEXA '97)*, 1997.
12. C. Ordonez and E. Omiecinski. Frem: Fast and robust em clustering for large data sets. In *Proceedings of CIKM*, pages 590-599, 2002.
13. M. Shafiei and E. Milios. A statistical model for topic segmentation and clustering. *Lecture Notes in Computer Science*, 5032, 2008. Svm light. http://svmlight.joachims.org/, 2010.

14. A. Tagarelli and G. Karypis. A segment-based approach to clustering multi-topic documents. In *Proceedings of the Text Mining Workshop, SIAM Data Mining Conference*, 2008.

15. M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the ACL*, pages 499-506, 2001.

16. H. Kozima. Text segmentation based on similarity between words full text. In *Proc. of the ACL*, pages 286-288, 1993.

17. H. Kozima and T. Furugori. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the ACL*, pages 232-239, 1993.

18. J. Lin and C. Dyer. Data-intensive text processing with mapreduce. *Synthesis Lectures on Human Language Technologies*, 2010.Apache mahout overview. http://lucene.apache.org/mahout/, 2010.

19. A. McCallum, K. Nigam, and L. Ungar. Efcient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD*

20. *International Conference on Knowledge Discovery and Data Mining (KDD00)*, pages 169-178, 2000.