Optimization of the K-means Clustering Algorithm through Initialized Principal Direction Divisive Partitioning

Bruce James

Mentor: Dr. Jacob Kogan

July 14, 2017

University of Maryland, Baltimore County

Abstract

Data clustering is invaluable to the automated analysis of large document sets. Documents are converted into vectors in a finite dimensional space, and the resulting collection of salient features is then processed through an algorithm of one's choice, such as the classic k-means clustering algorithm. Due to the size of the feature space, different algorithms offer a trade-off between accuracy and computational efficiency. This study investigates the Principal Direction Divisive Partitioning (PDDP) algorithm, described as a top-down hierarchical technique, as a plug-in to the k-means algorithm. K-means reliance on initial random partitioning builds computational cost into the analysis. Using a PDDP initialized partition to seed k-means, computational efficiency will be compared to a k-means trial without PDDP.

Introduction

Clustering techniques have found numerous uses within the disparate fields of research that rely on data analysis. A cursory list of beneficiaries includes data mining, genomics, bioinformatics, and signal processing, with the field of text mining providing an historic impetus in its own right. Since the advent of the World Wide Web, cluster analysis has further developed in tandem with the rapidly increasing volume of document data existing online. Where human-based classification of text is assisted by an unsupervised approach, the organization and retrieval of documents from large data sets invariably becomes an automated task. This research investigates the Principal Direction Divisive Partitioning (PDDP) clustering algorithm, as a plug-in to the well-known k-means algorithm, comprising one such automated technique [2].

The k-means algorithm has long been an industry standard in partition-based clustering of texts. Document sets are grouped together and relevant features, in the form of "content-bearing words" [3], are extracted and used to create a high-dimensional vector space. Individual documents would then exist as very sparse vectors (i.e. most words would be assigned to zero, as any given document uses only a small fraction of the lexical set) in the vector space. Then, the k-means algorithm would provide a similarity-measure for clusters of documents having affinity within a group, compared to clusters outside of the group.

Finding the desired cluster partitions then becomes an optimization problem. The k-means is a random partitioning method, relying on a seed partition to initialize the algorithm [3]. The algorithm converges very fast, and even on large data sets this is not a problem. However, there is no guarantee that k-means will converge to the global minimum. Convergence of the algorithm to the global minimum is NP-hard [4]. In contrast, PDDP relies on an initial "directed" partitioning, as in the program described by Boley [2]. Features of the document cluster are projected onto a vector subspace, where the orientation of the subspace "directs" the initial partitioning. We are interested in a comparison of k-means clustering that is initialized by a PDDP partitioning, against an unassisted k-means clustering.

The majority of this work will appeal to conventional linear algebra and optimization techniques that underwrite the validity of the above-mentioned algorithms. First, the relevant terminology will be developed from the literature. Then, the
similarity measure will be defined in terms of a Euclidean norm. Details will be presented on the linear algebra concepts involved in making the initial directed partitioning, as well as the principles of optimization that justify the k-means technique [1].

The goal of this investigation extends further to the coding of a working computer program, which will run the k-means algorithm with, and then without PDDP. The results of both trials will then be compared, where it is anticipated that the PDDP trial will run more efficiently than the solo k-means trial. A demonstrable efficiency in the PDDP initialized program may suggest that initial directed partitioning of large text documents is superior to randomly seeded partitioning, in both computational runtime and accuracy of document clustering.

Literature Review

The character of many types of clustering methods depends on the applications involved. Document analysis differ from other applications like spatial recognition and robotic vision; sample spaces in which large document sets rely, do not necessarily possess the regularity of extended space [4]. From its human-generated origins, text typically contains irregularity, where the addition of new text evolves in unpredictable ways [3]. Historically, probabilistic methods have been used to anticipate this irregularity. Ultimately this ignores vast amounts of text, and greatly affects the accuracy of the analysis [2]. Contrasting with probabilistic means, the popular k-means algorithm is deterministic in its implementation, often with strategic use of random partitioning [2].

The k-means algorithm has long been an industry standard in partition-based clustering of texts. As such, much of the developed literature starts with the procedure’s shortcomings when performing highly specialized tasks. Yet the algorithm has also been used as a benchmark in the literature for measuring the performance of many other boutique procedures. Here, a short description of the k-means algorithm will provide a wider coin of vantage to the survey of document clustering.

In the field document analysis, mathematical and information theoretic tools are used to prepare text for an objective treatment. First, document sets are grouped together and relevant features, in the form of “content-bearing words” [3], are extracted and used to create a high-dimensional vector space. Individual documents would then exist as very sparse vectors (i.e. most words would be assigned to zero, as any given document uses only a small fraction of the lexical set) in the vector space. Then, the k-means algorithm would provide a similarity-measure for clusters of documents having affinity within a group, compared to clusters outside of the group.

One computational consideration that arises when running k-means, originates from the random selection of centers used to initialize the algorithm. With no unique starting point, many local minima may be found. The algorithm converges very fast, and even on large data sets this is not a problem. However, there is no guarantee that k-means will converge to the global minimum. Convergence of the algorithm to the global minimum is NP-hard. This exceeds realistic computational time constraints, and requires some additional heuristics to reduce computational effort [4].

As shown in Dhillon, et al. [4], one such heuristic, the spherical k-means technique, was developed to work with the limitations in the k-means random partitioning. After multiple iterations, k-means tends to become stuck in local minima, and not fully realize the optimal clustering scheme. To address optimization, Dhillon, et al. [4] suggested a “ping-pong” strategy that increases the computational efficiency of k-means by forming a sequence of separate clusters and moving certain documents from one cluster to another. The objective function could be studied to find a better optimization, thereby circumventing the local minimum problem. As noted by the authors of that study, the approach was limited to static amounts of documents clustered. Further work would be required for large variances in sizes of the initial document set.

Optimization was extended further in Kogan, et al. [7] by developing the distance-like function from combining the Squared-Euclidean distance with an information theoretic quantity. It was suggested that the resulting similarity measures could be tailored to specific data sets. Whole classes of variations on the k-means theme could then be studied.

Limitations to the standard k-means algorithm have also been treated with various hybrid approaches [5], using an algorithm like PDDP to guide the k-means algorithm into trajectories with higher quality. The principal direction divisive partitioning (PDDP) algorithm under consideration in this proposed study was developed by Boley who used a “divisive” method, where initially large document sets were divided into smaller partitions [2]. Boley defined “principal direction” as a process of “directing” each iteration of division by a new computation of the document space. This process further developed the distance function and similarity measure, thereby increasing computational efficiency [2].

The mathematical treatment implemented in the PDDP algorithm, can be summarized as follows. The projection of a set of vectors onto the nearest line starts the principal direction. From there, the problem reverts to maximizing the eigenvalue of the covariance matrix, by using the power method [6]. This result can also be carried over to the calculation of the Frobenius norm, in conjunction with the Lanczos algorithm. Through Lanczos, a sequence of diagonal matrices are constructed. Then, eigenvalues of the matrices are computed by finding the convergence of the largest eigenvalue. Applying these linear algebra techniques significantly reduces computational complexity and run-time [1], thereby rendering PDDP more practical for use in high-dimensional cluster analysis.

Research Objective

The ultimate objective of this study is to investigate a hybrid application of the PDDP algorithm to initialize the k-means clustering algorithm, as a way to reduce the computational effort exerted by the stand-alone k-means algorithm. The reduction in computational cost would compensate for the irregularity of text data, and enable more accuracy for sorting and querying from large document sets. A comprehensive review of the optimization performed on the PDDP algorithm, as well as cluster validation measures, will inform further investigations on the quality of the resulting clusters, and suggest further work on the strengths and limitations of hybrid approaches.

Analysis of the PDDP Algorithm

High-dimensional vector spaces, equipped with some distance-like function, are the standard environment for large document sets. This study begins with an overview of the linear algebra and optimization techniques, which underwrite these spaces. With deference to the notation of the prior works, vectors representing documents with m features will be represented as boldfaced, lowercase letters, namely, $\mathbf{b} = [b_1, b_2, ..., b_m]^T$, and a collection of n documents will be denoted by the matrix $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$.

We start by finding the projection, $\mathbf{p}_{\mathrm{a}}$ on a line L in $\mathbf{R}^n$, parameterized by $\mathbf{y} + t\,\mathbf{x}$, on which is projected the document vector $\mathbf{a}$. Note that the vector $(\mathbf{y} - t_0\,\mathbf{x}) - \mathbf{a}$ is orthogonal to L, and also the vector $\mathbf{x}$ at the point of projection, for some t_0 , namely

$$(\mathbf{y} + t_0\,\mathbf{x} - \mathbf{a})^{\mathrm{T}}\,\mathbf{x} = 0 \;\;\Rightarrow\;\; \mathbf{y}^{\mathbf{T}}\mathbf{x} + t_0\,\|\mathbf{x}\|^2 - \mathbf{a}^{\mathbf{T}}\mathbf{x} = 0$$

$$\Rightarrow\;\; t_{\,0} = (\mathbf{a}^{\mathbf{T}}\mathbf{x} - \mathbf{y}^{\mathbf{T}}\mathbf{x})/\,\|\mathbf{x}\|^2.$$

So, $\mathbf{p}_{\mathrm{a}} = \mathbf{y} + t_0\,\mathbf{x}$, where $t_0 = (\mathbf{a}^{\mathbf{T}}\mathbf{x} - \mathbf{y}^{\mathbf{T}}\mathbf{x})/\,\|\mathrm{x}\|^2$, which is the point that $\mathbf{a}$ projects on L. With this, then $(\mathbf{a} - \mathbf{p}_{\mathrm{a}})^{\mathrm{T}}\mathbf{x} = 0$. If we take $\mathbf{x}$ to lie on the unit sphere, and $\mathbf{y}$ orthogonal to $\mathbf{x}$, such that $\mathbf{x}^{\mathbf{T}}\mathbf{x} = 1 \Rightarrow \|\mathbf{x}\|^2 = 1$, and $\mathbf{y}^{\mathbf{T}}\mathbf{x} = 0$, then

$$\begin{aligned}\mathbf{p}_{\mathrm{a}} &= \mathbf{y} + \;[(\mathbf{a}^{\mathbf{T}}\mathbf{x} - \mathbf{y}^{\mathbf{T}}\mathbf{x})/\,\|\mathbf{x}\|^2]\;\mathbf{x}\\ &= \mathbf{y} + (\mathbf{a}^{\mathrm{T}}\mathbf{x})\;\mathbf{x}.\end{aligned}$$

In this form, we hope to find the least squares approximation of a set of document vectors. To do this, we will minimize the sum of the distances of each projection $\mathbf{p}_i$ on L, with $\mathbf{a}_i$. In terms of $\mathbf{x}^{\mathbf{T}}\mathbf{x} = 1$,and $\mathbf{y}^{\mathbf{T}}\mathbf{x} = 0$**,** the sum, then is

$$\Sigma_{1\le i\le n}\;\;|\mathbf{a}_i - (\mathbf{y} + (a_i^{\,T}x)\;\mathbf{x})|^2.$$

Regrouping and simplifying each term $|\mathbf{a} - \mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x} - \mathbf{y}|^2$ of the sum, gives

$$\begin{aligned}|(\mathbf{a} - \mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x}) - \mathbf{y}|^2 &= (\mathbf{a} - \mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 - 2\mathbf{y}(\mathbf{a} - \mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x}) + \mathbf{y}^2\\ &= \mathbf{a}\cdot\mathbf{a} - 2\mathbf{a}(\mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x}) + (\mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 - 2\mathbf{a}\cdot\mathbf{y} + 2\mathbf{y}\cdot\mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{x} + \mathbf{y}\cdot\mathbf{y}\\ &= \mathbf{a}\cdot\mathbf{a} - 2(\mathbf{a}\cdot\mathbf{x})(\mathbf{a}\cdot\mathbf{x}) + (\mathbf{x}\cdot\mathbf{x})(\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 - 2\mathbf{a}\cdot\mathbf{y} + 2(\mathbf{y}\cdot\mathbf{x})\mathbf{a}^{\mathrm{T}}\mathbf{x} + \mathbf{y}\cdot\mathbf{y}\\ &\mathbf{=}\; \mathbf{a}\cdot\mathbf{a} - 2(\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 \;+ (1)(\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 - 2\mathbf{a}\cdot\mathbf{y} + 2(\mathbf{0})\mathbf{a}^{\mathrm{T}}\mathbf{x} + \mathbf{y}\cdot\mathbf{y}\\ &= \mathbf{a}\cdot\mathbf{a} - (\mathbf{a}^{\mathrm{T}}\mathbf{x})^2 - 2\mathbf{a}^{\mathrm{T}}\mathbf{y} + \mathbf{y}\cdot\mathbf{y}\\ &= (\mathbf{a} - \mathbf{y})^2 - (\mathbf{a}^{\mathrm{T}}\mathbf{x})^2.\end{aligned}$$

So,

$$\Sigma_{1\le i\le n}\;\;|\mathbf{a}_i - (\mathbf{y} + (a_i^{\,T}x)\;\mathbf{x})|^2 = \Sigma_{1\le i\le n}\;\;|\mathbf{a}_i - \mathbf{y}|^2 - |\mathbf{a}_i^{\mathrm{T}}\mathbf{x}|^2.$$

Assuming a value for $\mathbf{x}$, we now use the method of Lagrange multipliers to find $\mathbf{y}$, such that

$$\min_{\mathrm{y}} \left\{\Sigma_{1\le i\le n}\ |\mathbf{a}_i - \mathbf{y}|^2, \text{ subject to } \mathbf{y}^{\mathrm{T}}\mathbf{x} = 0\right\}. \qquad (2)$$

Let $f(\mathbf{y})$ be the vector-valued objective function $f(y_1,\ldots,y_n) = \Sigma_{1\le i\le n}\ |\mathbf{a}_i - \mathbf{y}|^2$, and $g(\mathbf{y}) = \mathbf{y}^{\mathrm{T}}\mathbf{x} = 0$ be the constraint function. Taking the gradient of f yields

$$\nabla f(\mathbf{y}) = \Sigma_{1\le j\le n}\ f_j(y_1,\ldots,y_n)\ \mathbf{e}_j,$$

where f_j is the partial derivative of f with respect to the j-th component, and $\mathbf{e}_j$ is the unit basis vector of the j-th component, namely $\mathbf{e}_j = [0,\ldots,\ \mathbf{e}_j = 1,\ldots,0]^{\mathrm{T}}$.

With $|\mathbf{a}_i - \mathbf{y}|^2 = \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i - 2\mathbf{a}_i^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y}$ for each i, then

$$\nabla f(\mathbf{y}) = \Sigma_{1\le j\le n}\ \partial/\partial y_j\ \left[\Sigma_{1\le i\le n}\ |\mathbf{a}_i - \mathbf{y}|^2\right]\mathbf{e}_j$$

$$= \Sigma_{1\le j\le n}\ \partial/\partial y_j\ \left[\Sigma_{1\le i\le n}\ \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i - 2\mathbf{a}_i^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y}\right]\mathbf{e}_j$$

Since we are in $\mathbf{R}^n$, and the 2-norm is an assignment from $\mathbf{R}^n$ to $\mathbf{R}$ ($|\mathbf{a}_i - \mathbf{y}|^2$ is the Euclidean norm assigning the n-vector $\mathbf{y}$ to a real number), then from a theorem in real analysis, f is continuous. So, from another property of analysis, we can interchange the summand with partial derivative, yielding

$$\nabla f(\mathbf{y}) = \Sigma_{1\le j\le n}\,\Sigma_{1\le i\le n}\ \partial/\partial y_j\ \left[\mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i - 2\mathbf{a}_i^{\mathrm{T}}\mathbf{y} + \mathbf{y}^{\mathrm{T}}\mathbf{y}\right]\mathbf{e}_j$$

$$= \Sigma_{1\le j\le n}\,\Sigma_{1\le i\le n}\ \partial/\partial y_j\ \left[\mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i - 2(a_{ji}y_j) + y_j^2\right]\mathbf{e}_j$$

$$= \Sigma_{1\le j\le n}\left[\Sigma_{1\le i\le n}\ (-2a_{ji} + 2y_j)\ \mathbf{e}_j\right]$$

$$= \Sigma_{1\le j\le n}\left[\Sigma_{1\le i\le n}\ (-a_{ji})\ \mathbf{e}_j + \Sigma_{1\le i\le n}\ y_j\ \mathbf{e}_j\right]$$

$$= 2\left[\Sigma_{1\le i\le n}\ (-\mathbf{a}_i) + \Sigma_{1\le i\le n}\ \mathbf{y}\right]$$

$$= 2n\mathbf{y} - 2 \textstyle\sum_{1\le i\le n} \mathbf{a}_i.$$

Now, taking the gradient of *g,* gives

$$\nabla g(\mathbf{y}) = \textstyle\sum_{1\le j\le n} \partial/\partial y_j\, [y_j\, x_j]\, \mathbf{e}_j$$

$$\nabla g(\mathbf{y}) = \partial/\partial y_j\, [y_1 x_1 + \ldots + y_n x_n]\mathbf{e}_j$$

$$= \textstyle\sum_{1\le j\le n}\ (x_j)\mathbf{e}_j$$

$$= \mathbf{x}.$$

Employing the Lagrange multiplier, and solving the system for lambda yields

$$\nabla f(\mathbf{y}) = \lambda \nabla g(\mathbf{y}) \quad \Rightarrow \quad 2n\mathbf{y} - 2 \textstyle\sum_{1\le i\le n} \mathbf{a}_i = \lambda\, \mathbf{x}$$

Now, left-multiplying by $\mathbf{x}^{\mathrm{T}}$, and using the conditions $\mathbf{y}^{\mathrm{T}}\mathbf{x} = 0 \Rightarrow \mathbf{x}^{\mathrm{T}}\mathbf{y} = 0$, and $\mathbf{x}^{\mathrm{T}}\mathbf{x} = 1$,

$$\mathbf{x}^{\mathrm{T}}\, (2n\mathbf{y} - 2 \textstyle\sum_{1\le i\le n} \mathbf{a}_i\,) = \mathbf{x}^{\mathrm{T}}\, \lambda\, \mathbf{x}$$

$$\Rightarrow \qquad \mathbf{x}^{\mathrm{T}}\, (2n)\mathbf{y} - \mathbf{x}^{\mathrm{T}}\, (2) \textstyle\sum_{1\le i\le n} \mathbf{a}_i = \lambda\, (\mathbf{x}^{\mathrm{T}}\, \mathbf{x})$$

$$\Rightarrow \qquad (2n)\mathbf{x}^{\mathrm{T}}\mathbf{y} - 2\mathbf{x}^{\mathrm{T}} \textstyle\sum_{1\le i\le n} \mathbf{a}_i = \lambda\, (\mathbf{x}^{\mathrm{T}}\, \mathbf{x})$$

$$\Rightarrow \qquad \lambda = -2\mathbf{x}^{\mathrm{T}} \textstyle\sum_{1\le i\le n} \mathbf{a}_i.$$

Substituting for lambda gives

$$2n\mathbf{y} - 2 \textstyle\sum_{1\le i\le n} \mathbf{a}_i = -2\mathbf{x}^{\mathrm{T}} \left(\textstyle\sum_{1\le i\le n} \mathbf{a}_i\, \right)\mathbf{x}$$

$$\Rightarrow \qquad \mathbf{y} = 1/n \left[\textstyle\sum_{1\le i\le n} \mathbf{a}_i - \mathbf{x}^{\mathrm{T}} \left(\textstyle\sum_{1\le i\le n} \mathbf{a}_i\, \right)\mathbf{x}\right],$$

with **x** and its transpose as known entities. So, to minimize (2),

$$\mathbf{y} = 1/n \left[\textstyle\sum_{1\le i\le n} \mathbf{a}_i - \mathbf{x}^{\mathrm{T}} \left(\textstyle\sum_{1\le i\le n} \mathbf{a}_i\, \right)\mathbf{x}\right].$$

Now, we turn to the collection of column vectors representing the document set. First, we note the following property:

Property 1:

For $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, with $A = \mathbf{b}_1\mathbf{b}_1^{\mathrm{T}} + \mathbf{b}_2\mathbf{b}_2^{\mathrm{T}} + \ldots + \mathbf{b}_\mathrm{n}\mathbf{b}_\mathrm{n}^{\mathrm{T}}$, then $BB^{\mathrm{T}} = A$.

Proof:

Since matrix B is an m x n array and B^{T} is an n x m array, then BB^{T} is an

m x m array. Similarly for A, each $\mathbf{b_k b_k^T}$ term, where

$$\mathbf{b_k b_k}^{\mathrm{T}} = [b_{\mathrm{k1}}\ b_{\mathrm{k2}} \ldots b_{\mathrm{km}}\ ]^{\mathrm{T}}\ [b_{\mathrm{k1}}\ b_{\mathrm{k2}} \ldots b_{\mathrm{km}}\ ],$$

is an m x m array, whose sum for $1 \leq k \leq n$ is also an m x m array. Let the ij^{th} entry of BB^{T} be denoted by c_{ij}. Then, performing matrix multiplication yields,

$$c_{ij} = b_{i1}b_{1j} + b_{i2}b_{2j} + \ldots + b_{im}b_{mj}.$$

Let the ij^{th} entry of $\mathbf{b_k b_k}^{\mathrm{T}} = d^{(k)}{}_{ij}$. Carrying out matrix multiplication gives $d^{(k)}{}_{ij} = b_{ik}b_{kj}$. Performing addition for $1 \leq k \leq n$ yields,

$$\Sigma_{1\leq k\leq n}\, d^{(k)}{}_{ij} = b_{i1}b_{1j} + b_{i2}b_{2j} + \ldots + b_{im}b_{mj}.$$

This sum is the ij^{th} entry of matrix A. Since $c_{ij} = \Sigma_{1\leq k\leq n}\, d^{(k)}{}_{ij}$ for the ij^{th} entries of BB^{T} and A respectively, therefore,

$$BB^{\mathrm{T}} = A.$$

Now, we want to exploit the symmetric and positive semi-definite properties of the matrix A. For

Property 2:

A is positive semi-definite.

Proof:

From the associative properties of matrix multiplication,

$$\mathbf{x}^{\mathrm{T}}\mathbf{B}\ \mathbf{B}^{\mathrm{T}}\mathbf{x} = (\mathbf{x}^{\mathrm{T}}\mathbf{B})(\mathbf{B}^{\mathrm{T}}\mathbf{x}).$$

Since $\mathbf{x}^\mathrm{T}\mathbf{B} = (\mathbf{B}^\mathrm{T}\mathbf{x})^\mathrm{T}$, and $\mathbf{x}^\mathrm{T}\mathbf{B}$ is the transpose of a column vector $\mathbf{B}^\mathrm{T}\mathbf{x}$, then $(\mathbf{x}^\mathrm{T}\mathbf{B})(\mathbf{B}^\mathrm{T}\mathbf{x})$ is a dot product of $\mathbf{B}^\mathrm{T}\mathbf{x}$ with itself. From the positivity property of inner product spaces (i.e. $\langle v,v\rangle \geq 0$, for all $v \in \mathbf{R}^n$), then

$$\mathbf{x}^\mathrm{T}\mathbf{B}\ \mathbf{B}^\mathrm{T}\mathbf{x} \geq 0.$$

So, A is positive semi-definite.

With the fact that A is positive semi-definite, then we know also that all of the eigenvalues of A are real. We know this from the real spectral theorem, and since **A** can be decomposed into the product $\mathbf{UDU}^\mathrm{T}$, where **U** is an orthogonal matrix and **D** is diagonal, with real entries on the main diagonal equal to the eigenvalues of **A**. Hence, the eigenvalues are real. *(From Elementary Linear Algebra, Ron Larson, 7th Ed., p.362; also http://web.mit.edu/jorloff/www/18.03-esg/notes/symmetricMatrices.pdf)*

We state the next property:

Property 3:

Every eigenvalue of A is non-negative.

Proof:

Since **A** is symmetric, then for any arbitrary eigenvalue λ of **A** satisfying $\mathbf{Ax} = \lambda\mathbf{x}$, $\lambda \in \mathbf{R}$, left multiplying $\mathbf{x}^\mathrm{T}$ to both sides of $\mathbf{Ax} = \lambda\mathbf{x}$, yields

$$\mathbf{x}^\mathrm{T}\mathbf{Ax} = \mathbf{x}^\mathrm{T}\lambda\mathbf{x} = \lambda\mathbf{x}^\mathrm{T}\mathbf{x}.$$

Since **A** is positive semi-definite,

$$\mathbf{x}^\mathrm{T}\mathbf{Ax} \geq 0 \;\Rightarrow\; \lambda\mathbf{x}^\mathrm{T}\mathbf{x} \geq 0,$$

and by positivity of inner product spaces,

$$\mathbf{x}^T\mathbf{x} \geq 0 \quad \Rightarrow \quad \lambda \geq 0.$$

Therefore, the eigenvalues of A are non-negative.

Methodology

The latter work of this study explores two programs running the PDDP and k-means algorithms over a set of document data. The algorithms are written in Python. Documents are procured from web-based sources, such as MEDLINE, for sample research abstracts and larger documents. An off-the-shelf porter stemmer from the National Institute of Standards and Technology is required to reduce the morphological complexity of the texts.

Procedure

In many applications, the elements of a vector have mostly zero values. Such a vector is said to be sparse, yet intermediate steps are required to operate on large and irregular matrices. From the specifications determined in the preliminary analysis, the initial directed partitioning is coded first in pseudo code, and then in Python.

The clustered documents come from varied disciplines, with optimal variance (as dictated in the preliminary treatment). Two sets of trials are made, with one trial set consisting of a collection of research abstracts, and another trial set from the bodies of the research papers. Each document set is then run through a porter stemmer, and the resulting morphologically reduced data set is then fed into the PDDP initialized k-means program, and again, through the solo k-means program.

Analysis

Computational run-time is measured in Python. In keeping with prior research, clustering validation measures are employed to determine the quality of the final clusters. The accuracy of each trial is determined through a confusion matrix [7]. Anticipating a uniform effect from the back-end of both k-means applications (with and without PDDP), the clusters are qualitatively assessed by comparing the "true" class size with the experimental results. Since entropy-based assessments of the resulting clusters may not adequately measure uniform effects, a Coefficient of Variation statistic can, here be applied to the trial clusters [8].

Expected Results

The results of this investigation will be in keeping with the findings of the prior work in partition-based cluster analysis; limitations in the k-means random partitioning will be seen throughout the trials [2,3,4]. This study proposes that a PDDP initialized k-means algorithm will converge on relative minima much more quickly than a stand-alone k-means approach, and that these minima will be closer to the global minimum. As PDDP operates on sparse, high dimensional vectors, the hybrid is expected to converge faster in the trials using large-sized documents, when compared to solo k-means [3]. Likewise, the trials on the smaller research abstract set are not expected show considerable performance differences between the assisted and stand-alone algorithms. From these results, the PDDP hybrid would promise to make large document text mining more efficient.

It is proposed that the accuracy of the final clustering results, as determined through the confusion matrix [7], will be greater in the PDDP initialized trials on large-sized documents. The rationale here stems from the suspected advantage that directed partitioning used in PDDP has, over the random initializations of the k-means stand-alone. This effect is expected to abate in the trials with large variance in the text data, where uniform effects belie "true" class sizes [8]. Here, it is expected that before and after changes in the coefficient of variation will be observed. This would suggest further work is needed to increase the responsiveness of partitioning methods to wider variances in the initial data.

References

[1] Berry, M. W., Dumais, S. T., and O'Brien, G. W. Using linear algebra for intelligent information retrieval. *SIAM Review*, (4), 573, 1995.

[2] Boley, D. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, *2*(4), 325–34, 1998.

[3] Dhillon, I. S., Guan, Y., and Kogan, J. Iterative clustering of high dimensional text data augmented by local search. In *IEEE International Conference on Data Mining, 2002. Proceedings.* 131–138, 2002.

[4] Hamerly, G., Elkan, C., Kalpakis, K., Goharian, N., and Grossman, D. Alternatives to the k-means algorithm that find better clusterings. In K. Kalpakis, N. Goharian, and D. Grossman (Eds.). Presented at the Proceedings of the Eleventh International Conference on Information and Knowledge Management. CIKM, 2002.

[5] Kogan, J. Hybrid clustering of large text data. In *21st International Conference on Advanced Information Networking and Applications Workshops, 1*, 2007.

[6] Kogan, J. *Introduction to clustering large and high-dimensional data*. Cambridge New York : Cambridge University Press, 2007.

[7] Kogan, J., Teboulle, M., and Nicholas, C. Data driven similarity measures for k means like clustering algorithms. *Information Retrieval, (8)* 331-349, 2005.

[8] Wu, J. *Advances in K-means clustering. [electronic resource] : a data mining thinking.* Berlin ; New York : Springer, pages 17-35. 2012.