**JUSTICE NII-AYITEY**

**STAT517 LITERATURE REVIEW REPORT**

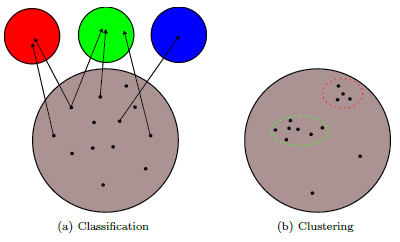
**RECENT DEVELOPMENTS IN DOCUMENT CLUSTERING**

**ABSTRACT**

Clustering under unsupervised learning is one of the sectors in machine learning which is really advancing, and this report aims to elaborate on the current state of document clustering research and to present its recent developments in a well-organized way. We would also focus on discriminative and generative algorithms as the two main forms of clustering algorithm and identify what we assume as good, bad or ugly clustering. As part of this report too, we would know the reasons behind some of the recent developments with probably their strength and weaknesses respectively.

**INTRODUCTION**

Document (or text) clustering is a subset of the larger field of data clustering, which borrows concepts from the fields if information retrieval (IR), natural language processing (NLP), and machine learning (ML). Just as statistics is used to represent populations, we therefore refer to document clustering simply as clustering. Clustering should not be confused with classification because they are two different concepts. In the context of machine learning, classification is supervised learning and clustering is unsupervised learning. Classification has prior knowledge of classes and classify new sample into known classes. It uses Decision Trees, Bayesian classifiers, etc. as its algorithms and has labeled samples from a set of classes. On the other hand, clustering has no prior knowledge of classes and suggest groups based on patterns in data and uses K-means, Expectation Maximization, etc. as its algorithms with unlabeled samples. *Figure 1* below distinct the two concepts in a diagram.



*Figure1: In (a), three classes are known a priori, and documents are assigned to each of them. In (b), an unknown number of groupings must be inferred from the data based on a similarity criterion (in this case, distance).*

**DISCUSSION (SOME RECENT DEVELOPMENTS IN CLUSTERING)**

For the purpose of this report, we are going to focus on this content below:

* Good, Bad, and Ugly Clustering
* Vector Space Model
* Dimensionality Reduction
* Extensions to *k*means
* Generative Algorithms
* Spectral Clustering
* Phrase-Based Models
* Comparative Analysis

**Good, Bad, and Ugly Clustering**

There are many doubts as in to which clustering approach is the best but unfortunately, there is little agreement over which is the best way to do so. Clustering simply is said to be good when collections are organized into groups such that each group has similar documents and comparatively different to other groups. The choice of evaluation methods frequently depends on the domain in which the research is being conducted. Artificial Intelligence researcher for instance, might prefer to use mutual information while someone from the field of Information Retrieval would choose F-measure. Precision and recall are two intuitive ways to check for performance of clustering. In Information Retrieval, precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Both precision and recall are therefore based on an understanding and measure of relevance. Given R as recall and P as precision, the generalized F-measure is defined as;

Fα=

**Vector Space Model**

Most existing clustering approaches choose to represent each document as a vector. This reduces a document to a representation suitable for traditional data clustering approaches. Under the vector model, a collection of *n* documents with *m* unique terms is represented as an *m \* n* term-document matrix (where each document is a vector of *m* dimensions). Two important properties should be noticed here. First, in a collection of heterogeneous topics, the number of unique terms will be quite large which results in document vector of high dimensionality and second, a matrix resulting from a typical corpus under the vector model will be highly sparse, since the corpus contains many more terms than the individual documents that compose it. Preprocessing consists of steps that take as input a plain text document and output a set of tokens to be included in the vector model and these steps includes filtering, tokenization, stemming, stopword removal, and pruning.

**Dimensionality Reduction**

In statistics, machine learning, and information theory, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables (Roweis and Saul 2000). It can be divided into feature selection and feature extraction and below are some few examples of how dimension reduction can be done.

1. Principal component analysis (PCA); they are orthogonal (i.e. uncorrelated) projections that together explain the maximum amount of variation in a dataset. In practice, principle components can be found by computing the singular value decomposition on the correlation matrix of the dataset. *Figure 2* below demonstrates PCA.

A close up of a map

Description automatically generated

*Figure 2: The two dashed lines represent PCA’s capturing the variability in the dataset*

1. Non-negative matrix factorization (NMF); it decomposes a non-negative matrix to the product of two non-negative ones (i.e. the technique breaks the original term-document matrix *A* into the matrices), which has been a promising tool in fields where only non-negative signals exist (Lee and Seung 1999). It was originally developed for computer vision applications but has been effectively used for document clustering.
2. Soft spectral coclustering; it induces membership weights from the disjoint partitions of words and documents. That is, a soft partition of documents can be induced from a partition of terms, and a soft partition of terms can be induced from a partition of documents.
3. Kernel PCA; principal component analysis can be employed in nonlinear way by means of the Kernel trick and the result technique can construct nonlinear mappings that maximize the variance I the data. The resulting technique is entitled Kernel PCA.

**Extensions to *k*means**

Historically, hierarchical and partitional algorithms have been the dominant clustering methods. In hierarchical clustering, each document is initially its own cluster and its algorithms work by successively merging documents, converging at the root of the hierarchy (also called a dendrogram or tree), which includes all documents in the corpus (“bottom up” clustering). Advantage of this method are that several clusters need not be supplied in advance, and that the resulting cluster hierarchy is “browsable”. Partitional methods (e.g. *k*means) start by choosing *k* initial documents as clusters and iteratively assign documents to clusters while updating the centroids of these clusters (“top down” clustering).

It is well known that text data is directional, and so it is typical to normalize document vectors, and to use a cosine similarity measure rather than Euclidian distance. The resulting algorithm is called spherical *k*means (cite). Below are some of the extensions to *k*means;

1. Online spherical kmeans (oskmns); it is an extension of the original spherical *k*means (skmns) that uses competitive learning techniques to speed up clustering while achieving similar or better accuracy.

A close up of a logo

Description automatically generated

*Figure 3: (a) kmeans has trouble clustering and (b) kernel means successfully separates the two clusters*

1. Kernel kmeans; figure 3 above illustrates a dataset which *k*means has trouble correctly clustering because the points are not linearly separable. The idea behind kernel *k*means is to find a mapping to a higher dimensional space where it is more likely that the documents can be linearly separated.

**Generative Algorithms**

Clustering algorithms can be divided into two types which are;

1. Discriminative algorithms; they classify points without providing a model (i.e. its algorithms operate on pairwise similarities between every document). A classical example is the *k*means.
2. Generative algorithms; they assume an underlying distribution of the data (i.e. its algorithms make structure assumptions on the model) and below are some few examples;
3. Gaussian model; it represents a dataset as a set of means and covariance matrices. Under this model, each cluster is centered at the mean and is described by its associated matrix.
4. Expectation maximization (EM); its algorithm is an efficient iterative procedure to compute a Maximum Likelihood (ML) solution to a model (Neal and Hinton 1998). It is composed of two steps;
5. Expectation step (E-step); the missing data is estimated given the observed data (the collection of documents) and current estimate of the model (the clusters).
6. Maximization step (M-step); the likelihood function is maximized under the assumption that the missing data are known.
7. von Mises-Fisher model (vMF); its distribution is the analogue of the Gaussian distribution for directional data.
8. Model-based *k*means; this is a more constrained (disjoint) version of the EM algorithm. This algorithm alternates between a model re-estimation step and a sample re-assignment step, resulting in runtime complexity.

**Spectral Clustering**

A matrix is a natural representation for adjacency information between vertices, and therefore the vector model can be interpreted as a graph. Spectral clustering involves finding cuts (*e.g. ratio cut, normalized cut, and min-max cut*) in graphs to produce good clusters (i.e. its algorithm aims to optimize). Few examples are;

1. Divide & merge clustering; this is the notion of conductance, measures “how tightly nit" a graph is and has been proposed as a criterion for finding optimal cuts in a graph (Kannan et al. n.d.). The divide and merge clustering algorithm proceeds in two phases. First, a hierarchical clustering is produced by recursive cuts on the graph resulting from the term-document matrix. The second phase finds a tree-respecting clustering from the output of the divide phase.
2. Fuzzy coclustering; regular fuzzy clustering algorithms capture the *degree* to which documents belong to each cluster but fuzzy coclustering implicitly assigns degrees of membership to words as well. Thus, the difference between fuzzy coclustering and regular coclustering is that the boundaries between clusters in the former case fuzzified in accordance with certain membership functions.

**Phrase-Based Models**

They naturally describe clusters by phrases and it is generally agreed that these are more descriptive of the cluster contents. Some examples are;

1. Suffix tree clustering (STC); this was originally developed for web snippet clustering and uses the tokens output from the preprocessing phase to construct a suffix tree structure (Zamir and Etzioni n.d.).
2. Document index graph (DIG); it is like the suffix tree model, but it encodes word order information and defines similarity based on matches in word order. Thus, the DIG represents each word as a vertex in a directed multigraph. The main differences between the DIG and the suffix model is that the DIG stores words explicitly as vertices, and maintains frequency information at each vertex, thereby avoiding storing redundant information.

**Comparative Analysis**

Two major things to be considered under comparative analysis are;

1. Query clustering; this refers to the grouping documents returned from an IR query, such as a Google search, into clear equivalence classes. An important consideration in this case is the highly variable nature of the set of documents returned.
2. Collection clustering; this refers to the grouping of documents in a static collection. This is an easier problem than query clustering, as the model selection and initialization problems can both be addressed by brute force strategies. This amounts to, simply put, “seeing what works best.” For instance, kernel clustering has shown superior accuracy for some collections, but the choice of kernel function is not always obvious and can depend on the collection.

**CONCLUSIONS**

We have discussed quite several recent developments in document clustering but one thing to be noticed is that, they both have their strengths and weakness and with that, the most “natural” grouping is not always produced. One other interesting thing is that, it has been noted that clustering is ultimately in the eye of the beholder (Estivill-Castro and Vladimir 2002) . There are many amounts of literature on clustering but unfortunately, there is little agreement over which is the best way to do so. In conclusion, the choice of evaluation methods frequently depends on the domain in which the research is being conducted.

**REFERENCES**

Estivill-Castro, V., and Vladimir (2002), “Why so many clustering algorithms,” ACM, 4, 65–75. https://doi.org/10.1145/568574.568575.

Kannan, R., Vempala, S., and Vetta, A. (n.d.). *On Clusterings: Good, Bad and Spectral*.

Lee, D. D., and Seung, H. S. (1999), “Learning the parts of objects by non-negative matrix factorization.,” 401, 788–91. https://doi.org/10.1038/44565.

Neal, R. M., and Hinton, G. E. (1998), “A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants,” in *Learning in Graphical Models*. https://doi.org/10.1007/978-94-011-5014-9\_12.

Roweis, S. T., and Saul, L. K. (2000), “Nonlinear dimensionality reduction by locally linear embedding.,” 290, 2323–6. https://doi.org/10.1126/science.290.5500.2323.

Zamir, O., and Etzioni, O. (n.d.). *Web Document Clustering: A Feasibility Demonstration*.