ACONS: A New Algorithm for Clustering Documents

Andrés Gago Alonso, Airel Pérez Suárez, and José E. Medina Pagola

Advanced Technologies Application Center (CENATAV),
7a ♯ 21812 e/ 218 y 222, Rpto. Siboney, Playa, C.P. 12200, La Habana, Cuba
{agago,asuarez,jmedina}@cenatav.co.cu

Abstract. In this paper we present a new algorithm for document clustering called Condensed Star (ACONS). This algorithm is a natural evolution of the Star algorithm proposed by Aslam *et al.*, and improved by them and other researchers. In this method, we introduced a new concept of star allowing a different star-shaped form; in this way we retain the strengths of previous algorithms as well as address previous shortcomings. The evaluation experiments on standard document collections show that the proposed algorithm outperforms previously defined methods and obtains a smaller number of clusters. Since the ACONS algorithm is relatively simple to implement and is also efficient, we advocate its use for tasks that require clustering, such as information organization, browsing, topic tracking, and new topic detection.

Keywords: Clustering, Document processing.

1 Introduction

Clustering is the process of grouping a set of data objects into a set of meaningful subclasses, called clusters; these clusters could be disjoint or not. A cluster is a collection of data objects that have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

Initially, document clustering was evaluated for improving the results in information retrieval systems [9]. Clustering has been proposed as an efficient way of finding automatically related topics or new ones; in filtering tasks [2] and grouping the retrieved documents into a list of meaningful categories, facilitating query processing by searching only clusters closest to the query [10].

Several algorithms have been proposed for document clustering. One of these algorithms is Star, presented and evaluated by Aslam *et al.* [1]. They show that the Star algorithm outperforms other methods such as Single Link and Average Link in different tasks; however, this algorithm depends on data order and produces illogical clusters. Another method that improves the Star algorithm is the Extended Star method proposed by Gil *et al.* [6]. The Extended Star method outperforms the original Star algorithm, reducing considerably the number of clusters; nevertheless this algorithm can leave uncovered objects and

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 664–673, 2007.

in some cases produce unnecessary clusters. Another version of the Extended Star method was proposed by Gil *et al.* to construct a parallel algorithm [7]. However, this version also has some drawbacks.

In this paper we propose a new clustering method, called Condensed Star or ACONS. In ACONS, we introduced a new definition of star allowing a different star-shaped sub-graph, in this way we retain the strengths of previous algorithms as well as solve the above-mentioned drawbacks. The experimentation – comparing our proposal against the original Star and the Extended algorithms – shows that our method outperforms those algorithms.

The basic outline of this paper is as follows. Section 2 is dedicated to related work. Section 3 contains the description of the ACONS method. The experimental results are discussed in section 4. The conclusions of the research and some ideas about future directions are exposed in section 5.

2 Related Work

In this section we analyze the Star algorithm and two proposed versions of the Extended Star method for document clustering, and we show their drawbacks.

The Star algorithm was proposed by Aslam *et al.* in 1998 [1], with several extensions and applications in filtering and information organization tasks [2,3]. They formalized the problem representing the document collection by its similarity graph, finding overlaps with dense sub-graphs; it is done so because the clique cover of the similarity graph is an NP-complete problem, and it does not admit polynomial time approximation algorithms. With this cover approximation by dense sub-graphs, in spite of loosing intra-cluster similarity guarantees, we can gain in computational efficiency.

Let $V = \{d_1, ..., d_N\}$ be a collection of documents and $Sim(d_i, d_j)$ a similarity (symmetric) function between documents d_i and d_j, we call similarity graph to an undirected and weighted graph $G = \langle V, E, w \rangle$, where vertices correspond to documents and each weighted edge corresponds to the similarity between two documents. Considering a similarity threshold σ defined by the user we can define a thresholded graph G_σ as the undirected graph obtained from G by eliminating all the edges whose weights are lower than σ. The Star algorithm approximate a clique cover of G_σ using denser star-shaped sub-graphs [1].

This algorithm has some drawbacks: (i) dependency on the data order processing, and (ii) production of "illogical" clusters, since two star centers are never adjacent. These drawbacks were properly explained in [6]. The Extended Star algorithm was proposed by Gil *et al.* to solve the aforementioned drawbacks [6]. They represent also the document collection by its thresholded similarity graph, defining a new notion of star center, obtaining as a consequence, different star-shaped clusters that are independent of data order.

Unlike the Star algorithm, the obtained clusters are independent of data order. Nevertheless, the Extended Star algorithm has also some drawbacks. First of all, it can leave uncovered vertices, producing an infinite loop. This situation is illustrated in Fig. 1 (A).

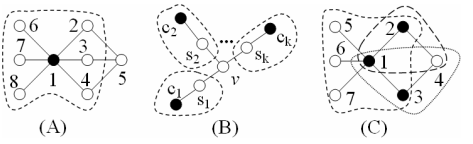

Fig. 1. Drawbacks of Extended algorithm

This situation is not an isolated case. We can generalize that any time that there is a vertex v – such as the illustrated in graph (B) of Fig. 1 – that satisfies the condition described in (1), then the algorithm produces an infinite loop, leaving the vertex v uncovered.

$$\forall s_i, 1 \leq i \leq k, |v.Adj| > |s_i.Adj| \wedge \forall c_i, 1 \leq i \leq k, |c_i.Adj| > |v.Adj| \ . \quad (1)$$

In this graph, each s_i represents the corresponding neighbours (adjacent vertices) of v, and c_i, is the adjacent center of s_i with highest degree. In (1) and in the following expressions, $x.Adj$ represents the set of adjacent vertices of the vertex x.

The second drawback of this algorithm is that it can produce unnecessary clusters, since more than one center can be selected at the same time. As can be noticed in graph (C) of Fig. 1, vertex 2 and vertex 3 should not be centers at the same time because we only need one of them to cover vertex 4.

A different version of the Extended Star algorithm was proposed by Gil *et al.* to construct a parallel approach [7]. This new version is also independent of data order, and solves the first drawback of the former Extended Star algorithm, but it can produce unnecessary clusters and illogical (less dense) clusters.

3 ACONS Algorithm

In this section we introduce a new concept of star allowing a different star-shaped form and as a consequence a new method, called ACONS, is obtained. As with the aforementioned algorithms, we represent the document collections by its thresholded similarity graph G_σ.

3.1 Some Basic Concepts

In order to define this new star concept and to describe the method, we define a finite sequence of directed graphs called transition graphs. Each new transition graph removes the unnecessary edges to get better clusters. Thus, the last transition will hold the vertices with real possibilities to be centers.

We call the *first transition* of $G_\sigma = \langle V, E_\sigma \rangle$ to the directed-graph $G_\sigma^{(0)} = \langle V, E_\sigma^{(0)} \rangle$ resulting from adding the directed-edge (v, u) to $E_\sigma^{(0)}$ iff the edge $(v, u) \in E_\sigma$.

Let $n \geq 0$ be an integer number, we call the *next transition* of $G_\sigma^{(n)} = \langle V, E_\sigma^{(n)} \rangle$, to the directed-graph $G_\sigma^{(n+1)} = \langle V, E_\sigma^{(n+1)} \rangle$, resulting from adding the directed--edge (v, u) to $E_\sigma^{(n+1)}$ iff $(v, u) \in E_\sigma^{(n)}$ and $v.out[n] \geq u.out[n]$, where $v.out[n]$

denote the *out-degree* of v in $G_\sigma^{(n)}$, i.e the number of edges $(v,x) \in E_\sigma^{(n)}$. It is important to notice that as $G_\sigma^{(n)}$ is not affected in the construction of $G_\sigma^{(n+1)}$, we can conclude that this process does not depend on data order.

Thus, starting from G_σ, we can construct a *sequence of graph transitions* $\{G_\sigma^{(0)}, G_\sigma^{(1)}, \ldots, G_\sigma^{(n)}, \ldots\}$. Furthermore, the integer positive sequence $\{e_n\}_{n=0}^{\infty}$, where $e_n = |E_\sigma^{(n)}|$, is decreasing and there is a unique integer $h \geq 0$ such that the finite sequence of terms $\{e_n\}_{n=0}^{h}$ is strictly decreasing and the sequence $\{e_n\}_{n=h}^{\infty}$ is constant. Then we say that $G_\sigma^{(h)}$ is the *last transition* of G_σ. Given $u, v \in V$, we say that u is an *r-satellite* of v, if $0 \leq r \leq h$ and $(v,u) \in E_\sigma^{(r)}$. We denote $v.Sats[r] = \{u \in V \mid u \text{ is an } r\text{-satellite of } v\}$ as the set of all r-satellites of v.

A *condensed star-shaped sub-graph* of $m+1$ vertices in G_σ, consists of a single center c and m adjacent vertices, such that $c.out[h] > 0$. Each isolated vertex in G_σ will be considered as a degenerated condensed star-shaped sub-graph with only one vertex.

Starting from this definition and guaranteing a full cover C of G_σ, this method should satisfy the following post-conditions:

$$\forall x \in V \ , \ x \in C \vee x.Adj \cap C \neq \emptyset \ , \tag{2}$$

$$\forall c \in C, \forall u \in c.Sats[h] \ , \ c.out[h] \geq u.out[h] \ . \tag{3}$$

The first condition (2) guarantees that each object of the collection belongs at least to one group, as a center or as a satellite. On the other hand, the condition (3) indicates that all the centers satisfy the condensed star-shaped sub-graph definition.

3.2 ACONS Algorithm

In order to define the ACONS algorithm, we introduce the concepts of voting-degree of a vertex and the redundancy of a center.

Let $G_\sigma^{(h)}$ be the last transition of G_σ and $v \in V$ a non-isolated vertex. The *voting-degree* ($v.vd$) of a vertex v is $v.vd = |\{u \mid v \in u.Electees\}|$, where $u.Electees = \arg\max_x \{x.out[h] \mid x \in u.Adj \cup \{u\}\}$.

Let C be a set of centers obtained by the algorithm, a center vertex c will be considered *redundant* if it satisfies the following conditions:

1. $\exists d \in c.Adj \cap C, d.out[h] > c.out[h]$, i.e. vertex c has at least one adjacent center (with greater out-degree) on its neighborhood.
2. $\forall s \in c.Sats[h], s \in C \vee |s.Adj \cap C| > 1$, i.e. vertex s has more than one adjacent center (a neighboring center different to c) on its neighborhood or vertex s is a center.

The logic of the ACONS algorithm is to generate a cover of G_σ by the densest condensed star-shaped sub-graphs. The centers are selected from a candidates list, formed by the vertices with positive voting-degree in the last transition of G_σ. The algorithm is summarized in Fig 2.

The functions *FindFirstTransition* and *FindLastTransition* are applied to construct the first and the last transition of G_σ based on the concepts and definitions

Algorithm 1: ACONS

Input: $V = \{d_1, d_2, \ldots, d_N\}$, σ-similarity threshold
Output: SC-Set of clusters

```
// Phase 1
G_σ^(0) := FindFirstTransition(V, σ);
G_σ^(h) := FindLastTransition(G_σ^(0));
forall vertex v ∈ V do v.Electees := arg max_x{x.out[h] | x ∈ v.Adj ∪ {v}};
forall vertex v ∈ V do v.vd := |{u | v ∈ u.Electees}|;
L := {v ∈ V | v.vd > 0};
// Phase 2
C := {v ∈ V | v.Adj ≠ ∅} ;
U := ∅;
while L ≠ ∅ do
    v := arg max_x{x.vd | x ∈ L} ;            // Only one vertex is selected
    if v.Adj ∩ C ≠ ∅ then C := C ∪ {v}
    else
        F = {u ∈ v.Sats[h] | u.Adj ∩ C ≠ ∅};
        if F ≠ ∅ then
            if ∃f ∈ F, v.out[h] > f.out[h] then C := C ∪ {v}
            else U := U ∪ {v};
        end
    end
    L := L \ {v};
end
// Phase 3
forall vertex v ∈ U do
    if ∃u ∈ v.Sats[h], u.Adj ∩ C ≠ ∅ then C := C ∪ {v};
end
// Phase 4
"Sort C in ascending order by out-degree";
SC := ∅;
forall center c ∈ C do
    if c is redundant then C := C \ {c}
    else SC := SC ∪ {{c} ∪ c.Adj};
end
```

Fig. 2. Pseudo-code of ACONS Algorithm

mentioned in section 3.1. Both functions are very easy to be implemented, because it is not necessary to preserve all transition states.

The algorithm is made up of five phases: (1) computes the last transition of G_σ, and calculates the candidates list L using voting-degrees, (2) determines centers list C and uncertain centers lists U from L, (3) processes U to find new centers, and (4) removes from C the redundant centers and constructs the set of clusters.

The phase (1) is very important, because it guarantees the selection of vertices that actually have real possibilities to be selected as center, i.e. vertices that could form a dense condensed star-shaped sub-graph. Notice that the starting candidates list L after phase (1) is made up of the vertices $v \in V$ with $v.vd > 0$. Thus, the vertices outside L are isolated or satellites with at least one adjacent vertex in L.

The isolated vertices are selected as centers at the beginning of the phase (2). Afterward, the vertices of L are processed in a decreasing order regarding the voting-degree; in this way, we ensure that any selected center will satisfy the post-condition (3). In each iteration, the vertex v is processed considering the following situations:

1. If v has not been covered yet by an adjacent vertex $c \in C$ then we add v to C; thus we try to reduce the overlapping among sub-graphs and ensure that v is covered at least by itself.
2. If v has some adjacent vertex f that has not been covered yet and satisfy:
   (a) If f has a lesser out-degree than v then we add v to C; thus we ensure that such vertex f will belong to a sub-graph denser than the one it can form.
   (b) Otherwise, v is added to uncertain list U postponing the selection of v as center.

At the end of each iteration, we remove the vertex v from L to guarantee the phase (2) ending.

During phase (3) all of the vertices $v \in U$ are processed in the insertion order, selecting v as center if it is needed to cover some adjacent vertex. Thus, each vertex s outside C has at least one adjacent vertex in C, i.e. the post-condition (2) is fulfilled. Finally (phase(4)), we check the redundancy of each vertex to eliminate the redundant centers in C .

3.3 General Considerations of ACONS Algorithm

The ACONS method – as the original Star algorithm and the two versions of the Extended algorithm – generates clusters which can be overlapped and guarantees also that the pairwise similarity between satellites vertices in a condensed star-shaped sub-graph be high.

As we can see in Fig. 3, unlike its previous algorithms, the ACONS algorithm can not produce illogical clusters because all the centers satisfy the condensed

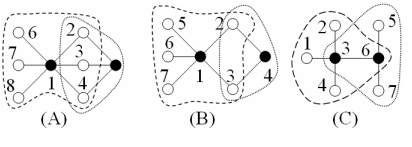

Fig. 3. Solutions to uncovered vertices (A), unnecessary clusters (B) and illogical clusters (C)

star-shaped sub-graph definition. The ACONS algorithm does not produce uncovered vertices – this property is ensured by the fulfillment of postcondition (2) – and avoid the generation of unnecessary clusters presented in graph (A) and (C) of Fig. 1 respectively.

The dependence on data order is a property that the Extended Star method certainly solves. Nevertheless, as we had previously indicated, it is necessary only when that dependence affects the quality of the resulting clusters. Thus, the ACONS algorithm solves the dependence on data order (for non symmetric or similar solutions) observed in the Star algorithm.

4 Experimental Results

In this section we present the experimental evaluation of our method, comparing its results against the Extended Star method and the original Star algorithms. The produced clustering results are evaluated by the same method and criterion to ensure a fair comparison across all algorithms.

Two data sets widely used in document clustering research were used in the experiments: TREC-5 and *Reuters*-21578. These are heterogeneous regarding document size, cluster size, number of classes, and document distribution. The data set TREC-5 contains news in Spanish published by AFP during 1994 (`http://trec.nist.gov`); *Reuters*-21578 was obtained from `http://kdd.ics.uci.edu` . We excluded from data sets the empty documents and also those documents do not have an associated topic.

In our experiments, the documents are represented using the traditional vector space model. The index terms of documents represent the lemmas of the words appearing in the texts. Stops words, such as articles, prepositions and adverbs are removed from document vectors. Terms are statistically weighted using the term frequency. We use the traditional cosine measure to compare the documents.

The literature abounds in measures defined by multiple authors to compare two partitions on the same set. The most widely used are: Jaccard index, and F-measure.

Jaccard index.- This index (noted j) takes into account the objects simultaneously joined [8]. It is defined as follows:

$$j(A,B) = \frac{n_{11}}{\frac{N(N-1)}{2} - n_{00}} \ . \tag{4}$$

In this index, n_{11} denotes the number of pairs of objects which are both in the same cluster in A and are also both in the same cluster in B. Similarly, n_{00} is the number of pairs of objects which are in different clusters in A and are also in different clusters in B.

The performances of the algorithms in the document collections considering Jaccard index are shown in Fig. 4 (A) and (B).

F-measure.- The aforementioned index and others are usually applied to partitions. In order to make a better evaluation of overlapping clustering, we have

considered F-measure calculated over pairs of points, as defined in [4]. Noted as *Fmeasure*, this measure is the harmonic mean of *Precision* and *Recall*:

$$Fmeasure = \frac{2 * Presicion * Recall}{Presicion + Recall} \quad , \tag{5}$$

where:

$$Presicion = \frac{n_{11}}{\text{Number of identified pairs}}, \quad Recall = \frac{n_{11}}{\text{Number of true pairs}} \quad .$$

The performances of the algorithms in the document collections considering F-measure are shown in Fig. 4 (C) and (D).

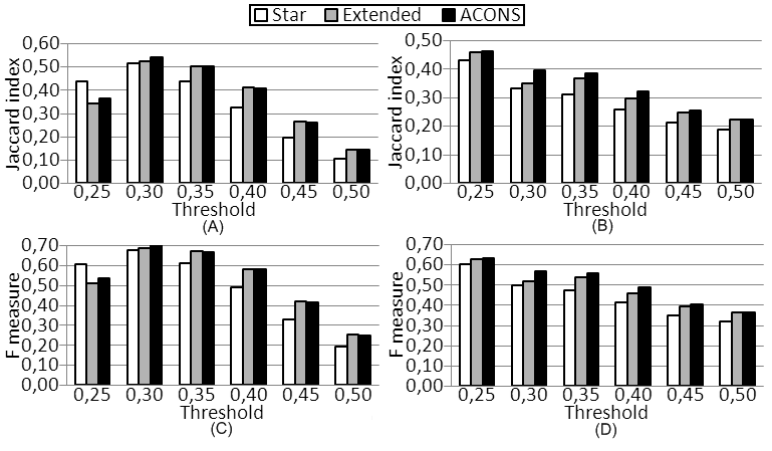

Fig. 4. Behavior in AFP (A,C) and Reuters (B,D) collections with Jaccard index and F-measure

As can be noticed, the accuracy obtained using our proposed algorithm is in most cases (for all the indexes) comparable with that obtained from the other methods investigated; moreover, our proposal can outperform those methods for all the indexes. But, this behavior is not homogeneous for all similarity thresholds; for each collection, there is a minimum value for which ACONS outperforms previous Star methods. Starting from this minimum value, the accuracy of ACONS is in general as good as, or even in many cases higher than, the others.

Furthermore, ACONS in all cases obtains lesser clusters than the other algorithms (see Fig. 5), and in most cases obtains denser clusters. This behavior could be of great importance for obtaining a minimum quantity of clusters without loosing precision.

It is important to notice that the Extended algorithm could cover all the vertices, but only in these experiments. Nevertheless, as it was explained, theoretically the Extended algorithm may fail with other repositories.

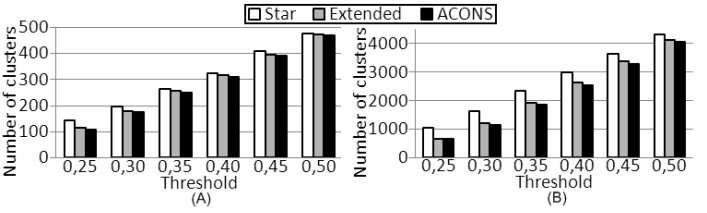

Fig. 5. Number of generated clusters in AFP (A) and Reuters (B) collections

Despite the experiments carried out by Aslam *et al.* in [1], and in order to ensure the effectiveness of our proposed algorithm, we made a new experimentation to compare the performance of ACONS algorithm against the Single Link and Average Link [5] algorithms, which uses different cost functions. For a fair comparison across all algorithms, we used the same thresholds of the previous experiments, stopping the execution of the Single Link and Average Link algorithms when the two selected clusters to be joined do not satisfy the current threshold, meaning that the evaluation of the cost function for all pair of clusters in the current algorithm return a value greater than the selected threshold. After that, we evaluated each algorithm considering the Jaccard index and F-measure, and we selected the average value of each algorithm for the selected measures for all thresholds.

The performances of the algorithms in the document collections considering Jaccard index, and F-measure are shown in Fig. 6.

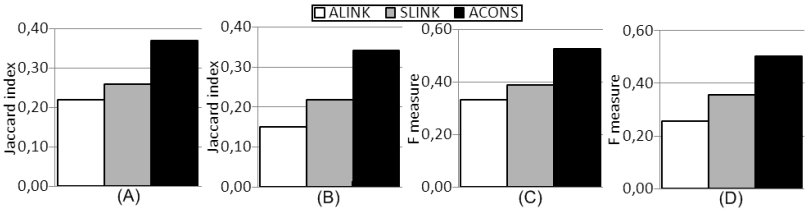

Fig. 6. Behavior in AFP (A,C) and Reuters (B,D) collections with Jaccard index and F-measure

As we can see, our proposal also outperforms the Single Link and Average Link algorithms in both collections. Thus, the ACONS algorithm represents a 68.2% improvement in performance compared to average link and a 42.3% improvement compared to single link in AFP collection considering the Jaccard index; if we consider F-measure, then the ACONS algorithm represents a 57.6% improvement in performance compared to average link and a 33.3 improvement compared to single link in the same collection. In the case of the Reuters collection the improvements are higher and even in some cases it doubles the result.

5 Conclusions

In this paper we presented a new clustering algorithm called Condensed Star (ACONS). As a consequence, we obtained different star-shaped clusters. This algorithm solves the drawbacks observed in Star and Extended Star methods: the dependence on data order (for non symmetric or similar solutions), the production of uncovered vertices and the generation of illogical and redundant clusters.

We compared the ACONS algorithm with the original Star and the Extended Star methods. The experimentation shows that our proposal outperforms previous methods for all the measures and aspects. These performances prove the validity of our algorithm for clustering tasks.

This algorithm can be used for organizing information systems, browsing, topic tracking and new topic detection. Although we employ our algorithm to cluster documents collections, its use is not restricted to this area, since it can be applied to any problem of pattern recognition where clustering is needed.

As a future work, we want to do some other experiments considering other representations of the documents and other similarity measures. These experiments could help us to decide a priori how to choose the threshold value in order to obtain the best performance of our algorithm.

References

1. Aslam, J., Pelekhov, K., Rus, D.: Static and Dynamic Information Organization with Star Clusters. In: Proceedings of the 1998 Conference on Information Knowledge Management, Baltimore (1998)
2. Aslam, J., Pelekhov, K., Rus, D.: Using Star Clusters for Filtering. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, USA (2000)
3. Aslam, J., Pelekhov, K., Rus, D.: The Star Clustering Algorithm for Static and Dynamic Information Organization. Journal of Graph Algorithms and Applications 8(1), 95–129 (2004)
4. Banerjee, A., Krumpelman, C., Basu, S., Mooney, R., Ghosh, J.: Model Based Overlapping Clustering. In: Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD) (2005)
5. Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley & Sons, Chichester (2001)
6. Gil-García, R.J., Badía-Contelles, J.M., Pons-Porrata, A.: Extended Star Clustering Algorithm. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 480–487. Springer, Heidelberg (2003)
7. Gil-García, R.J., Badía-Contelles, J.M., Pons-Porrata, A.: Parallel Algorithm for Extended Star Clustering. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, p. 402. Springer, Heidelberg (2004)
8. Kuncheva, L., Hadjitodorov, S.: Using Diversity in Cluster Ensembles. In: Proceedings of IEEE SMC 2004, The Netherlands (2004)
9. van Rijsbergen, C.J.: Information Retrieval, Buttersworth, London, 2nd edn. (1979)
10. Zhong, S., Ghosh, J.: A Comparative Study of Generative Models for Document Clustering. In: Proceedings of SDM Workshop on Clustering High Dimensional Data and Its Applications (2003)