Experimental analysis of impact of term weighting schemes on cluster quality

G. Hannah Grace* and Kalyani Desikan

Department of Mathematics,
School of Advanced Sciences,
VIT University,
Chennai, India
Email: hannahgrace.g@vit.ac.in
Email: kalyanidesikan@vit.ac.in
*Corresponding author

Abstract: Term weighting schemes are used to identify the importance of each term with respect to a collection and assign weights to them accordingly. Document clustering uses these term weights to compare the similarity between documents. Several term weighting schemes are in use today, but none of them are specific to the clustering algorithms. The term frequency-based clustering techniques consider the documents as a bag of words while ignoring the relationship between the words. So, in this paper we focus our analysis on different term weighting schemes such as term frequency (tf), term frequency-inverse document frequency (tfidf), automatic text categorisation (ATC) without normalisation and ATC-inverse document frequency (ATCidf). In this paper, we have used the clustering tool CLUTO to experimentally study the impact of term weighting schemes on the quality of the clustering solution obtained by applying the Repeated Bisection Partitional Algorithm.

Keywords: document clustering; term weighting scheme; cluster quality; criterion functions; entropy; purity.


Biographical notes: G. Hannah Grace is currently an Assistant Professor in the Department of Mathematics, School of Advanced Sciences, VIT University, Chennai. She has over eight years of academic experience. She is doing her research under the guidance of Dr. Kalyani Desikan, Professor, School of Advanced Sciences, VIT Chennai in the area of text clustering. She has five publications to her credit.

Kalyani Desikan is currently a Professor in the Department of Mathematics, School of Advanced Sciences, VIT, Chennai. She has over 20 years of experience in academics and software industry. She has 20 journal publications in both international and Indian journals of repute. She is currently guiding five research students in the areas of text clustering, link analysis, categorical data clustering, automata theory and spectral graph theory.

Copyright © 2018 Inderscience Enterprises Ltd.
1 Introduction

Document clustering is an unsupervised technique in which users do not provide any information about the given document to the clustering algorithm. It needs to find the hidden information in the document by itself (Murugesan and Zhang, 2011). Many algorithms are available in the literature for performing document clustering (Han and Kamber, 2009; Jain et al., 1999; Steinbach et al., 2000; Xu, 2005). Out of these, two major categories of algorithms, partitioning algorithms and hierarchical algorithms are commonly used for document clustering (Bai et al., 2010). Partitioning clustering algorithm divides the collection of documents into partitions, where each partition represents a cluster. Hierarchical clustering creates a hierarchical decomposition of the given document collection thus achieving a hierarchical structure. Partitioning algorithms generally perform better than hierarchical algorithms (Zhao and Karypis, 2002b; Steinbach et al., 2002; Latika, 2015).

Partitional clustering methods find clusters by optimising a certain objective function that defines the optimal solution (Hartigan, 1975; Huang, 2008; Zhao and Karypis, 2002a). A partitional clustering algorithm finds all the non-overlapping clusters at once by dividing the set of documents based on an objective function. These algorithms try to minimise or maximise an objective function. The partitional clustering algorithms are well suited for clustering large document datasets due to their relatively low computational requirements (Charu et al., 1999; Bai et al., 2010; van de Cruys, 2010). For example, the k-means algorithm minimises the squared error in the resulting cluster structure, by assigning each point to its closest cluster in each iteration. Noting that exhaustive search through all possible partitions for the optimal solution is computationally prohibitive (Hartigan, 1975). It is common to approximate this by running the algorithms multiple times with different initialisation, each time generating a different partition of the dataset, and then use the best clustering result.

The most widely used partitioning algorithm is the k-means algorithm. Although k-means algorithm is simple, it is quite sensitive to the selection of initial cluster centres (centroids) (Prabhu and Anabazhagan, 2011; Latika, 2015), i.e., the results of clustering greatly depend on the position of the initial cluster centres which are randomly chosen in the standard k-means algorithm. Several methods have been proposed to improve the results of the standard k-means algorithm.

Given a document collection, we first pre-process the documents using techniques like tokenisation, removal of white spaces, removal of punctuation marks, conversion of all alphabets to lower case, removal of numerals, removal of hyperlinks, stop words removal and stemming (Grace and Desikan, 2015; Gowtham et al., 2014; da Cruz Nassif and Hruschka, 2013). Documents can be clustered only after they are represented in a format that can be given as input to the clustering algorithm. One such representation is the term document matrix. The entire collection of documents can be viewed as a term document matrix. The term document matrix is a document collection represented as a matrix where each row corresponds to a document in the collection and the column vectors are the term vectors of the document. Term Frequency is the frequency of occurrence of a term in a document. It shows how important a document is in the collection.
Term weighting is related to how a term appears in a document or a collection of documents (Plansangket and Gan, 2015). The effect of the term weighting schemes on partitional clustering algorithms shows that different term weighting schemes substantially lead to different results. Hence, we focus our attention on the effect of different term weighting schemes such as tf, tfidf, ATC without normalisation and ATCidf without normalisation on two different document datasets.

The main focus of this paper is to experimentally study the impact of term document scaling/weighting schemes. The clustering tool we have used for our experiment is CLUTO (Zhao and Karypis, 2002c; Zhao and Karypis, 2002a; Karypis et al., 2002). CLUTO provides tools for analysing the discovered clusters to understand the relations between the objects assigned to each cluster and the relationship between different clusters, and tools for visualising the discovered clustering solutions. CLUTO can identify the features that best describe and discriminate each cluster. These set of features can be used to gain a better understanding of the set of objects assigned to each cluster and to provide concise summaries about the cluster’s contents. CLUTO is used for clustering low and high dimensional datasets and for analysing the characteristics of the various clusters. CLUTO can be used to cluster documents based on similarity or distance measures like cosine similarity, correlation coefficient, Euclidean distance and extended Jaccard coefficient. CLUTO also provides cluster quality details such as entropy and purity.

We have used the I2 criterion function available in CLUTO, in the context of the partitional approach, namely the repeated bisection clustering algorithm to cluster the documents. We have experimentally analysed cluster quality, based on Entropy and Purity, for two different document sets. We have chosen the I2 criterion condition because in our paper (Grace and Desikan, 2014; Karypis et al., 2002) we have experimentally proved that I2 criterion condition performs the best with respect to clustering time.

Our paper is organised as follows. Section 2 introduces the preliminaries such as term weighting schemes with descriptions for tf, tfidf, ATC without normalisation and ATCidf weighting schemes, clustering solution using partitional clustering algorithm and cluster quality measures. Section 3 describes the document sets used and the experimental methodology. In Section 4, we present the results and analysis and finally in Section 5, we present the conclusion.

2 Preliminaries

2.1 Term weighting schemes

Term weighting schemes identify the importance of each term with respect to a document collection and assign weights to them accordingly. Term weighting schemes can be classified as supervised and unsupervised approaches (van de Cruys, 2010). Most of the term weighting schemes are variants of the tfidf weighting scheme which belongs to the class of unsupervised weighting schemes. Document clustering uses these term weights to compare the similarity between two documents. Several term weighting schemes are in use today, but none of them are specific to clustering algorithms.

A wide variety of approaches are adopted to assign and assess the importance of terms and assign weights to them. Some employ genetic algorithm for assigning weights
to terms (Robertson et al., 1986), Keen (1991) use a scheme concept for weighting. Some papers introduce a modelling method based on the sources of documents to determine term importance (Wilbur, 1993). In a few papers like Boger et al. (2001), artificial neural networks is employed, Gordon and Dumais (1998) uses latent semantic technique in indexing and some apply probability theory to solve the same problem (Greiff, 1998; Melucci, 1998; Ponte and Croft, 1998; Robertson and Willett, 1996; van Rijsbergen, 1997). Clearly, term significance measure in a full text can be influenced by its frequency, type of document, its content in the document, its function, its position in the document and other factors (Zhang and Nguyen, 2005). Hence, term frequency (tf) is mostly applied to determine its significance in automatic information processing.

Term weighting (Murty et al., 2011) is a numerical representation of a collection of documents comprising of ‘N’ documents and ‘p’ terms from a document corpus which is represented as a matrix as given below

\[
X = \begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1p} \\
    x_{21} & x_{22} & \cdots & x_{2p} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{N1} & \cdots & \cdots & x_{Np}
\end{bmatrix}
\]

This is a sparse matrix called the term-document matrix whose rows correspond to documents and columns correspond to the stemmed terms appearing in the documents. Stemmed terms are those which are obtained at the end of the document preprocessing stage. Preprocessing stage includes tokenisation, stop words elimination and stemming.

There are three different factors that determine the term weighting schemes (Chisholm and Kolda, 1999): local, global and normalisation.

The local weight of a certain term varies depending on the document that contains the term. Local weights are functions of how many times each term appears in a document. Local weighting schemes perform well and they are based on the principle that terms with higher within-document frequency are more pertinent to that document (Murty et al., 2011). Local weight is also known as the term frequency component of the weighting scheme.

Global term weighting schemes use global document information such as document frequency and total term frequency across documents. The global weight of a certain term is fixed, that is, it is independent of the document in which the term occurs. Global weights are functions of how many times each term appears in the entire document collection. Global weighting assigns a ‘discrimination value’ to each term. Many schemes are based on the idea that the less frequently a term appears in the whole collection, the more discriminating it is (Premalatha and Natarajan, 2010). Global weight is also known as the collection frequency component of the weighting scheme. A commonly used global weight is the inverse document frequency measure, or IDF.

The normalisation factor compensates for discrepancies in the lengths of the documents in the document collection. Term weighting normalisation has been one of the main issues in information retrieval (IR) for many years (He and Ounis, 2003).
Various weighting schemes are available; to list a few we have tf, tfidf, okapi, ATC, MI, tficf, etc. (Zaragoza et al., 2004; Dinçer and Karaoğlan, 2004). For our experimental analysis we have considered four weighting schemes viz. tf, tfidf, ATC without normalisation and ATCidf.

2.1.1 TF weighting scheme

Term frequency (TF) is the simplest measure to assign weights to each term in a document. Term frequency (TF) measures how frequently a term occurs in a document (Salton and Buckley, 1988). In this weighting scheme, a term document matrix comprises of rows corresponding to documents in the collection and columns corresponding to terms. The elements of this matrix show which document contains which term and how many times the term appears in each of the documents.

2.1.2 TFIDF weighting scheme

TFIDF is the term frequency – inverse document frequency weighting scheme which is commonly used. This reflects how important a word is to a document in a collection or corpus (Jin et al., 2001). The tfidf value increases proportional to the number of times a word appears in the document using the frequency of the word in the corpus. It assigns weight to terms in the documents in proportion to the number of occurrences of the term in the document and in inverse proportion to the number of documents in the collection in which the terms occurs at least once (Mutchima and Sanguansat, 2012).

The basic form of tfidf (Premalatha and Natarajan, 2010) is given by

$$w_{ij} = x_{ij} \log \frac{N}{n_j}$$

where $w_{ij}$ is the weight of term $j$ in document $i$; $x_{ij}$ is the number of occurrences of term $j$ in document $i$ (TF); $N$ is the total number of documents in the document collection; and $n_j$ is the number of documents in which term $j$ occurs at least once. $\frac{N}{n_j}$ is often referred to as the document frequency (DF) of term $j$ and naturally, $\frac{n_j}{N}$ is called the inverse document frequency (IDF) of term $j$.

IDF is the most frequently used collection frequency factor. Logarithm is used to adjust within-document frequency because a term that appears ten times in a document is not necessarily ten times more important than a term that appears once in that document. It reduces the effect of multiple occurrences of terms. This helps to adjust the weight assigned to some words that appear more frequently.

2.1.3 ATC weighting scheme (without normalisation)

ATC (Saad et al., 2006; Reed et al., 2006) stands for augmented tf weighting scheme. The ATC weighting scheme (without normalisation) is given by (Jin et al., 2001; Buckley et al., 1996):
Experimental analysis of impact of term weighting schemes on cluster quality

\[ w_{ij} = 0.5 + 0.5 \left( \frac{x_{ij}}{\max_i(x_{ij})} \right) \]

where \( x_{ij} \) is the term frequency, \( \max_i(x_{ij}) \) is the maximum term frequency in each document.

### 2.1.4 ATCidf weighting scheme (proposed weighting scheme)

We have proposed a new weighting scheme which is a combination of ATC (without normalisation) and idf. We have also shown experimentally that this scheme works well for the benchmark document set ‘classic’. The formula for ATCidf is given by

\[ w_{ij} = \left[ 0.5 + 0.5 \left( \frac{x_{ij}}{\max_i(x_{ij})} \right) \right] \times \log \left( \frac{N}{n_j} \right) \]

where \( x_{ij} \) is the term frequency, \( \max_i(x_{ij}) \) is the maximum term frequency in each document, \( N \) is the total number of documents in the collection, \( n_j \) is the number of documents in which term \( j \) occurs at least once.

### 2.2 Weighting schemes in CLUTO

In CLUTO (Zhao and Karypis, 2002c), the term weighting schemes are referred to as Scaling schemes. They are primarily used to smooth large values in the term document matrix. In CLUTO scaling of terms can be achieved by applying what are known as row models and column models to the term document matrix. The choice of the options for both row model and column model are motivated by the clustering requirement of high dimensional datasets arising in document datasets.

Row model refers to the weighting scheme to be used to scale elements of each row of the term document matrix. The possible weighting schemes that can be applied include none, maxtf, sqrt and log. When you choose ‘none’, the values of the term document matrix are used as is. This is the default setting in CLUTO. If you choose maxtf, the values in each row are scaled to lie between 0.5 and 1.0. In the sqrt model, the values of each row are scaled to be equal to the square root of their actual values. In the log model, the values of each row are scaled to be equal to the log of their actual values.

The column model scales the various columns of the term document matrix globally across all the rows. The possible options include none and idf. The default setting is ‘idf’ where the columns of the matrix are scaled according to the inverse document frequency paradigm.

Tables 1 and 2 give the descriptions for the different row and column scaling schemes available in CLUTO.
Table 1  
Row scaling schemes

<table>
<thead>
<tr>
<th>Row model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Columns of each row are not scaled and used as they are provided in the input file.</td>
</tr>
<tr>
<td>maxtf</td>
<td>Columns of each row are scaled so that their values are between 0.5 and 1. In particular, the $j^{th}$ column of the $i^{th}$ row of the matrix $w_{ij}$ is scaled to be equal to</td>
</tr>
</tbody>
</table>

$$w'_{ij} = 0.5 + 0.5 \left( \frac{w_{ij}}{\max_i \{w_{ij}\}} \right)$$

This scaling is motivated by a similar scaling of document vectors in information retrieval.

| sqrt      | Columns of each row are scaled to be equal to the square root of their actual values. |

$$w'_{ij} = \text{sign}(w_{ij}) \sqrt{|w_{ij}|}$$

where $\text{sign}(w_{ij})$ is 1 or -1 depending on whether or not $w_{ij}$ is positive or negative.

| log       | Columns of each row are scaled to be equal to the log of their actual values. |

$$w'_{ij} = \text{sign}(w_{ij}) \log_2 |w_{ij}|$$

Table 2  
Column scaling schemes

<table>
<thead>
<tr>
<th>Column model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Columns of the matrix are not globally scaled, and they are used as is.</td>
</tr>
<tr>
<td>idf</td>
<td>Columns of the matrix are scaled according to the inverse-document frequency (idf) paradigm. In particular, if $rf_i$ is the number of rows that the $i^{th}$ column belongs to then each entry of the $i^{th}$ column is scaled by $-\log_2 \left( \frac{rf_i}{n} \right)$. The effect of this scaling is to deemphasise column entries that appear in many rows.</td>
</tr>
</tbody>
</table>

Table 3 gives the mapping between the various row and column models combinations available in CLUTO and the term weighting schemes.

Table 3  
Scaling models in CLUTO and equivalent term weighting schemes

<table>
<thead>
<tr>
<th>Column model</th>
<th>Row model</th>
<th>Equivalent term weighting schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>None</td>
<td>Term frequency (tf)</td>
</tr>
<tr>
<td>idf</td>
<td>None</td>
<td>Term frequency-inverse document frequency (tfidf)</td>
</tr>
<tr>
<td>None</td>
<td>maxtf</td>
<td>ATC without normalisation</td>
</tr>
<tr>
<td>idf</td>
<td>maxtf</td>
<td>A combination of ATC without normalisation (ATCidf) and inverse document frequency</td>
</tr>
</tbody>
</table>

2.3  
Clustering solution using partitional clustering algorithm:

The partitioning-based cluster algorithms (Grace and Desikan, 2015; Zhao and Karypis, 2002c) minimise a given clustering criterion by iteratively reallocating data points between clusters until an optimal partition is attained. The partitional techniques produce clusters by optimising a criterion function defined either locally or globally.
Experimental analysis of impact of term weighting schemes on cluster quality

2.3.1 Criterion functions

The main role of different clustering criterion functions is to determine which cluster to bisect next in the repeated bisection partition method (Saad et al., 2006). Seven criterion conditions \( I_1, I_2, E_1, G_1, G_1p, H_1 \) and \( H_2 \) were introduced by George Karypis (Zhao and Karypis, 2002c; Aggarwal et al., 1999). Theoretical analysis of the criterion functions shows that their relative performance depends on the:

1. degree to which they can correctly operate when the clusters are of different tightness
2. degree to which they can lead to reasonably balanced clusters.

In our experimental study, we have restricted our analysis to the \( I_2 \) criterion function. \( I_2 \) criterion function maximises the intra-cluster similarity between the elements of a cluster. It is given by the formula

\[
\text{maximise } \sum_{i=1}^{k} \sqrt{\sum_{u,v \in S_i} \text{sim}(u,v)}
\]

\( \text{sim}(u, v) \) in the formula refers to the similarity/distance between the document vectors \( u \) and \( v \). \( k \) is the number of clusters and \( S_i \) is the subset of a collection of documents. Any similarity/distance measure such as cosine similarity, Euclidean distance and so on can be used.

2.3.2 Cluster quality

Clustering quality measure is defined (Saad et al., 2006) as a function that maps pairs of the form (dataset, clustering) to some ordered set (say, the set of non-negative real numbers), so that these values reflect how ‘good’ or ‘cogent’ that clustering is. The quality of the clusters produced is measured using two external measures (Huang, 2008): entropy and purity.

Entropy measures how various classes of documents are distributed within each cluster. This provides the measure of ‘goodness’ for clusters that are not nested (Bai et al., 2010). The smaller the entropy values, better the clustering solution. Given a particular cluster \( S_r \) of size \( n_r \), the entropy of this cluster (Huang, 2008) is defined as

\[
E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_i}{n_r} \log \left( \frac{n_i}{n_r} \right)
\]

where \( q \) is the number of classes in the dataset and \( n_i \) is the number of documents of the \( i \)th class that are assigned to the \( r \)th cluster. The entropy of the entire clustering is the sum of the individual cluster entropies weighted according to the cluster size. For \( k \) clusters, we have

\[
\text{Entropy} = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r).
\]

Smaller entropy values indicate better clustering solutions.
Using the same mathematical notation, the purity of a cluster (Murty et al., 2011) is defined as,

$$Pu(S_r) = \frac{1}{n_r} \max_n n_r'$$

The purity of the clustering solution is the weighted sum of the individual cluster purities,

$$\text{Purity} = \sum_{r=1}^{k} \frac{n_r}{n} Pu(S_r)$$

Larger purity value indicates better clustering solution.

3 Document datasets and experimental methodology

In this paper we have considered two document datasets: classic dataset and hi-tech dataset whose general characteristics are summarised in Table 4.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of documents</th>
<th>Number of terms</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic</td>
<td>7,094</td>
<td>41,681</td>
<td>4</td>
</tr>
<tr>
<td>Hi-tech</td>
<td>2,301</td>
<td>126,373</td>
<td>6</td>
</tr>
</tbody>
</table>

3.1 Work flow of our study
4 Results and analysis

We have analysed our results for classic dataset and hi-tech dataset whose descriptions are given in Table 4. We have calculated the entropy and purity for the clustering solutions obtained by varying the cluster sizes from 5 to 50 using CLUTO for the various row models and column models pertaining to tf, tfidf, ATC without normalisation and a combination of idf with ATC. The entropy and purity values for classic dataset are given in Tables 5 and 6, respectively. We have analysed the quality of the clustering solution by varying the cluster size.

To analyse the effect of the row and column scaling schemes on the quality of clusters for the I2 criterion function we have presented the entropy and purity values for 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 number of clusters graphically in Figures 1–4 for both classic dataset and hi-tech dataset.

<table>
<thead>
<tr>
<th>Table 5</th>
<th>Entropy values for classic dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>k/Row model</td>
<td>Column model = none</td>
</tr>
<tr>
<td></td>
<td>Column model = none</td>
</tr>
<tr>
<td></td>
<td>maxtf</td>
</tr>
<tr>
<td>5</td>
<td>0.371</td>
</tr>
<tr>
<td>10</td>
<td>0.289</td>
</tr>
<tr>
<td>15</td>
<td>0.26</td>
</tr>
<tr>
<td>20</td>
<td>0.253</td>
</tr>
<tr>
<td>25</td>
<td>0.237</td>
</tr>
<tr>
<td>30</td>
<td>0.226</td>
</tr>
<tr>
<td>35</td>
<td>0.225</td>
</tr>
<tr>
<td>40</td>
<td>0.222</td>
</tr>
<tr>
<td>45</td>
<td>0.216</td>
</tr>
<tr>
<td>50</td>
<td>0.213</td>
</tr>
</tbody>
</table>

In Table 5, we note that the lowest entropy value is 0.139 which corresponds to row model ‘maxtf’ and column model ‘idf’ for 45 clusters. This corresponds to the ATCidf term weighting scheme as given in Table 3.

**Figure 1** Entropy for classic dataset (see online version for colours)
Table 6  Purity values for classic dataset

<table>
<thead>
<tr>
<th>k/Row model</th>
<th>Column model = none</th>
<th>Column model = idf</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxtf</td>
<td>none</td>
<td>maxtf</td>
</tr>
<tr>
<td>5</td>
<td>0.781</td>
<td>0.849</td>
</tr>
<tr>
<td>10</td>
<td>0.849</td>
<td>0.935</td>
</tr>
<tr>
<td>15</td>
<td>0.882</td>
<td>0.935</td>
</tr>
<tr>
<td>20</td>
<td>0.882</td>
<td>0.935</td>
</tr>
<tr>
<td>25</td>
<td>0.904</td>
<td>0.935</td>
</tr>
<tr>
<td>30</td>
<td>0.904</td>
<td>0.95</td>
</tr>
<tr>
<td>35</td>
<td>0.904</td>
<td>0.95</td>
</tr>
<tr>
<td>40</td>
<td>0.905</td>
<td>0.95</td>
</tr>
<tr>
<td>45</td>
<td>0.905</td>
<td>0.95</td>
</tr>
<tr>
<td>50</td>
<td>0.905</td>
<td>0.95</td>
</tr>
</tbody>
</table>

In Table 6, we see that the highest purity value is 0.95 which corresponds to row model ‘maxtf’ and column model ‘idf’ which is equivalent to ATCidf as mentioned in Table 3.

Figure 2  Purity for classic dataset (see online version for colours)

Table 7  Entropy values for hi-tech dataset

<table>
<thead>
<tr>
<th>k/Row model</th>
<th>column model = none</th>
<th>column model = idf</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxtf</td>
<td>none</td>
<td>maxtf</td>
</tr>
<tr>
<td>5</td>
<td>0.673</td>
<td>0.677</td>
</tr>
<tr>
<td>10</td>
<td>0.625</td>
<td>0.644</td>
</tr>
<tr>
<td>15</td>
<td>0.592</td>
<td>0.625</td>
</tr>
<tr>
<td>20</td>
<td>0.563</td>
<td>0.595</td>
</tr>
<tr>
<td>25</td>
<td>0.553</td>
<td>0.583</td>
</tr>
<tr>
<td>30</td>
<td>0.547</td>
<td>0.564</td>
</tr>
<tr>
<td>35</td>
<td>0.533</td>
<td>0.553</td>
</tr>
</tbody>
</table>

Figure 3  Entropy for hi-tech dataset (see online version for colours)
Experimental analysis of impact of term weighting schemes on cluster quality

Table 7  Entropy values for hi-tech dataset (continued)

<table>
<thead>
<tr>
<th>k/Row model</th>
<th>column model = none</th>
<th>column model = idf</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>maxtf none</td>
<td>maxtf none</td>
</tr>
<tr>
<td>40</td>
<td>0.529</td>
<td>0.546</td>
</tr>
<tr>
<td>45</td>
<td>0.52</td>
<td>0.535</td>
</tr>
<tr>
<td>50</td>
<td>0.511</td>
<td>0.525</td>
</tr>
</tbody>
</table>

For hi-tech dataset, the minimum value of entropy is 0.474 for row scaling scheme ‘none’ and column scaling scheme ‘none’ which corresponds to tf weighting scheme as given in Table 3.

Figure 3  Entropy for hi-tech dataset (see online version for colours)

Table 8  Purity values for hi-tech dataset

<table>
<thead>
<tr>
<th>k/Row model</th>
<th>Column model = none</th>
<th>Column model = idf</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>maxtf none</td>
<td>maxtf none</td>
</tr>
<tr>
<td>5</td>
<td>0.517</td>
<td>0.549</td>
</tr>
<tr>
<td>10</td>
<td>0.585</td>
<td>0.575</td>
</tr>
<tr>
<td>15</td>
<td>0.615</td>
<td>0.582</td>
</tr>
<tr>
<td>20</td>
<td>0.621</td>
<td>0.598</td>
</tr>
<tr>
<td>25</td>
<td>0.625</td>
<td>0.601</td>
</tr>
<tr>
<td>30</td>
<td>0.625</td>
<td>0.603</td>
</tr>
<tr>
<td>35</td>
<td>0.637</td>
<td>0.611</td>
</tr>
<tr>
<td>40</td>
<td>0.638</td>
<td>0.617</td>
</tr>
<tr>
<td>45</td>
<td>0.642</td>
<td>0.623</td>
</tr>
<tr>
<td>50</td>
<td>0.645</td>
<td>0.624</td>
</tr>
</tbody>
</table>

The maximum purity value 0.67 for hi-tech dataset clearly shows that it corresponds to tf weighting scheme.
From the graphs in Figures 1 and 2 for classic dataset it is clear that using a combination of ATC without normalisation and inverse document frequency (our proposed term weighting scheme) gives better quality clusters. But for hi-tech dataset from Figures 3 and 4 we notice that tf weighting scheme gives clusters of good quality.

5 Conclusions

For classic dataset, it can be seen from Tables 5 and 6 that when the number of clusters is 5 the clustering solution is good when we take the column model to be ‘idf’ and row model as ‘none’. From Table 3, we notice that this corresponds to tfidf term weighting scheme.

Subsequently, when we increase the number of clusters, we notice that the clustering solution is the best when we take the column model to be ‘idf’ and row model as ‘maxtf’. When we apply this scaling scheme the quality of the clustering solution is vastly improved. From Table 3, we see that this corresponds to the term weighting scheme which is a combination of ATC without normalisation and inverse document frequency (ATCidf).

Through experimental analysis we have shown that for the classic dataset the ideal term weighting scheme is a combination of ATC without normalisation and inverse document frequency.

For the hi-tech dataset from Tables 7 and 8, it is clear that when we apply the tf weighting scheme, i.e., when we take row model as ‘none’ and column model as ‘none’, the clustering quality is good. Hence, we conclude that for hi-tech dataset, the best term weighting scheme is tf.

We can clearly see that the term weighting schemes impact the quality of the clustering solution. Hence, we need to choose and apply the term weighting scheme appropriate for a dataset before we run the clustering algorithm to cluster the data points. This would ensure good quality clusters.

As a future work we would work on new term weighting schemes and identify a term weighting scheme that would be apt for a wide variety of document collections.
Experimental analysis of impact of term weighting schemes on cluster quality

References


