Clustering Scientific Literature Using Sparse Citation Graph Analysis

Levent Bolelli[1], Seyda Ertekin[1], and C. Lee Giles[1,2]

[1] Department of Computer Science and Engineering,
The Pennsylvania State University,
University Park, PA, 16802, USA
{bolelli, sertekin}@cse.psu.edu
[2] College of Information Sciences and Technology,
The Pennsylvania State University,
University Park, PA, 16802, USA
giles@ist.psu.edu

Abstract. It is well known that connectivity analysis of linked documents provides significant information about the structure of the document space for unsupervised learning tasks. However, the ability to identify distinct clusters of documents based on link graph analysis is proportional to the density of the graph and depends on the availability of the linking and/or linked documents in the collection. In this paper, we present an information theoretic approach towards measuring the significance of individual words based on the underlying link structure of the document collection. This enables us to generate a non-uniform weight distribution of the feature space which is used to augment the original corpus-based document similarities. The experimental results on the collection of scientific literature show that our method achieves better separation of distinct groups of documents, yielding improved clustering solutions.

1 Introduction

Document clustering refers to the task of extracting latent groupings in text databases. In broad terms, clustering is an optimization problem that attempts to find a partition of the document collection such that the items belonging to the same cluster are as similar as possible (cluster compactness) and the discovered clusters as separate as possible (cluster distinctness) based on a specified (dis)similarity metric within the high dimensional space that the document objects exist. In document collections where the only measure of similarity is textual content of documents, the traditional approach is the identification of meaningful features from documents and selection of the subset of features from the text corpus that yield better separation of distinct groups of documents. The clustering algorithm is then applied to this lower dimensional input space to discover distinct clusters.

The rapidly growing world wide web and the increasing volume of scientific literature available in digital format on the web has stimulated supervised and unsupervised data mining research to focus on linked documents. For linked documents, in addition to the textual content similarity, which can be thought of as an *implicit similarity*, we now have the link graph of the documents that depicts the *relatedness information*

J. Fürnkranz, T. Scheffer, and M. Spiliopoulou (Eds.): PKDD 2006, LNAI 4213, pp. 30–41, 2006.

conveyed by the authors of the digital content. Conventional clustering algorithms use attribute information to group documents under the assumption that two documents are related to each other if they have similar attribute values. However, relational data are richer in structure, hence provide more information available to disambiguate groupings. Therefore, link structure analysis has been studied extensively and has shown to be a significant aid for both supervised and unsupervised data analysis tasks. In this paper, we focus on clustering in the collection of scientific literature to discover topical groupings of papers using the textual content of papers combined with the citation graph of the collection. In a citation graph, papers are represented as vertices of the graph and citations as directed edges between citing and cited documents. The papers and citation graph have been obtained from CiteSeer's[1] repository.

CiteSeer [6] is a scientific literature digital library that has grown to index over 740.000 academic publications in Computer Science and related fields. Citations of the papers are extracted and linked to cited papers by Autonomous Citation Indexing [16]. The citation graph that is constructed through this process provides wealth of information since citations in research publications represent an important knowledge source regarding the context of scientific work. The citation relationships have been shown to be a valuable resource for a number of tasks such as ranking search results, identification of related research documents, trend analysis and social network analysis. Besides topical relevance, there have been identified multiple factors influencing citations, including the desire to publicize own research [10] and promoting own field, author's ability to access the document [15] and to read the language that it is written in [24]. Regardless of the reason for citations, comparatively, citation relationships between scientific documents convey a more valuable information than a collection of linked web documents. However, the citation graph itself can have limited clustering performance in digital libraries due to the following issues:

1) *Cited Document Availability*. CiteSeer collects the papers by crawling the web. Thus, the citations of a paper (i.e. target papers) may not be locally available in CiteSeer's repository due to several reasons: a) the citations may not be available on the web, b) they may just not have been crawled, or c) they may not be related to Computer Science or a similar field and may not be indexed. If any of these cases is true, the citations point to virtual metadata records that is identified by the extracted fields of the citation, including title, authors, publication venue, etc. However, the unavailability of the textual content of the cited papers prevents detailed analysis on the semantic similarity between the citing and cited papers.
2) *Identity Uncertainty*. Citations are references to unique documents, but their representations may vary, and finding the best matches for citations is a problem known as identity uncertainty [19]. The task of ACI is to uncover the identity of the paper that a citation refers to in order to group together similar citations to the same document, and to link citations to real documents – those that exist inside the ACI system and those that are yet to be crawled. Although ACI has been highly effective, it is still possible that distinct representations of the same citation may be mapped to different documents, or two citations to different papers be linked to the same target paper.

[1] http://citeseer.ist.psu.edu

The aforementioned reasons lead us to use only the citations where both the cited and citing documents are available in the collection, which sparsifies the link graph significantly. In this paper, we show that taking an information theoretic approach towards textual content analysis of pairs of documents with citation relationships provides a significant improvement in the discovery of document clusters. Further, we believe that the methodology presented here is applicable to web document collections where similar link constraints can be observed. One example is hierarchical clustering of documents where lower level taxonomies may not exhibit strong connectivity. Another application domain is search engine result clustering [9], an often employed technique to facilitate users' quick browsing through search results. Both applications suffer from the lack of sufficient links between the documents in a given subspace of the entire collection, which can be addressed by the algorithm proposed here.

2 Related Work

Document Clustering algorithms can be broadly categorized as text-based [20,2,21], link-based [22,11] and hybrid [23,18,14] approaches. In the domain of linked documents, link analysis for clustering and classification purposes has generally been studied in the context of web documents. PageRank [1] and HITS [13] are two of the most popular algorithms showing the importance of link structure for analyzing associations between documents.

For merging text-based and link-based information, [5] and [3] use generative probabilistic models of document content and connectivity. He et al. [23] use the hyperlink structure to cluster web pages using spectral graph partitioning. In their work, the link graph is used as the dominant source of similarity between documents, and the link-based similarity measures are augmented by textual content similarity and co-citation similarity. [14] propose a probabilistic model of link structure based on the cluster membership. The model is optimized based on observed data where the attributes determine the group membership and group membership determines the link structure. Modha et al. [17] propose an algorithm for clustering hypertext documents by using both the document contents and link structure. The algorithm uses an extended version of the classical Euclidean K-means clustering algorithm that performs clustering based on word similarity, in-link similarity and out-link similarity. The effect of each similarity is controlled by a parameter, which needs to be explicitly set by the user.

A number of algorithms have been proposed for *link prediction*, which is the task of identifying the missing entity or entities of a partially observed link by using the existing observation of the data sample available in the domain. [4] uses directed graphical models (Bayesian Networks and Probabilistic Relational Models) to represent a probabilistic model of both links and data object attributes. A comparison of various machine learning approaches for link prediction/completion is given in [8]. One major drawback of model-based link prediction is the dependence on the training data. That is, the learner's builds a probabilistic model on the training data, it will lack confidence in the probabilities of the entities that have not been included in the training set.

3 Problem Description

Documents on similar topics exhibit specific characteristics that separate them from non-relevant documents. Similar documents cite each other and they contain some level of textual similarity, measured by the amount of overlapping words/phrases. Some of those terms are very general and are not useful for clustering purposes. Some, on the other hand, are highly correlated with the topics of the papers and they are very valuable for identifying topical clusters in the collection. Although both textual content and link structure can be used independently for topical clustering, an algorithm that merges both sources of heterogeneous data has the potential of yielding better clustering solutions than using either data source alone. If the link structure of the documents are dense enough, then link based clustering, augmented by textual content, will generally yield well separated clusters. On the other hand, in situations where link graph is sparse, access to linking and/or linked documents is limited, or there is some sort of ambiguity in the link structure itself, the link graph can not be used as the dominant source of clustering. Thus, it is crucial to find a text-based clustering solution that incorporates information from the available link structure as well. Our work addresses this problem and provides an algorithm that bridges the disconnect between text and citations of papers by discovering the set of words that are most informative in terms of identifying citation relationships. We then place higher emphasis on such words in the clustering stage, and discover topical clusters in the citation-augmented feature space.

4 Algorithm

In this section, $d_i \in \mathbb{R}^n$ denotes the m documents in the collection and C denotes the non-symmetric citation matrix where $C_{ij} = 1$ if d_i cites d_j, and zero otherwise. Each document is represented as a vector in the feature space. Following L_2 normalization of the document vectors so that each $||\boldsymbol{d_i}|| = 1$, we generate a similarity matrix S from the cosine similarities of each document pair:

$$S_{ij} = cos(d_i, d_j) = \frac{\boldsymbol{d_i^T} \cdot \boldsymbol{d_j}}{||\boldsymbol{d_i}|| \cdot ||\boldsymbol{d_j}||} \tag{1}$$

We then calculate, for each citing document, the average distance of its citations using the similarity matrix S and the citation graph as follows:

$$\mathbb{D}_i = \frac{\sum_{j=1}^{n} S_{ij} \cdot C_{ij}}{k_i} \tag{2}$$

where $\mathbb{D}_i$ represents the average citation distance(ACD) of document d_i, and k_i is the total number of citations of document d_i that is present in the collection. In this definition, only the citations in the collection can affect the distance metric, since, for missing citations, we do not have the text of the document and hence, S_{ij} will be zero. We are interested in evaluating the significance of each word by comparing its popularity both in document pairs *with* and *without* citations. To achieve this goal, the ACDs enable us to view the document space from these two perspectives by populating the following

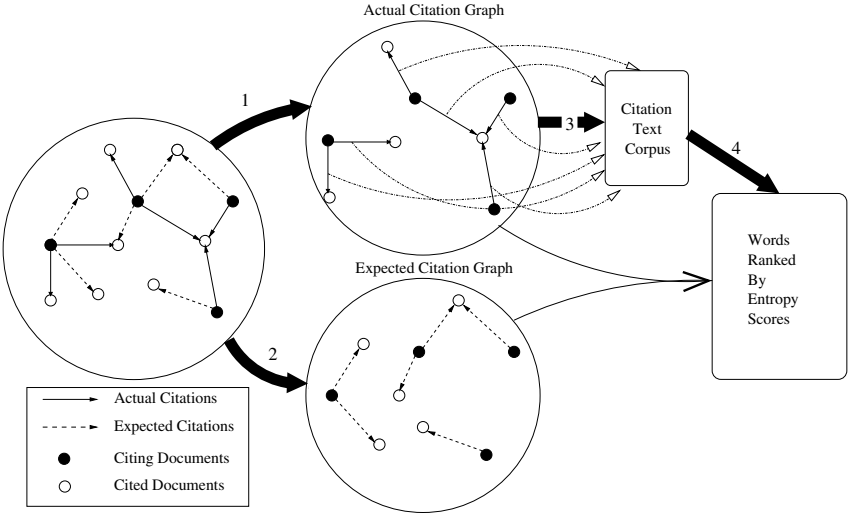
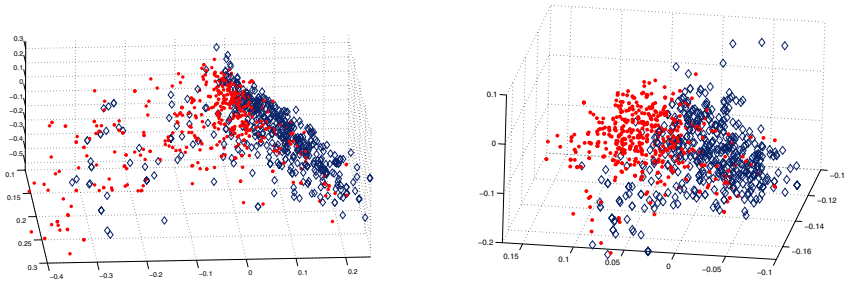


Fig. 1. Schematical view of the algorithm. The ACDs are used to find the expected citations and the given link structure is split into Actual Citation Graph G^A and Expected Citation Graph G^E (Steps 1 & 2). The set of words appearing in both in citing and cited documents in G^A are inserted in the Citation Text Corpus T (Step 3). For each word in T, we use the link and word co-occurence information from G^A and G^E to calculate the expected entropy loss scores (Step 4).

two sets: The first set, G^A is the *Actual Citation Graph* and is populated with the citing papers and their citations. This set is the collection of documents that form the citation graph. The second set, G^E, is the *Expected Citation Graph* and it is populated using the $\mathbb{D}_i$'s in the following fashion. For each document d_i having k_i citations (i.e. the citing documents in G^A), we select k_i documents that are **not** cited by d_i and is separated from d_i by a distance closest to a radius $\mathbb{D}_i$. That is, for each citing document d_i, we find k_i documents such that their content-wise similarity to d_i suggests that d_i should also be citing these documents, but no such citation exists in the graph for d_i. This set of documents is called *Expected Citation Graph* since we would expect these citations to exist based on the textual content of the papers.

Algorithm. Non-uniform Feature Weighting

1. Populate G^A with the documents in the citation graph
2. Initialize $G^E \leftarrow \emptyset$, $T \leftarrow \forall t_{ij}$ for $C_{ij} = 1$
3. **for** each citing document d_i in G^A with k_i citations **do**
4. $G^E \leftarrow G^E \cup \{k_i$ not-cited documents of d_i closest to $\mathbb{D}_i\}$
5. **end**
6. **for** each $t_p \in T$ **do**
7. E_p = Entropy loss calculated from equation 5.
8. $\bar{w}(d_i, t_p) \leftarrow (1 - \lambda) \cdot w(d_i, t_p) + \lambda \cdot E_p, \forall d_i \in \{d_1, d_2, \cdots, d_m\}$
9. **end**



(a) Original Document Space

(b) Document vectors adjusted by Entropy loss

Fig. 2. Effect of integrating entropy scores of citation corpus. Documents are mapped to 3D space by Singular Value Decomposition (SVD).

After populating G^E with the citing documents in G^A and their respective *expected citations*, the sets G^A and G^E have exactly the same number of edges, since we restrict G^E to contain the same linking vertices as G^A and insert exactly the same number of (expected) citations to it. This way, we enable each vertex (i.e. citing document) to be equally represented both in G^A and G^E. We then collect the common words between citing and cited documents in G^A. We do not consider the shared terms in the document pairs that are in G^E, since our aim is to identify the importance of the terms that appear in actual citation relationships.

We use expected entropy loss measure [7] to calculate the amount of information that each term in T conveys about citations. Our intuition is to find a numerical representation of the importance of each feature that is shared by the documents linked together. This also enables us to learn what makes document A cite document B and not cite C, although B and C may also be similar based on textual content. Clearly, it is not possible for a paper about, say, *data mining* to cite all the literature about this topic. Due to this fact, since lack of a citation can't be regarded as irrelevance, if we can identify the terms that influence citations, we can reflect the information obtained from the citations to better utilize the textual information of the documents for clustering purposes. If a word occurs frequently between citing and cited documents in G^A, but not in the expected citations in G^E, this word is regarded as a good candidate for being a topical word and is emphasized in the clustering algorithm. This approach serves as a means of eliminating one shortcoming of clustering algorithms; that is, each feature is weighted based on some corpus statistics and almost all clustering algorithms treat the attributes of data objects uniformly. We break this uniformity by reflecting the information obtained from the citation graph by scoring the shared terms of the citations using expected entropy loss.

4.1 Expected Entropy Loss

Given a text corpus comprising of n distinct features and k categories, expected entropy loss measures amount of categorical discriminative power of each feature in the dataset.

In formal definition, let C^A and C^E be the events of a sample being a member of the specified class, where the superscripts A and E refer to the actual and expected citation graphs, respectively. A sample in our case is the shared term between citing and cited documents. The prior entropy of the class distribution is

$$e = -P(C^A)lgP(C^A) - P(C^E)lgP(C^E) \tag{3}$$

The posterior entropy of the class distribution when feature f is present in the citation text corpus is

$$e_f = -P(C^A|f)lgP(C^A|f) - P(C^E|f)lgP(C^E|f) \tag{4}$$

The posterior entropy of the class distribution when feature f is absent in the corpus is denoted as $e_{\bar{f}}$ and can be found in a similar manner. Thus, the posterior expected entropy is $e_f P(f) + e_{\bar{f}} P(\bar{f})$ and expected entropy loss is defined as

$$Ent.Loss(f) = e - (e_f P(f) + e_{\bar{f}} P(\bar{f})) \tag{5}$$

which is always positive for every feature f.

4.2 Feature Weight Adjustment

The citation text corpus T contains the shared words between citing and cited documents in G^A (which is a subset of the original feature space) and we use this subset to realign the document vectors. Expected entropy loss based ranking of the most and least informative words in the corpus T is given in Table 1. It can be noted that more meaningful and topic bearing terms rank higher than less informative terms. Hence, by integrating the entropy loss information into the document vector representations, it is possible to achieve better separation of the distinct clusters. For each word in T, we update each document vector containing that feature as follows:

$$\bar{w}(d_i, f_j) \leftarrow (1 - \lambda) \cdot w(d_i, f_j) + \lambda \cdot Ent.Loss(f_j) \tag{6}$$

for $i = [1 \cdots n]$, $\forall f_j \in \boldsymbol{d_i}$. $w(d_i, f_j)$ represents the original Term Frequency-Inverse Document Frequency (TF-IDF) score of feature f_j in document d_i and λ is a parameter that adjusts the effect of the information gain of the feature on the final weight, which can also be thought of as relative bias of that term in the document. $\lambda = 0$ refers to the original weighting scheme and $\lambda = 1$ corresponds to purely entropy score based weighting. Hence, λ has the effect of proportionally reducing the significance of the features that don't exist in the citation corpus. Following the weight updates of the features of all documents, the document vectors are re-normalized to unit length. We then perform clustering on the updated document vectors. A visual representation of the effects of the weight readjustment is shown in Figure 2 for categories 1 and 4 of our dataset, which are the two clusters most difficult clusters to separate. It can be seen that comparably cleaner separation of the clusters can be achieved by entropy based weight readjustment of the features for these most overlapping categories. Computationally, given a dataset with N documents, C citations and a text corpus of T, the complexity of generating the similarity matrix and formation of the expected citation graph is $O(TN^2)$ and the calculation of the expected entropy losses is bounded by $O(CT)$. So the overall complexity of the algorithm is $O(T(N^2 + C))$.

Table 1. Features ranked by decreasing expected entropy loss

Rank	Feature
1.	automata
2.	radio
3.	collapse
4.	realtime
5.	switchboard
6.	tcp
7.	molecular
8.	fluctuate
9.	grayscale
10.	dendogram
...	...
...	...
...	...
5547.	statement
5548.	quinlan
5549.	roth

5 Experiments

We used a selection of 7227 papers from CiteSeer's repository as our dataset. The papers are split into 5 groups based on their publication venues. The categorical distribution of the publication venues is shown in Table 2. We selected the first 1000 words of each paper, resulting in a text corpus of 9601 distinct features after preprocessing the text by stemming, stop word and infrequent word removal. The clustering is performed both using the original TF-IDF scores of words and the scores augmented by the entropies of the words. A total of 4404 citation relationships exist between the papers in the dataset. The text corpus T of the citation relationships consists of distinct 5449 words. We used the Cluto [12] clustering toolkit in our experiments. Cluto implements some of the most widely used clustering algorithms in the literature, including agglomerative, divisive and graph-based techniques and hence, provides good baseline comparisons.

5.1 Evaluation Metric

The clustering performance is evaluated by comparing the predicted cluster of each document with the categorical labels (venues) from the document corpus. We used the standard F_1 and entropy measures as our evaluation criteria. F_1 measure combines precision (p) and recall (r) with equal weight in the form of $F_1(p, r) = \frac{2.p.r}{p+r}$. We report results both on Macro-averaged F_1 and Micro-averaged F_1 scores. The key difference between those two F_1 measures is that macro-averaging gives equal weight to each cluster, whereas micro-averaging equally weights each document. The cluster entropy measure shows the distribution of various classes of documents within each cluster. For each cluster C_i of size n_i, the entropy of this cluster is defined as

$$E(C_i) = -\frac{1}{log\ k} \sum_{j=1}^{k} \frac{n_i^j}{n_i} log \frac{n_i^j}{n_i} \tag{7}$$

Table 2. Dataset Venue Distribution

Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
Venue	Samples	Venue	Samples	Venue	Samples	Venue	Samples	Venue	Samples
AAAI	662	POPL	599	ICCV	682	ICML	990	VLDB	1049
IJCAI	599	PLDI	664	CVPR	830	ECML	211		
ICTAI	232					ML	80		
						KDD	629		
total	1493	*total*	1263	*total*	1512	*total*	1910	*total*	1049

where k is the number of classes in the dataset and n_i^j is the number of documents of the i^{th} class that were assigned to the jth cluster.

The entropy of the entire clustering solution is the average of the cluster entropies adjusted by their respective sizes, given by $\sum_{i=1}^{k} \frac{n_i}{n} E(C_i)$. A smaller entropy score indicates better clustering solution over the entire dataset.

5.2 Results on Four Criterion Functions

We evaluated our algorithm using the following four different similarity criterion functions. Each criterion function represents the objective that we try to optimize for discovering clusters. The first criterion, I_{sim}, is an *internal* similarity metric that tries to maximize the similarity between each document and the centroid of its assigned cluster. The second criterion function, E_{sim}, is an *external* approach that tries to separate the documents of each cluster from the entire collection. The *hybrid* approach, H_{sim}, tries to find a clustering solution by optimizing the inter-cluster (I_{sim}) and intra-cluster (E_{sim}) similarity metrics simultaneously. The final criterion, G_{sim}, uses the similarity graph of the documents and tries to find the optimum cuts of the graph using MinMaxCut algorithm.

$$maximize\ I_{sim}(S) = \sum_{r=1}^{k} \sum_{d_i \in S_r} cos(d_i, C_r) \tag{8}$$

$$minimize\ E_{sim}(S) = \sum_{i=1}^{k} n_i \frac{\sum_{v \in S_i, u \in S} cos(v, u)}{\sqrt{\sum_{v,u \in S_i} cos(v, u)}} \tag{9}$$

$$maximize\ H_{sim}(S) = \frac{I_{sim}}{E_{sim}} \tag{10}$$

$$minimize\ G_{sim}(S) = \sum_{m=1}^{k} n_m^2 \frac{cut(S_m, S - S_m)}{\sum_{d_i, d_j \in S_m} cos(d_i, d_j)} \tag{11}$$

The results of the clustering solutions using the four criterion functions are given in Table 3 for $\lambda = 0$ and $\lambda = 0.15$. S and $\mathbb{S}$ refer to the original and updated document similarities, respectively.

In all four criterions, we were able to achieve better clustering solutions using the entropy-based weight adjustments of the features. The most benefit can be observed

Table 3. Results on four different clustering criterion functions

	Internal Similarity		External Similarity		Hybrid		Graph-based	
	$I_{sim}(S)$	$I_{sim}(\$)$	$E_{sim}(S)$	$E_{sim}(\$)$	$H_{sim}(S)$	$H_{sim}(\$)$	$G_{sim}(S)$	$G_{sim}(\$)$
$F_1(Micro)$	80.7%	**85.7%**	80.8%	**81.9%**	76.1%	**82.5%**	81.5%	**84.4%**
$F_1(Macro)$	81.5%	**86.7%**	81.4%	**82.3%**	76.8%	**83.2%**	81.8%	**84.8%**
Entropy	36.7%	**28.7%**	37.5%	**33.5%**	41.8%	**32.8%**	36.3%	**34.3%**

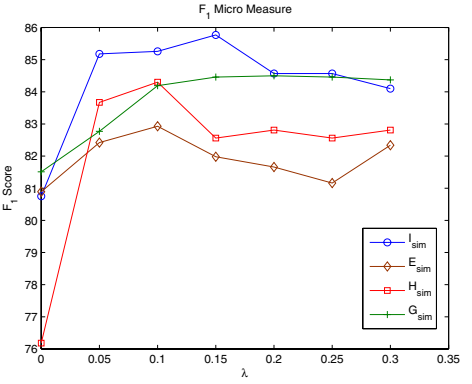


Fig. 3. F_1 Micro score variation based on λ

for I_{sim} and H_{sim} similarity metrics, indicating that similar documents are grouped into much compact clusters. This behavior is expected since the citations we used were mostly to the papers that are in the same category, hence we boosted the weights of the terms that collectively define their respective categories, hence maximizing the internal similarity of the documents of the same cluster. In Figures 3 and 4, we show the effect of varying λ on F_1 and entropy scores of the clustering solution for all criterion functions. Even for the $\lambda = 0.05$ case which indicates only a slight support from the

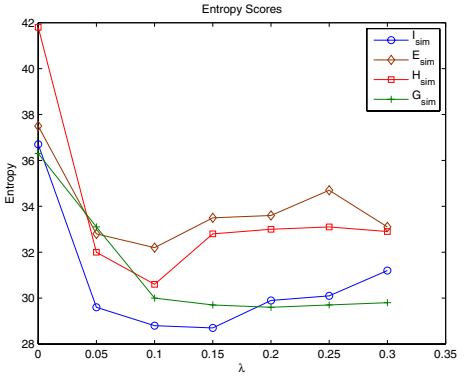


Fig. 4. Entropy score variation based on λ

entropies on the feature values, all four criterion functions achieve significant accuracy improvement. Further increasing λ over 0.25 either has no, or negative effect on the clustering solution.

Since the entropy values are needed for the *bias* effect on feature weights, increasing λ beyond a certain point starts to cause a dominating effect on the document vectors. In that case, the documents containing just a couple of common words (i.e. "database", "collection", "learning") tend to group together, causing an adverse effect. It is therefore desirable to keep λ at values that is sufficient enough to contribute to the weights without significantly modifying them.

6 Conclusions

Most clustering algorithms assume that the components of data objects are independent and identically distributed. This assumption has led to the design of numerous supervised and unsupervised learning algorithms to work on such "flat" data, where each data instance is a fixed length vector of attribute values. For data sets where the data set has richer structure, such as hyperlinks in web documents and citations in scientific literature, an efficient and effective solution to incorporate the connectivity information in the clustering solution yields better clustering performance. In this paper, we presented an algorithm that incorporates the citation graph of a collection of scientific literature to the clustering solution to better identify distinct groups of documents. The existence and non-existence of citation relationships of papers are used to identify the most important topic-bearing words in the papers, based on expected entropy loss measure. We have shown that a feature weighting scheme incorporating the citation-based extraction of topically significant words and applying partial bias for those terms can effectively discover clusters of similar papers.

References

1. S. Brin and L. Page. The anatomy of a large scale hypertextual web search engine. In *7th WWW Conference*, 1998.
2. T. Chiu, D. Fang, J. Chen, Y. Wang, and C. Jeris. A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *KDD '01*, pages 263–268, 2001.
3. David Cohn and Thomas Hoffmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS*, 2001.
4. L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational data. In *ICML'02*, pages 170–177, 2001.
5. Lise Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, pages 679–708, 2002.
6. C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In *The 3rd ACM Conf. on Digital Libraries*, pages 89–98, 1998.
7. E. Glover, S. Lawrence G. Flake, A. Kruger, D. Pennock, W. P. Birmingham, and C. L. Giles. Improving category specific web search by learning query modifications. In *SAINT '01*, page 23, 2001.
8. Anna Goldenberg, Jeremy Kubica, Paul Komarek, Andrew Moore, and Jeff Schneider. A comparison of statistical and machine learning algorithms on the task of link completion. In *KDD Workshop on Link Analysis for Detecting Complex Behavior*, August 2003.

9. Z. Chen H-J. Zeng, Q-C. He, W-Y Ma, and J. Ma. Learning to cluster web search results. In *SIGIR'04*, pages 210–217, 2004.
10. K. Hayland. Self-citation and self-reference: credibility and promotion in academic publication. *Journal of the Academic Society for Information Science*, 54(3):251–259, 2003.
11. J. Hou and Y. Zhang. Utilizing hyperlink transitivity to improve web page clustering. In *Proc. of 14th Australasian database conference on Database technologies*, pages 49–57, 2003.
12. George Karypis. Cluto, 2002. http://glaros.dtc.umn.edu/gkhome/views/cluto/.
13. Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
14. J. Kubica, A. Moore, J. Schneider, and Y. Yang. Stochastic link and group detection. In *18th National Conference on Artificial Intelligence (AAAI'02)*, 2002.
15. Steve Lawrence. Online or invisible. *Nature*, 411(6837):521, 2001.
16. Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.
17. D. Modha and W. Spangler. Clustering hypertext with applications to web searching. In *11th ACM Conf. on Hypertext and Hypermedia*, 2000.
18. J. M. Neville and D. Jensen. Clustering relational data using attribute and link information. In *Text Mining and Link Analysis Workshop, 18th Int'l Conf. on Artificial Intelligence*, 2003.
19. H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty in citation matching. In *Advances in Neural Information Processing*, 2003.
20. R. Rastogi S. Guha and K. Shim. Cure: an efficient clustering algorithm for large databases. In *SIGMOD '98*, pages 73–84, 1998.
21. Y. Gong W. Xu, X. Liu. Document clustering based on non-negative matrix factorization. In *SIGIR'03*, pages 267–273, 2003.
22. Yitong Wang and M. Kitsuregawa. Use link-based clustering to improve web search results. In *Second Int'l. Conf. on Web Information Systems Engineering*, December 2001.
23. C. Ding X. He, H. Zha and H. Simon. Web document clustering using hyperlink structures. *Computational Statistics and Data Analysis*, 41:19–45, 2002.
24. M. Yitzhaki. The language preference in sociology: measurements of 'language self-citation', 'relative own language preference indicator' and 'mutual use of languages'. *Scientometrics*, 41:243–254, 1998.