A Neighborhood-Based Approach for Clustering of Linked Document Collections

Ralitsa Angelova, Stefan Siersdorfer

MPI–I–2006–5-005 August 2006

Authors' Addresses

Ralitsa Angelova, Stefan Siersdorfer
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
{angelova, stesi}@mpi-inf.mpg.de

Abstract

This technical report addresses the problem of automatically structuring linked document collections by using clustering. In contrast to traditional clustering, we study the clustering problem in the light of available link structure information for the data set (e.g., hyperlinks among web documents or co-authorship among bibliographic data entries). Our approach is based on iterative relaxation of cluster assignments, and can be built on top of any clustering algorithm (e.g., k-means or DBSCAN). These techniques result in higher cluster purity, better overall accuracy, and make self-organization more robust. Our comprehensive experiments on three different real-world corpora demonstrate the benefits of our approach.

Keywords

Clustering, Exploiting Link Structure

Contents

1 Introduction **2**
- 1.1 Motivation . . . 2
- 1.2 Related Work . . . 3
- 1.3 Contribution . . . 4
- 1.4 Outline . . . 5

2 Technical Basics **6**

3 A Probabilistic Framework for Graph-based Clustering **8**
- 3.1 Content Combination . . . 8
- 3.2 Confidence Measures for Clustering . . . 9
- 3.3 Including Neighborhood Information . . . 9

4 Extensions of the Framework and Parameter Estimation **12**
- 4.1 Soft Clustering . . . 13
- 4.2 Hard clustering . . . 13
- 4.3 Separation cost . . . 14
- 4.4 Edge pruning and weighting . . . 14
- 4.5 Incorporating Metric Cluster Distances . . . 15

5 Experiments **17**
- 5.1 Quality Metrics for Clustering . . . 17
- 5.2 Setup . . . 17
- 5.3 Results . . . 18

6 Conclusion and Future Work **23**

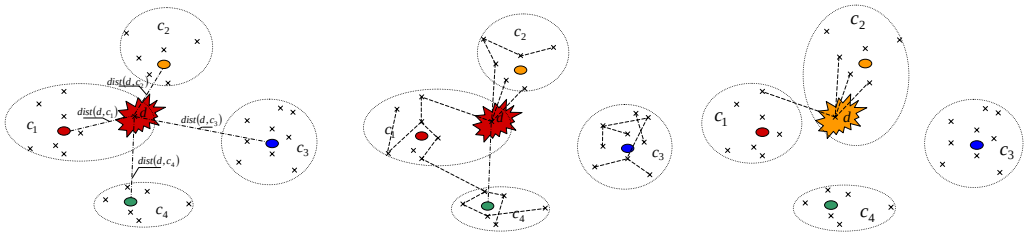
1 Introduction

1.1 Motivation

This report tackles the problem of automatically structuring heterogeneous document collections into thematically coherent subsets. This issue is relevant for a variety of applications, such as organizing large personal email folders, dividing topics in large web directories into subtopics, structuring large amounts of company and intranet data, etc. Two major techniques are employed to address the problem: one is based on supervised classification, which requires explicit manually labeled training data, the other takes advantage of the so called unsupervised clustering. It is quite often the case that explicit training data is unavailable, or very time consuming and expensive to gather, so that clustering is the only viable option.

For conventional document clustering, a purely content based representation, e.g. based on frequencies of words or word stems in a Web page, is used as an input for the clustering algorithm. This "context-free" approach does not exploit the available information about relationships between documents. However, such information might be crucial for the purity of the final clusters. For example, if we are asked to cluster companies selling products on eBay into trustworthy or not trustworthy, it would be of great help if we could access all forums and customer pages that discuss the products in question and their quality. Thus link information provides refined clues that later on can crucially influence the final clustering.

The intuition behind our approach is sketched in Figure 1.1. Figure 1.1 (a) shows the clustering entirely based on the content information about each document. Here document d is assigned to its nearest (most similar) cluster. The similarity of document d is measured with respect to the cluster centroids, (e.g. produced by a content-based clustering algorithm such as k-means - see Section 2). Figure 1.1 (b) shows the link structure among the documents. In the content-only world, such link information is completely ignored. Our attempt to make use of it starts with the observation that



(a) Content-Based Clustering (b) Link Structure (c) Final Clustering

Figure 1.1: Neighborhood-Based Clustering

document d is linked to a significantly higher number of documents from cluster c_2 than from cluster c_1. Furthermore, the graph structure suggests that documents from cluster c_2 typically link to documents that belong to this very same cluster - c_2. Thus, with high probability a document can be clustered in c_2 if it is linked to many documents that belong to cluster c_2. In our toy example, this leads to reassigning document d to cluster c_2. The final clustering after taking both content and link information into account is shown in Figure 1.1 (c).

1.2 Related Work

There has been considerable prior work on *classification* that takes advantage of the available neighborhood information. In [16] such a classification setting is formalized as a metric labeling problem. Chakrabarti et. Al. [4] propose an iterative relaxation method for combining link and content information to derive better classification performance. Other methods [22, 18] try to incorporate the link information directly into the document vector representation.

Closest to the approach presented in this technical report is our own recent work on neighborhood-based classification [2]. However, all of these approaches are supervised, thus based on training data, and are not considered in the context of (unsupervised) clustering.

Graph-based clustering is well established in the literature. For an overview of existing methods see [25]. The underlying graph G is constructed by representing each data point as a node in G and each edge, connecting any two data points, by a weight, indicating the distance (dissimilarity) between its end points. To cluster such a graph G, the graph-based algorithms typically first create a minimum spanning tree of the graph G. Then, they repeatedly

remove edges whose weight is largest in relation to the edge weights in their environment in the minimum spanning tree of G until the number of desired clusters is achieved [29, 27, 14]. Other methods addressing the graph-based clustering problem use Singular Value Decomposition [6].

Clustering plays an important role in image processing. Instead of coping with documents represented by a feature vector, they operate over an image represented by color components. Some of the developed image clustering algorithms [15, 13, 21] are based on Markov Random Field (MRF). They first cluster the image regions based on their color values, and then refine the clustering using MRF processing. However, compared to neighborhood-aware clustering of document collections, the size of the neighborhood considered in image processing (i.e., typically consisting of 8 neighboring pixels) and the data item representation vectors (limited to tens of features) are very small.

Another way of addressing the graph-based clustering problem is proposed in [25]. In brief, it is an extension of the well known k-means algorithm [12] that is based on graph theoretic distance measures.

However, our approach is orthogonal to all these approaches as we use statistical knowledge about the cluster assignments of the nodes in the formed neighborhoods in G. Furthermore, the assignment of the edge weights, and thus the type of graphs used by the above approaches, are based on node-node similarity, and it is not clear how to carry this forward to a hyperlinked environment.

1.3 Contribution

Our contributions can be summarized into three major points:

- We develop a systematic and comprehensive framework for graph-based clustering that uses the theory of Markov Random Fields [5, 17] and a Relaxation Labeling technique [23] to derive robust clustering algorithm.

- We identify methods to efficiently estimate the parameters in this framework, and beneficial extensions of the framework leading to significant gains in the performance, for example, by introducing metric cluster distances or selective neighborhood influence on content level.

- We provide a comprehensive experimental study of the performance of the proposed algorithms over three real life data collections: DBLP[1],

[1]Computer Science Bibliography Data Base

IMDB[2], and Wikipedia[3].

1.4 Outline

The rest of this report is organized as follows. In Section 2 we briefly review the technical basics of clustering methods. Section 3 presents our framework of neighborhood-based clustering. Section 4 deals with the efficient parameter estimation within this framework, and we describe further extensions to improve the overall clustering quality. Section 5 provides experiments on different real-world datasets.

[2]Internet movie database
[3]Online Encyclopedia

2 Technical Basics

Clustering algorithms partition the set of given data objects into groups called *clusters*. The data items we consider are documents and each document is represented by a feature vector. In the prevalent bag-of-words model the features are derived from word occurrence frequencies [3, 20] (e.g., capturing tf or $tf * idf$ weights of terms). In addition, feature selection algorithms [19] can be applied to reduce the dimensionality of the feature space and eliminate "noisy", non-characteristic features. This type of noise filters are based on term frequencies or advanced information-theoretic measures for feature ordering (e.g., mutual information (MI) or information gain [19]).

Clustering methods can be divided into the following groups [9]: partitioning methods, hierarchical methods, density based methods, grid based methods, and model based methods. In this paper we consider partitioning methods: the dataset is divided into disjoint partitions. For this family of clustering algorithms, the number k of clusters is a tuning parameter [11].

A simple and very powerful member of the family of partitioning clustering methods is *k-means* [12]: k initial centers (points) are chosen, every document vector is assigned to the nearest center (according to some distance or similarity metric), and new centers are obtained by computing the means (centroids) of the sets of vectors in each cluster. After several iterations (according to a stopping criterion) one obtains the final centers, and, respectively, a clustering of the given document set can be derived. A similar algorithm, which can be considered as a "smoothed" form of k-means is *EM clustering* [11, 20]: in every iteration the probabilities of the objects for being contained in the different clusters are updated using the expectation-maximization technique.

The result and run-time for k-means and other iterative clustering algorithms are strongly dependent on the initial partitioning (for k-means this corresponds to the initial centers). A standard heuristics for this initialization phase is *pre-clustering* [9]: before starting the actual clustering algorithm, a

clustering is computed on a much smaller subset. This way one can often obtain better starting points.

3 A Probabilistic Framework for Graph-based Clustering

In this section we first describe a simple way of using the underlying link structure by combining the content of a document with the content of its neighbors. We show how to obtain confidence values from the content-based clustering. Last, we introduce a probabilistic framework which enables us to take advantage of the information contained in the neighborhood of each document.

3.1 Content Combination

An intuitive way of combining the content of a document d with the content of its neighbors $d' \in N(d)$ is to assign term weights $w'(t_i, d)$ to all terms $t_i \in d$ while considering in a linear way the term weights t_i in d's neighbors $d' \in N(d)$.

More precisely, let $w(t_i, d)$ be the original term weight (e.g., obtained by using the traditional $tf * idf$ value). We can compute the adjusted weight $w'(t_i, d)$ as follows:

$$w'(t_i, d) = w(t_i, d) + \sum_{d' \in N(d)} \alpha \cdot w(t_i, d') \tag{3.1}$$

Here, the parameter α controls the impact of the neighborhood content on the final term weights in document d. The correspondingly adjusted feature vectors can be used as an input to all vector-based clustering algorithms, e.g., the k-means algorithm.

To avoid the potential increase in the level of noise, we can consider only a subset of "good" neighbors. These neighbors should be similar enough to

the document in question, d. For this purpose, we introduce a similarity threshold, S-threshold, which can be computed as the cosine-similarity between the pair of neighboring documents and which selectively determines the neighborhood of each document d.

A similar approach was used in [22] for document classification. Although this is a fairly straightforward idea, we are not aware of prior literature that has explicitly considered such a re-weighting of terms in the context of clustering.

3.2 Confidence Measures for Clustering

Some clustering algorithms assign to each document a probability value of membership to any of the possible clusters. An example for such an algorithm is the EM clustering algorithm [20, 11]. Confidence values for cluster memberships can be also assigned to the results of other clustering algorithms, e.g., k-means.

The k-means algorithm provides us with a set of k centroids $\{z_1, \ldots, z_k\}$, where centroid z_i is a representation of cluster i. Note, that such a centroid or a similar representation can be easily computed for the clustering obtained by any other clustering algorithm. Each document d is assigned to a cluster i so that the similarity $sim(d, z_i)$ between z_i and the document d is maximized. As a similarity measure one could adopt the cosine similarity between d and z_i.

But the similarities $sim(d, z_i)$ can also be used to make soft assignments. The most intuitive way to do this is to assign to each document d a confidence value $\sigma(i, d)$ for the membership to cluster i proportional to $sim(d, z_i)$:

$$\sigma(i, d) = \frac{sim(d, z_i)}{\sum_{j=1}^{k} sim(d, z_j)} \tag{3.2}$$

We chose the normalization constant such that:

$$\sum_{j=1}^{k} \sigma(j, d) = 1 \tag{3.3}$$

3.3 Including Neighborhood Information

Our approach adopts a probabilistic formulation of the clustering problem. It is based on the so called relaxation labeling technique which was initially

proposed for resolving problems in image processing but is successfully applied in classification [4, 2] and as we show - in clustering. We propose two major approaches for finding the maximally likely clustering of the given test graph: *hard* and *soft* clustering. A maximally likely clustering of a graph aims to minimize the sum of two types of costs: assignment and separation cost. The first one is based on the individual choice of a cluster to which a document is assigned. The latter reflects the choice of a pair of clusters to which two neighboring documents to be.

Formally, we aim to cluster a set of documents $\mathcal{D}$, where each document $d \in \mathcal{D}$ corresponds to a vertex in the graph G and each link between two documents in $\mathcal{D}$ corresponds to an edge in G. The clustering algorithm requires as an input the the text of each document d and information about which documents of G constitute its neighborhood, $N(d)$. Let $c(d)$ denote the cluster of node d whose validity can be associated with a probability. The content of document d is represented as a set of terms that occur in d and denoted by $\tau(d)$. The output of the algorithm should be an assignment of clusters to the graph nodes such that each document $d \in G$ belongs to its maximally likely cluster i, selected from a finite set of clusters $[1..m]$.

Taking into account the underlying link structure and document d's content-based feature vector, the probability of a document d to be assigned to cluster i is:

$$\Pr[c(d) = i \mid \tau, G] = \Pr[c(d) = i \mid \tau(d), c(d_1), \cdots, c(d_l)]$$

where d_1 through d_l are the documents in $\mathcal{D}$.

In principle, such a model could even consider long-range influences among transitively related documents, with decreasing influence as the distance in the graph increases. For tractability, however, it makes sense to focus on the strongest dependencies among immediate neighbors. Such a model is called a first-order Markov Random Field or MRF [17, 23]. Computing the parameters of an MRF such that the likelihood of the observed training labels is maximized is a difficult problem that cannot be solved in closed analytic form and is typically addressed by an iteration technique known as relaxation labeling (RL). Our approach builds on this mathematical technique.

In the spirit of emphasizing the influence of the immediate neighbors for each document, $N(d)$, we obtain
$\Pr[c(d) = i \mid \tau(d), G] = \Pr[c(d) = i \mid \tau(d), N(d)]$ and denote it by $\Phi_{i,d}$. This reflects the MRF assumption that the label of a node (as a random variable) is conditionally independent of the labels of other nodes in the graph given the labels of its immediate neighbors. We abbreviate $\Pr[c(d) = i \mid \tau(d)]$, the graph-unaware probability based only on d's local content, by $\sigma_{i,d}$. Let $c(N(d))$ denotes the assignment of clusters to the group of neighbors of d, $N(d)$. The probability that a test document neighborhood $N(d)$ is assigned

to $c\,(N\,(d))$ is denoted by $\Pr\,[c\,(N\,(d))]$. Applying, for tractability, the additional independence assumption that there is no direct coupling between the content of a document and the labels of its neighbors, the following central equation holds for the total (prior, i.e., unconditional) probability $\Phi_{i,d}$, summing up the posterior (i.e., conditional) probabilities for all possible cluster assignments to the neighborhood $N\,(d)$:

$$\Phi_{i,d} = \sum_{c(N(d))} \sigma_{i,d} \cdot \Pr\,[c\,(d) = i \mid c\,(N\,(d))] \cdot \Pr\,[c\,(N\,(d))]$$

In the same vein, if we further assume independence among all neighbor labels of the same node (but still capturing the dependence between a node and each of its neighbors), we reach the following formulation for our neighborhood-conscious clustering problem:

$$\Phi_{i,d} = \sigma_{i,d} \cdot \sum_{c(N(d))} \left(\prod_{d' \in N(d)} \Pr\,[c\,(d) = i \wedge c\,(d') = j] \right).$$

This can be computed in an iterative RL manner as follows:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \sum_{c(N(d))} \left(\prod_{d' \in N(d)} \Pr\,[c\,(d) = i \wedge c\,(d') = j] \right)^{(r-1)}$$

where $r > 1$ and $i, j \in [1..m]$ are cluster assignments. With the short-hand notation
$\phi_{i,j} = \Pr\,[c\,(d) = i \wedge c\,(d') = j]$ we can rewrite this into:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \sum_{c(N(d))} \left(\prod_{d' \in N(d)} \phi_{i,j} \right)^{(r-1)} \tag{3.4}$$

4 Extensions of the Framework and Parameter Estimation

To this end we have presented the basic framework of our algorithm. This section tackles the problem of estimating the parameters for the initial solution and during the iteration of the relaxation labeling (clustering).

As we wish to take advantage of the content as well as the link information provided as an input to the algorithm, the cluster assignments to all test nodes for iteration ($r = 0$) are intuitively the assignments of a pure content-based clustering, e.g., k-means, to the test nodes. All iterations that follow are based on Equation (3.4). We iterate until the probabilities $\Phi_{i,d}$, for each document and cluster assignment, stabilize, i.e., the magnitude of change drops below some *stop parameter* ϵ. The relaxation is guaranteed to converge to a locally consistent assignment if initiated sufficiently close to a consistent labeling (clustering) [17, 23]. In [23] it is shown that the relaxation algorithms are local optimizers by nature (similarly to EM methods). They do not necessarily arrive at the global optimum. Given a relaxation clustering algorithm and the data, two factors affect the solution: the initial cluster assignments and the cost function used to iteratively find the global maximum of the cluster assigning function. In our case, the iterative scheme contains both factors: the initial cluster assignment from the content-based clustering in iteration $r = 0$, and the cost function involving a *dynamically* computed separation cost (i.e., assigning neighbors to different clusters), re-adjusted in each iteration.

Calculating the sum over all possible cluster assignments in Equation (3.4) is hard as we have $m^{|N(d)|}$summands, where m is the number of distinct clusters. To solve this problem we employ two major methods described in Subsections 4.1 and 4.2. Depending on the chosen method of clustering, we approximate the sum over all possible cluster assignments of the neighborhood to either its most significant summand, treating it as if it were the true set of clusters (*hard* clustering), or the p most significant summands (*soft*

clustering) and their associated probabilities where p is a tunable constant. The proposed graph-based clustering algorithm efficiently re-computes and updates the probabilities of particular cluster assignments to the neighborhood $N(d)$ after *each* RL iteration. An algorithm for computing the p most significant summands is proposed in [8]. Another fast and more suitable algorithm to compute them is presented in [2].

4.1 Soft Clustering

The *soft clustering* approach aims to achieve better accuracy of the clustering by avoiding the overly eager "rounding" that the *hard* clustering approach does. Instead, we take into account the p most significant combinations of cluster assignments to the test document neighborhood (Equation (3.4)). This is motivated by the observation that apart from the few most probable cluster assignments to the neighborhood (p), the remaining combinations of cluster assignments have very low probabilities. Thus, they do not contribute much to the calculation of $\Phi_{i,d}^{(r)}$ (the probability of document d to be assigned to cluster i) and can be ignored. This reduces the exponential number of summands in Equation (3.4) from $m^{|N(d)|}$ to p and makes its computation feasible.

4.2 Hard clustering

In contrast to the presented *soft* clustering approach, we also consider a method that takes into account only the *most probable* cluster assignments in the test document neighborhood to be significant for the $\Phi_{i,d}^{(r)} = \Pr\left[c\left(d\right) = i \mid \tau, N(d)\right]^{(r)}$ computation. We call this newly employed approach *hard clustering*. It might be seen as a crude approximation of the sum in Equation (3.4) but depending on the sets $\mathcal{C}$ and $\mathcal{D}$ this approximation gives very good empirical results (see Section 5). Our hopes for this method would be that it is more efficient and possibly more robust than the soft clustering method. Hard clustering can be thought of as a "high-level" noise reduction, since it trusts only the most probable neighborhood cluster assignments and ignores assignments with lower probability, thus reducing the chance of an incorrectly assigned node in the neighborhood to influence the cluster assignment of the current test node.

Let $c_{max}(d')$ be the maximum probable cluster for each document $d' \in N(d)$ as of iteration $(r-1)$:

$$c_{max}(d') = \arg_j \max \Pr\left[c\left(d'\right) = j\right]^{(r-1)}.$$

Then considering only the maximum probable $c(N(d))$ according to the node label probabilities of iteration $(r-1)$, Equation (3.4) can be written as:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \prod_{d' \in N(d)} \left(\phi_{i,c_{max}(d')}\right)^{(r-1)},$$

and its simplified form, presenting the product over the set of cluster assignments rather than documents in the neighborhood $N(d)$:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \left(\prod_{c_{max}(d') \in \mathcal{C}} \left(\phi_{i,c_{max}(d')}\right)^{n(c_{max}(d'))} \right)^{(r-1)}.$$

where $n(j)$ is the frequency of label assignment j in $N(d)$: $n(j) = |\{d' \in N(d) \mid c(d') = j\}|$.

4.3 Separation cost

The probability $\phi_{i,j}$ of the endpoints of the edge between documents d and d' to be assigned to a pair of clusters i, j depends on the cluster assignment conditional probabilities of the corresponding nodes from the previous iteration. We calculate $\phi_{i,j}$ by a smoothed estimator based on the frequencies of edges tentatively assigned to a pair of clusters i, j in round $(r-1)$ of the iteration scheme (using Laplace smoothing).

We use this method in combination with the *hard* and *soft* node clustering approach and abbreviate the corresponding clustering algorithms as $GC[H]$ and $GC[S]$.

4.4 Edge pruning and weighting

Our method uses link weights based on the following rationale. The cluster assignment of a document should be more influenced by neighbors with a homogeneous content that is thematically closely related to the document's own content. As an intuitive example suppose we are interested if a given Amazon product page sells cosmetics or routers. Each page contains highly valuable outgoing links, e.g., links to the product description on the manufacturer page or comparison between similar products. But all too often we would also find on the same product page many non-relevant links like a link to the Amazon's generic "latest products" page, user's shopping cart, profile, etc.

We aim to ignore the unnecessary and most probably noisy information behind these irrelevant links by assigning to each edge e a weight w_e equal to

the cosine similarity between the feature vectors of the documents connected by the edge.

This edge weighting schema is applied for noise reduction in two ways: 1) we prune edges whose weight w_e is below a specified *similarity threshold (S-threshold)*), and 2) we differentiate links by using higher w_e to promote links that are considered more important. These considerations lead to the following revised label probability for the RL algorithm:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \sum_{c(N_d)} \prod_{d' \in N_d} \phi_{c(d),c(d')}^{(r-1)} \cdot w_e \tag{4.1}$$

4.5 Incorporating Metric Cluster Distances

Intuitively, neighboring documents should receive similar cluster assignments. For example, suppose we have a set of clusters $\mathcal{C} = \{$*Culture (C), Entertainment (E), Science (S)*$\}$ and we wish to find the most probable cluster for a test document d. Typically, documents that are related to culture or entertainment have overlapping areas of discussion like concerts, exhibitions, etc. This means, the clusters C and E are thematically close to each other. On the other hand, a document discussing scientific problems (S) would be much farther away from both C and E. So, a similarity metric $\Gamma(\cdot,\cdot)$ imposed on the set of clusters $\mathcal{C}$ would have high values for the pair (C, E) and small values for cluster pairs (C, S) and (E, S). Back to our example, suppose that document d has four neighbors and that these are assigned to clusters $c(N(d)) = \{C, C, E, E\}$. Then, the probability that document d would be assigned to cluster C should be higher than the same probability if the neighboring documents of d were clustered in $c(N(d)) = \{C, C, S, S\}$. Note that both of these neighbor assignments $c(N(d))$ have the same number of C clusters; the key lies in the different similarities between C and the other clusters in each of the two $c(N(d))$. The intuition suggests that the clustering result should approximately capture the topic structure in the data set.

This is why introducing a metric $\Gamma(\cdot,\cdot)$ should help improve the clustering result. In this metric, similar clusters are separated by a shorter distance and impose smaller separation cost on an edge cluster assignments.

The theory paper by Kleinberg and Tardos [16] suggests constructing an *r-HST tree* approximation for obtaining such a metric. The argument for this specific approximation is tractability of the otherwise NP-hard optimization problem, but the approach loses generality.

Our approach, on the other hand, is general, and we construct the metric Γ automatically from the test data. The metric value for a pair of labels

(i, j) is computed by the content similarity between the corresponding sets of documents contained in clusters i and j. In our experiments, we use the term-vector based cosine similarity between the "super-documents" that concatenate all documents of the same cluster. This metric is computed only once, and does not change as the relaxation labeling proceeds.

We incorporate the distance metric into the iterations for computing the probability of an edge cluster assignments $\phi_{i,j}$ by treating $\Gamma(i, j)$ as a scaling factor. This way, we magnify the impact of edges between nodes assigned to similar clusters and scale down the impact of edges between the nodes clustered into dissimilar ones:

$$\Phi_{i,d}^{(r)} = \sigma_{i,d} \cdot \sum_{c(N_d)} \prod_{d' \in N_d} \phi_{c(d),c(d')}^{(r-1)} \cdot w_e \cdot \Gamma(c(d), c(d')) \tag{4.2}$$

The product of the ϕ and Γ terms on the right-hand side can be viewed as the probability that d and d' are thematically related, having specific labels with probability ϕ and the labels being thematically close with probability Γ. Our experimental results (Section 5) show that incorporating Γ into the relaxation labeling significantly contributes to more accurate clustering results. In such cases the algorithm exploits the additional "guidance" that the cluster metric provides for estimating the edge label probabilities in each iteration.

5 Experiments

5.1 Quality Metrics for Clustering

Our quality measure describes the correlation between the actual topics of our datasets and the clusters found by the algorithm. Note that the cluster labels can be permutated: Given two classes $class_1$ and $class_2$, it does not matter whether a clustering algorithm assigns label a to all documents contained to $class_1$ and label b contained in $class_2$ or vice versa; the documents belonging together are correctly put together and the quality should reach its maximum value (i.e., 1) and the error should be 0.

Let k be the number of classes and clusters, N_i the total number of documents in $class_i$, N_{ij} the number of documents contained in $class_i$ and having cluster label j. We define the accuracy of the clustering as:

$$A = \max_{(j_1,\ldots,j_k)\in perm((1,\ldots,k))} \frac{\sum_{i=1}^{k} N_{i,j_i}}{\sum_{i=1}^{k} N_i} \tag{5.1}$$

5.2 Setup

We have tested our graph-based clustering algorithm on three different data sets. The first one includes approximately 16000 scientific publications chosen from the DBLP database. The set of classes includes "Database" (DB), "Machine Learning" (ML), and "Theory" as labels. We labeled the documents based on the conference in which they were published. For example, if a paper appeared in SIGMOD or VLDB proceedings, then it was assigned to the DB set. We chose co-authorship as the relation determining the connectivity between documents. For the initial step of purely content-based clustering we use the document titles as the only source.

The second dataset has been selected from the Internet movie database IMDB. For our tests we took into account all movies in which a given set of 80 famous actors occur (e.g., Johnny Depp, Bruce Willis, Mel Gibson,

etc.). This resulted in about 5000 movies grouped in 4 genres: "Action", "Comedy", "Documentary", and "Drama". This dataset is challenging for automatic clustering because initially many movies were "hand-tagged" with more than one genre. For our test we merged multiple genres of the same movie by using only the most "prominent" one as a "true" genre of a movie. For example, if a movie had a label *Action and Science fiction*, we considered it to be *Action*. For each movie we took its title and plot as a source of features for the initial content-based classification. We removed all stopwords and applied stemming to reduce the dimensionality of the feature space. This resulted in a feature set of around 19 000 terms. We built an edge between two documents (movies) in the graph if they have a starring actor in common. We applied similarity-based edge pruning for noise reduction. Our default S-threshold 0.25 left us with 800 nodes each of which has at least one neighbor. All "singleton" nodes without edges were disregarded in the experiments as all graph-aware methods would behave identically to content-based clustering on these nodes.

The third dataset used in the experiments was the online encyclopedia Wikipedia. We crawled about 5000 documents from the released Wikipedia dump file. As links between the pages we used their natural hyperlink connectivity. We restricted the crawler to follow only links within Wikipedia. The feature space had about 70 000 unique terms; no feature selection was performed. The distinguished classes are "Politics", "Computer Science", "Physics", "Chemistry", "Biology", "Mathematics", and "Geography". We started the crawl from the main pages for each of these subjects and used topic-specific words in the anchor text as indicators for whether an outgoing link should be followed. For example, we started gathering documents from the page *http://en.wikipedia.org/wiki/Chemistry* by following links that contained manually selected word stems like "chemist", "isomer", "branch", "organic", etc. All pages gathered from the starting chemistry page were considered to be "hand-tagged" as *Chemistry*. When the same page was discovered following paths with starting points of two different seeds, the page was discarded since no decision about its "true" label without human assessment could be taken.

All datasets for our experiments are available at the URL http://www.mpi-inf.mpg.de/∼angelova.

5.3 Results

We compared the following methods:

1. Content-based k-Means; we used the standard bag-of-words model [3]

(using term frequencies to build L1-normalized feature vectors, stemming with the algorithm of Porter [24], and deletion of stopwords) for document representation; feature selection according to *df*, (k**-Means**).

2. *Hard* graph-based clustering method as described in Section 4.2, (**GC[H]**).

3. *Soft* graph-based clustering method as described in Section 4.1, (**GC[S]**).

4. Graph-based clustering taking into account the cluster metric $\Gamma(\cdot,\cdot)$ and the edge weights w_e in the test graph as in Equation 4.2 of Section 4.5, (**wmGC[H]** or **wmGC[S]**).

5. GC using the edge weighting scheme as in Equation 4.1 of Section 4.3, (**wGC[H]** or **wGC[S]**).

6. Graph-based clustering taking into account the cluster metric $\Gamma(\cdot,\cdot)$ of Section 4.5, (**mGC[H]** or **mGC[S]**).

7. Content Combination described in Section 3.1 with α, (**CComb**[α]).

All graph-based methods 2 trough 6 use the result of the simple content based method 1 in the initialization step.

Adding the content combination approach provokes another variation of all other methods mentioned so far. To study the influence of this parameter we tested all graph-based methods 2 trough 6 with different values of α which are correspondingly shown in squared brackets after the method abbreviation. For example,

- GC[H][0.5] denotes the graph-based clustering GC[H] on top of CComb[0.5];
- wmGC[H][1.0] denotes the graph-based clustering wmGC[H] on top of CComb[1.0], and so on.

All experiments, unless otherwise stated, were performed using similarity threshold 0.3 and feature sets for DBLP and Wikipedia of size 2000, and for IMDB of size 2500. Note that the Wikipedia set is the most representative test data set. This is due to the fact that the documents in Wikipedia are with predefined true topics which ensures the precision of the clustering evaluation.

We also tested an MST-based graph-cut clustering algorithm [29], computing the edge weights as weighted sum of hyperlink based neighborhood and content similarity of the documents, and pruning edges in the corresponding spanning tree. However in our preliminary experiments, the k-means algorithm (despite of its simplicity) showed superior performance on

our data sets. Hence, we did not consider building our algorithm on top of a graph-cut approach.

The outcome of the comparison among the above methods along with the 95% confidence intervals is shown in Table 5.1. Table 5.2 describes the influence of the cluster metric (Section 4.5) on the performance of the proposed graph-based methods. A detailed sensitivity analysis of the parameters is presented in Figure 5.1.

Table 5.1: Comparison of Clustering Methods

	DBLP	**IMDB**	**Wikipedia**
	A	A	A
$kMeans$	0.4245±0.0074	0.3569±0.0145	0.5054±0.0133
$GC[H]$	0.4609±0.0075	0.3809±0.0147	0.5497±0.0133
$GC[S]$	0.4379±0.0075	0.3948±0.0148	0.5261±0.0133
$wmGC[H]$	0.4689±0.0075	0.3790±0.0147	0.5938±0.0131
$wmGC[S]$	0.4448±0.0075	0.3998±0.0149	0.5872±0.0131
$CCkMeans[0.5]$	0.4053±0.0074	0.4022±0.0149	0.5953±0.0131
$GC[H][0.5]$	0.4649±0.0075	0.4216±0.0150	0.6231±0.0129
$GC[S][0.5]$	0.5245±0.0075	0.3655±0.0146	0.6229±0.0129
$wmGC[H][0.5]$	0.4893±0.0075	0.4389±0.0147	0.6369±0.0128
$wmGC[S][0.5]$	0.4903±0.0075	0.4378±0.0150	0.6367±0.0128
$CCkMeans[1.0]$	0.5218±0.0075	0.4024±0.0149	0.6391±0.0128
$GC[H][1.0]$	0.5914±0.0074	0.4338±0.0150	0.6254±0.0129
$GC[S][1.0]$	0.5844±0.0075	0.4222±0.0150	0.6220±0.0129
$wmGC[H][1.0]$	0.6108±0.0075	0.4540±0.0147	0.6394±0.0128
$wmGC[S][1.0]$	0.5980±0.0075	0.4373±0.0150	0.6359±0.0128

The main observation are:

- The graph-based approach significantly outperforms all pure content-based methods. Our experiments show improvements of up to 9% over the k-Means algorithm as well as significant gains close to 10% over the content combination approach proposed in Section 3.1.
- The performance of the graph-based clustering is even better if the content combination technique discussed in Section 3.1 is used as initialization step for the graph-based methods.
- Including the cluster distance metric in the computations improves the graph-based clustering by gently imposing constraints on the separations cost estimates but resulting in up to 6% gain in accuracy in some cases.

For all data sets the similarity threshold (S-threshold) acts as a noise filter and improves the performance of the graph-based clustering. However, increasing the S-threshold to a very high value disregards too much potentially valuable neighborhood information.

Table 5.2: Cluster Similarity Metric Influence

Data	$kMeans$	$GC[H]$				$GC[S]$			
-	-	GC	mGC	wGC	$wmGC$	GC	mGC	wGC	$wmGC$
DBLP	0.4245	0.4609	0.4682	0.4597	0.4689	0.4379	0.4412	0.4379	0.4448
IMDB	0.3569	0.3809	0.3790	0.3803	0.3790	0.3948	0.4030	0.3992	0.3998
Wikipedia	0.5054	0.5497	0.5874	0.5850	0.5938	0.5261	0.5864	0.5858	0.5872

Using the term weight correction factor α helps obtaining better performance. Furthermore, above some α value (e.g., 0.7 for the IMDB and DBLP) the accuracy of the clustering is no longer sensitive to the specific choice of α, sparing us the effort of parameter fine tuning.

The cluster distance metric is a very powerful guide for the graph-based methods. Its impact is biggest when no term weight correction factor α is used or the value of α is small. This is due to the fact that α introduces new terms into each document based on the available terms in the document's neighborhood. This means that if many documents, which actually belong to different clusters, are highly connected, the presence of α introduces noisy term information into the clusters. Recall that the cluster metric Γ is based on the cosine similarity between the cluster centroids, thus with increasing α, the correcting information provided by Γ decreases its quality.

Finally, we conclude that the newly proposed graph-based clustering methods, especially the combination of the hard clustering approach applied on top of the content combination technique, are the clear winners and outperform the previously known state-of-the-art algorithms by a significant margin.

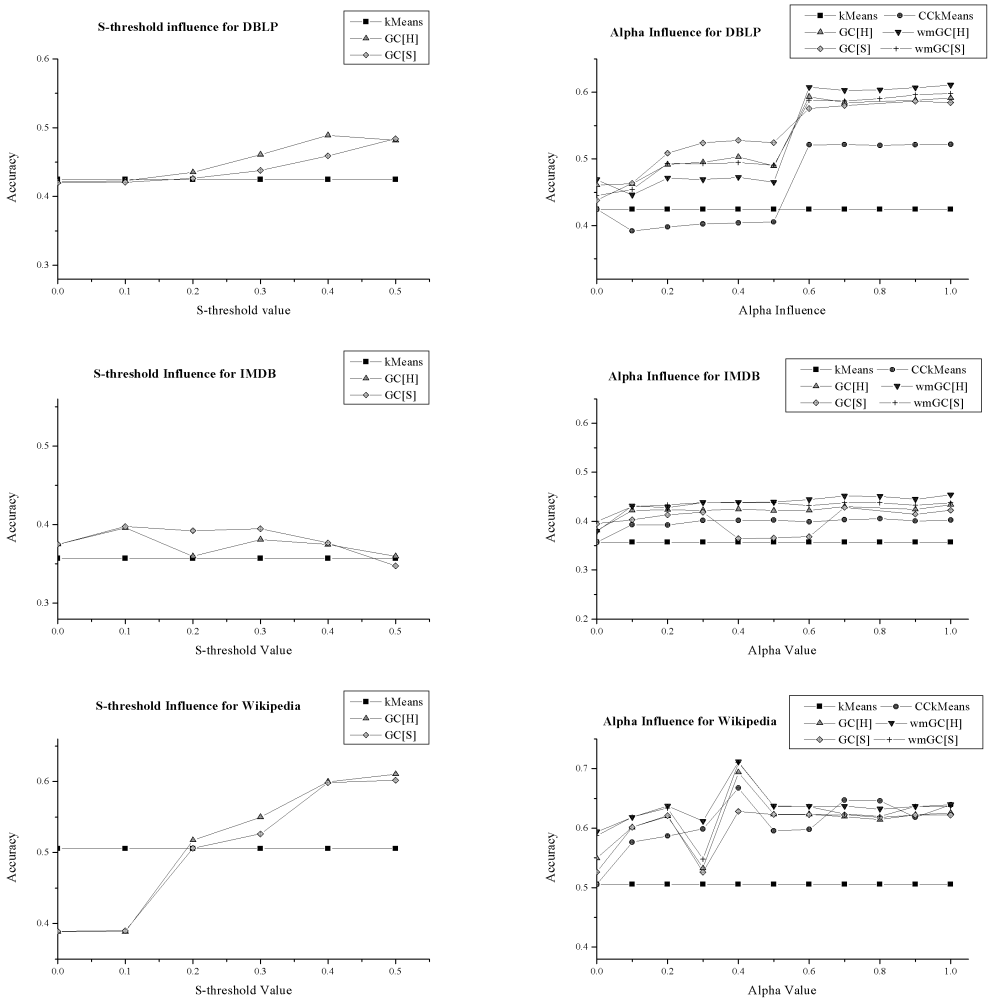


Figure 5.1: Parameter Influence for the Different Data Sets

6 Conclusion and Future Work

In this paper we proposed an approach for automatically clustering heterogeneous documents collections by using neighborhood information. The performed experiments confirm the hypothesis of highly valuable influence of the link structure over the clustering results. Our ongoing and future works includes

- the combination of our approach with orthogonal clustering approaches by using meta methods described in our work [26]
- the application of our framework on top of other clustering methods, such as density-based [10], grid-based [1] and cut-based [7] clustering
- applying the graph-based clustering method over heterogeneous graphs where nodes can have different types and each type has a specific set of possible cluster assignments.

We have experimentally shown that our neighborhood based approaches are more robust and have higher accuracy than the traditional content based approaches. A big advantage of the proposed methods is that they are suitable to be implemented as a last-stage refinement of the cluster purity produced by any clustering algorithm: content- or graph-based.

Bibliography

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD '98*, pages 94–105, 1998.

[2] R. Angelova and G. Weikum. Graph-based text classification: Learn from your neighbors. In *ACM SIGIR '06*, 2006.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD '98*, pages 307–318, 1998.

[5] R. Chellappa and A. Jain. Markov random fields: Theory and applications. 1993.

[6] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3):9–33.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[8] D. Eppstein. Finding the k shortest paths. In *IEEE Symposium on Foundations of Computer Science*, pages 154–165, 1994.

[9] M. Ester, H.-P. Kriegel, and J. Sander. *Knowledge Discovery in Databases*. Springer, 2001.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD '96*, pages 226–231, 1996.

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.

[12] J. Hartigan and M. Wong. A k-means clustering algorithm. *Applied Statistics, 28:100-108*, 1979.

[13] C.-L. Huang, T.-Y. Cheng, and C.-C. Chen. Color images' segmentation using scale space filter and markov random field. *Pattern Recognition*, 25(10):1217–1229, 1992.

[14] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[15] Z. Kato and T.-C. Pong. A markov random field image segmentation model using combined color and texture features. In *CAIP*, pages 547–554, 2001.

[16] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. In *FOCS '99*, page 14, 1999.

[17] S. Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[18] Q. Lu and L. Getoor. Link-based classification. In *ICML*, pages 496–503, 2003.

[19] W. Madison, Y. Yang, and J. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, 1997.

[20] C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[21] J. Mukherjee. Mrf clustering for segmentation of color images. *Pattern Recogn. Lett.*, 23(8):917–929, 2002.

[22] H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext catergorization method using links and incrementally available class information. In *ACM SIGIR '00*, pages 264–271, 2000.

[23] L. Pelkowitz. A continuous relaxation labeling algorithm for markov random fields. 20:709–715, 1990.

[24] M. Porter. An algorithm for suffix stripping. *Automated Library and Information Systems*, 14(3).

[25] A. Schenker, H. Bunke, M. Last, and A. Kandel. Graph-theoretic techniques for web content mining. *Series in Machine Perception and Artificial Intelligence*, 62, 2005.

[26] S. Siersdorfer and S. Sizov. Restrictive clustering and metaclustering for self-organizing document collections. In *ACM SIGIR '04*, pages 226–233, 2004.

[27] S. Theodoridis and K. Koutroumbas. Pattern recognition. *Academic Press*, 1999.

[28] Y. Yang and O. Pedersen. A comparative study on feature selection in text categorization. *ICML*, 1997.

[29] C. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comp.*, C20:68–86, 1971.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-Aware Global df Estimation in Distributed Information Retrieval Systems
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated Retraining Methods for Document Classification and their Parameter Tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An Emperical Model for Heterogeneous Translucent Objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric Calibration of High Dynamic Range Cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A.,. Magnor, H. Seidel	Joint Motion and Reflectance Capture for Creating Relightable 3D Videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and Design of Discrete Normals and Curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse Meshing of Uncertain and Noisy Surface Scattered Data
MPI-I-2005-4-001	M. Fuchs, V. Blanz, H. Lensch, H. Seidel	Reflectance from Images: A Model-Based Approach for Human Faces
MPI-I-2005-2-004	Y. Kazakov	A Framework of Refutational Theorem Proving for Saturation-Based Decision Procedures
MPI-I-2005-2-003	H.d. Nivelle	Using Resolution as a Decision Procedure
MPI-I-2005-2-002	P. Maier, W. Charatonik, L. Georgieva	Bounded Model Checking of Pointer Programs
MPI-I-2005-2-001	J. Hoffmann, C. Gomes, B. Selman	Bottleneck Behavior in CNF Formulas
MPI-I-2005-1-008	D. Michail	?
MPI-I-2005-1-007	I. Katriel, M. Kutz	A Faster Algorithm for Computing a Longest Common Increasing Subsequence
MPI-I-2005-1-003	S. Baswana, K. Telikepalli	Improved Algorithms for All-Pairs Approximate Shortest Paths in Weighted Graphs
MPI-I-2005-1-002	I. Katriel, M. Kutz, M. Skutella	Reachability Substitutes for Planar Digraphs
MPI-I-2005-1-001	D. Michail	Rank-Maximal through Maximum Weight Matchings
MPI-I-2004-NWG3-001	M. Magnor	Axisymmetric Reconstruction and 3D Visualization of Bipolar Planetary Nebulae
MPI-I-2004-NWG1-001	B. Blanchet	Automatic Proof of Strong Secrecy for Security Protocols
MPI-I-2004-5-001	S. Siersdorfer, S. Sizov, G. Weikum	Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning
MPI-I-2004-4-006	K. Dmitriev, V. Havran, H. Seidel	Faster Ray Tracing with SIMD Shaft Culling
MPI-I-2004-4-005	I.P. Ivrissimtzis, W.-. Jeong, S. Lee, Y.a. Lee, H.-. Seidel	Neural Meshes: Surface Reconstruction with a Learning Algorithm
MPI-I-2004-4-004	R. Zayer, C. Rössl, H. Seidel	r-Adaptive Parameterization of Surfaces
MPI-I-2004-4-003	Y. Ohtake, A. Belyaev, H. Seidel	3D Scattered Data Interpolation and Approximation with Multilevel Compactly Supported RBFs

MPI-I-2004-4-002	Y. Ohtake, A. Belyaev, H. Seidel	Quadric-Based Mesh Reconstruction from Scattered Data
MPI-I-2004-4-001	J. Haber, C. Schmitt, M. Koster, H. Seidel	Modeling Hair using a Wisp Hair Model
MPI-I-2004-2-007	S. Wagner	Summaries for While Programs with Recursion
MPI-I-2004-2-002	P. Maier	Intuitionistic LTL and a New Characterization of Safety and Liveness
MPI-I-2004-2-001	H. de Nivelle, Y. Kazakov	Resolution Decision Procedures for the Guarded Fragment with Transitive Guards
MPI-I-2004-1-006	L.S. Chandran, N. Sivadasan	On the Hadwiger's Conjecture for Graph Products
MPI-I-2004-1-005	S. Schmitt, L. Fousse	A comparison of polynomial evaluation schemes
MPI-I-2004-1-004	N. Sivadasan, P. Sanders, M. Skutella	Online Scheduling with Bounded Migration
MPI-I-2004-1-003	I. Katriel	On Algorithms for Online Topological Ordering and Sorting
MPI-I-2004-1-002	P. Sanders, S. Pettie	A Simpler Linear Time 2/3 - epsilon Approximation for Maximum Weight Matching
MPI-I-2004-1-001	N. Beldiceanu, I. Katriel, S. Thiel	Filtering algorithms for the Same and UsedBy constraints
MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nürnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nürnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-004	A. Podelski, A. Rybalchenko	Software Model Checking of Liveness Properties via Transition Invariants
MPI-I-2003-2-003	Y. Kazakov, H. de Nivelle	Subsumption of concepts in DL $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-018	G. Schaefer	A Note on the Smoothed Complexity of the Single-Source Shortest Path Problem
MPI-I-2003-1-017	G. Schäfer, S. Leonardi	Cross-Monotonic Cost Sharing Methods for Connected Facility Location Games
MPI-I-2003-1-016	G. Schäfer, N. Sivadasan	Topology Matters: Smoothed Competitive Analysis of Metrical Task Systems
MPI-I-2003-1-015	A. Kovács	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schäfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm