Grouping Web Pages about Persons and Organizations for Information Extraction

Shiren Ye, Tat-seng Chua, Jimin Liu, Jeremy R. Kei

School of Computing, National University of Singapore, Singapore, 117543
{yesr, chuats, liujm, jkei}@comp.nus.edu.sg

Abstract. Information extraction on the Web permits users to retrieve specific information related to the query especially on the name of a person or organization. As name is non-unique, the same name may be mapped to multiple entities. The aim of this paper is to describe an algorithm to cluster the Web pages returned by the search engine so that pages belonging to different entities are clustered into different group. The algorithm uses named entities as the features to divide the document set into direct or indirect pages. It then uses distinct direct pages as seeds of clusters to group indirect pages into different clusters. The algorithm has been found to be effective for Web-based applications.

1 Introduction

Information Extraction (IE) is a hot area of research. As defined in the Message Understanding Conference [1], IE involves the extraction of named entities (NEs), scenarios, and events from input documents. The application of effective IE technologies on the Web enables user queries to return not just documents as is currently done by the search engine, but more specific information related to the query. For example, a query on a person name on the Web should return a summary of all information related to the person, rather than a ranked list of Web pages containing one or more words in that person's name. Through the IE systen, the user will submit a name of a person or organization as query, and the system will search the Web, collect all relevant Web pages, and extract a summary of desired information to the user.

Figure 1 shows the overall process of information extraction on the web. As the name is typically non-unique, the same name may be mapped to multiple persons or organizations. To resolve this problem, we need to perform clustering of the Web pages returned so that pages belong to different entities (person or organization) are clustered into different groups. We can then concentrate on extracting information relating to a specific entity from each cluster. Search engine incorporating clustering can also return clusters to facilitate users in browsing the results.

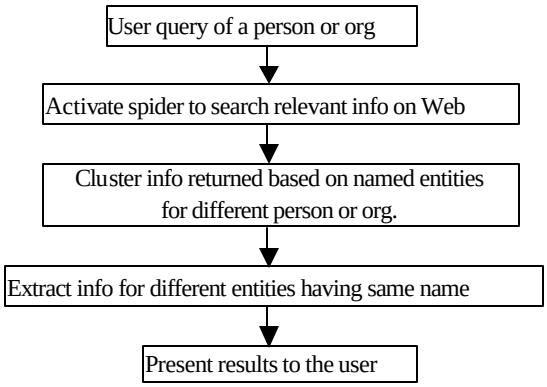


Fig. 1. The process of a Web-based information extraction

This paper presents our work on clustering Web pages on person and organization (PnO) names to support the overall information extraction procession. The aim is to cluster pages belonging to different entities (person or organization) into different clusters. We employ our tools to identify named entities in all the returned pages. We then use a combination of named entities, URL, and links as the features to perform the clustering. Our testing indicates that it is effective. The main contribution of this research is in providing an effective clustering tool for PnO pages. To the best of our knowledge, there is no other related work on this topic.

In section 2 we discuss the issues in clustering Web pages on PnO. Section 3 discusses named entity based document features; section 4 presents the algorithm to identify seeds of clusters. The method of delivering indirect pages into clusters is depicted in section 5. The results of our experiments and conclusion are respectively, contained in section 6 and section 7.

2 Clustering of Web Pages

Document Clustering algorithms attempt to identify groups of documents that are more similar to each other than the rest of the collection. In a typical document clustering algorithm, each document is represented as a weighted attribute vector, with each word in the entire document collection being an attribute in this vector (see *vector-space model* [2]). Besides probabilistic technique such as Bayesian, a priori knowledge for defining the distance or similarity measures among them are used to compare similarities between documents.

Information foraging theory [3] notes that there is a trade-off between the value of information and the time spent in finding it. The vast quantity of Web pages returned as the result of a search means that some form of clustering or summarization of the results is essential. Several new approaches have emerged to group or cluster Web pages. Those include association rule hyper-graph partitioning, principal direction di-

visive partitioning [4], Suffix Tree Clustering (STC) [5]. Scatter/Gather [6] (www.parc.xerox.com) cluster text documents according to their similarities and automatically compute an overview of the documents in each cluster. Unfortunately, most search engines at present do not use clustering as a regular procedure during Information Retrieval.

PnOs are common query requests during Internet surfing. Users frequently submit person or organization name to search engines which return results that are usually quite good and normally include the target among the top ranked results. However, there are still many problems with the search results as outlined below.

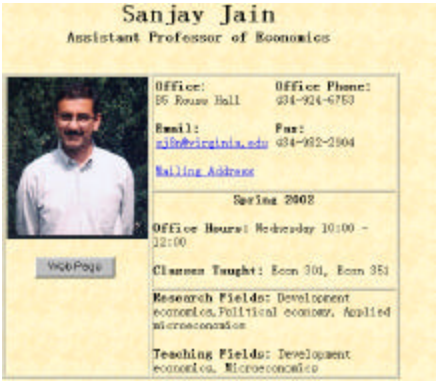
- Most users only have patience to browse only 10-20 pages. The number of pages returned could reach thousands.
- Search results may contain several persons and/or organizations whose names are the same as the query string. If the search results could be grouped into different clusters, where the pages about different entities are grouped separately, then the users could concentrate their interesting on more promising clusters.
- Some useless pages are completely irrelevant but are displayed nonetheless as return results because they contain phrases which are similar to the name of person(s) or organization(s) requested. For example, a fable page or AI research page may appear in query of "Oracle", when the user is really only trying to find information about the software company "Oracle Co.".
- The low-ranking pages listed in the rear of a result list may often be of only minor importance or could even be just tangentially related. However, they are not always useless. In some cases, novel or unexpectedly valuable results could be found in these pages. For instance, a report of a company involving in a fraud may be ranked at the bottom of thousands of returning pages, but pages such as this may be significant to users in correctly evaluating the worthiness of the company since only the most readily available or obvious information about the company will be insufficient.

As shown in Figure 2, when we submit the query "*Sanjay Jain*" to Google, at least ten persons named "*Sanjay Jain*" will be returned. Here, pages (a) and (b) are the homepages about two different persons. Page (c) is an introduction of a book authored by the person in page (a). Page (d) is the description of another person, but its style is different from that of pages (a) and (b). It can be seen that the search engine returns a great variety of both correct and incorrect results. If we are able to identify and partition the corresponding results into clusters (in this example, into three clusters for three different persons), it will facilitate users in browsing the results.

3 Document Features Based on Named Entities

In most clustering approaches, similarity (distance) between a pair of documents is computed as the cosine of the angle between the corresponding vectors in the feature space [7]. The feature vectors should be scaled to avoid skewing the result by different documents lengths or possibly by how common a word is across many documents. Many techniques such as TFIDF and stop word list [7] have been proposed for these problems. However, they do not work well for PnOs. For instance, there are two re-

sume pages about different persons. It is highly possible that they are grouped into one because they have many similar words and phrases, such as graduate, university, work, degree, join, employment, department and so on. This is especially so when their style, pattern and glossary are also similar. On the other hand, it is difficult to group a news page and resume page together even though the same person is mentioned in these pages. This is because the length and glossary are widely divergent among them. To solve this problem, it is essential that the right set of features be used to identify pages about PnO.



a. http://www.comp.nus.edu.sg/~sanjay/ b.http://www.virginia.edu/~econ/jainx.htm

c. http://mitpress.mit.edu/catalog/item /default.asp?ttype=2&tid=7277

d. http://www.bizjournals.com/stlouis /stories/2000/01/31/focus37.html

Fig. 2. Typical pages when " Sanjay Jain" is submitted to Google

In general, we observe that the occurences of PnO related named entities (NEs) in the web pages about PnOs is higher than that in the other type of pages. Here, PnO NEs include person, location and organization name, time and date, fax/phone number, currency, percentage, e-mail, URL, Link and so on. For simplicity, we called these entities collectively as NEs. We could therefore use NEs as the basis to identity

PnO pages. To support our claims, we collect and analyze 1,000 pages for PnOs and 1,000 pages for others from the Web. We found that the percentage of NEs in PnO pages is at least 6 times higher than that in other pages, if we ignore NEs of type number and percentage.

The finding is quite consistent with intuition, as NEs play important roles in semantic expression and could be used to reflect content of the pages, especially when human activities are depicted. The typical number of NEs appearing in the results of a search is around only hundreds or thousands, which means that it is feasible to use them as the features of search results about PnOs. Our analysis also shows that NEs is good in partitioning pages belonging to different persons or organizations, and the use of frequent phrases and words, such as degree, education, work etc, is not effective for this task.

NE recognition is a fundamental step to many natural language processing tasks. It was a basic task of the Message Understanding Conference (MUC) [1] and has been studied intensively. Recent research on NE recognition has been focused on the machine learning approach, such as hidden Markov model [8], decision tree [9], collocation statistics [10], and maximum entropy model [11]. As reported in MUC-7 [1], the accuracy and recall of NE recognition in English is beyond 95% and is as good as what human experts could achieve. The best reported results in NE recognition in Chinese is also beyond 90% [12]. Thus we could accurately extract NEs from the pages and then use them to reflect the content of those pages. In our system, we use the decision tree to detect English NEs and the Rationality computation and default decision tree [12][13] to detect Chinese NEs.

4 Identifying Seeds of Clusters

It is generally observed that there are two types of pages in the results returned by the search engine when the users querying on PnOs. The content of the first type of pages is almost entirely about the users' focus. Examples of such pages include the homepages, profiles, resumes, biographies and memoirs etc of PnOs. These pages contain a large number of NEs, such as graduation schools, contact information (phone, fax, e-mail, address), working organizations and experiences (time and organizations). We call such pages **Direct Pages**. The second type of pages is the **Indirect Pages**, where the concept related to the user's query string is just mentioned briefly. For instance, the person's name may appear in a page about the attendees of a conference, staff of a company, record of a transaction, or the homepage of his friend.

Since a large proportion of content of direct pages depicts items related to the query string, the relevance between direct pages and query is larger than that between the query and the indirect pages. Direct pages could provide more information than the indirect pages that satisfies the users' need. Therefore, we should choose the direct pages as the candidates of clusters' seeds. Of course, if there is more than one direct page about a target entity, then only the best one is selected as the seed for clustering.

To select the best direct pages of a target entity, we need to solve two problems. First we must be able to identify a direct page from the indirect pages. Second, in the case of multiple direct pages for the same target entity, we must be able to select the

best one. In this paper, we employ the following measures to identify the direct pages that can be used as the seeds for target entities. We observe that the number and percentage of NEs in the direct pages are much larger than those in the indirect pages do. Suppose that the number of NEs is N_{NE}, the number of tokens in pages is N_{token}. The percentage of NEs of a page is

$$f=N_{NE}/N_{token} \quad (1)$$

In our experiment, NEs within the HTML tags are not accounted into N_{token}. We use a measure that combines N_{NE} and f to provide a balance between both quantities to identify direct pages as

$$q=N_{NE}*f=N_{NE}^2/N_{token} \quad (2)$$

For example, if there are 7 NEs in a 100-token page, then q=7*7/100=0.49. The page is considered a direct page if θ is larger than a threshold τ_1.

Next, if there is more than one direct page found for a target entity, we need to find the best candidate as the seed. We observe that if both the homepage and resume of *John Smith* are selected as direct page, those two pages will share many similar NEs related to John Smith, such as the university that he graduated. Thus we could evaluate the similarity between two direct pages by examine their overlaps in instances of unique NEs. Here we use TFIDF to estimate the weight of each unique NEs as follows.

$$W_{i,j}=tf_{i,j}*\log(N/df_i) \quad (3)$$

where $tf_{i,j}$ is number of NE i in page j; df_i is the number of pages containing NE i; and N is the number of pages.

The similarity of direct page p_i and p_j could be expressed by their cosine distance as.

$$sim(p_i,p_j)=\frac{\sum_k (w_{k,i}^c * w_{k,j})}{\sqrt{\sum_k (w_{k,i})^2 * \sum_k (w_{k,j}^c)^2}} \quad (4)$$

If $sim(p_i,p_j)$ is larger than a pre-defined threshold t_3, then p_i and p_j are considered to be similar. The page that has more NEs will be used as the seed and the other will be removed. Because the number of direct pages is a small fraction of the search results, and the number of NEs in direct pages is usually less than hundreds, thus the computational cost in eliminating redundant direct pages is acceptable.

Third, because not all pages with high number of NEs, like member list of a conferences and stock price lists etc, are not direct pages. We should further check the roles of target entities those appear in the query in the text. In general, if the target entity appears in important locations, such as in HTML tags <title>, <H1> and <H2>, or it appears frequently, then the corresponding pages should be really direct pages and their topic is about the users' target. We could detect the trace of page topic using technology like wrapper rules [14].

According to the above discussion, the procedure to identify seeds of clusters is summarized as following:

```
Detect_seed(page_set)
{
  set seed_set=null;//the collection of candidate seeds
  for each (page in page_set){
    sum up N_token, N_NE and N_topic;
    //where N_topic is number of query strings
    //appearing in a page
    q= N_NE^2/N_token;
    if (q>t1 && (N_topic > t2 ||query_string is
      in title)) add page from page_set into seed_set;
  }
  for each pair p_i, p_j in seed_set:
    if (Sim(P_i, P_j)> t3){
      if (N_NE in p_i >N_NE in P_j)
        move p_j from seed_set into page_set;
      else
        move p_i from seed_set into page_set;
    }
  return seed_set;
}
```

At the end of the process, the pages remaining in the collection seed_set could be used as the seeds for clusters---they are representatives of entities named in the query string. The titles of seeds could be regarded as labels of the clusters. The NEs in titles or heads could be used as alternatives to labels.

5 Delivering Indirect pages to Clusters

Compared to direct pages, indirect pages provide less information about the target entity. Nevertheless, it does not mean that they are less important. Actually, the information extracted from indirect page may be more novel and provide more valuable information to the users. For example, your classmate may list your name in his homepage, though you have not contacted him for many years and do not know of such a page in Web. You must be surprised to find this page and feel that it is very useful. Generally, indirect page could:

- Provide additional information such as activity or experience of the target entity.
- Support or oppose the content in direct page whether they are consistent or not.
- Provide critical or negative content which may not appear in the direct page. It thus provides important information to evaluate the target entities fairly and integrality.

Therefore, we must explore an approach to link direct pages and indirect pages properly. In other words, we want to add indirect pages into clusters which are created by the seeds (direct pages). As mentioned in the above example, some of the NEs (such as your name, graduate school, or even period and degree) in your classmate's homepage is similar to those in your direct page. We can thus use the similarity be-

tween these NEs to link them together. In other words, we could compute their similarity based on a selected set of NE features.

Besides NEs in pages, URL and links in pages could also be used as heuristics to select and rank indirect pages with respect to a seed page. If the roots of URLs are same (such as www.comp.nus.edu and www.comp.nus.edu/~pris), or components of URLs are similar (such as www.nus.edu.sg and www.comp.nus.edu.sg), there should have some associations among them. Similarly, if there is a link between the direct page and the indirect page, they should not be separated. To avoid complicating the question, we suppose that URL, links and NEs have same weight, namely, URL and links are regarded as other types of NEs.

We use the algorithm below to select and link indirect pages to a seed page.

```
Arrange_indirect_page(page_set,cluster_set)
//clusters are presented by their seeds
{
  set unknown_set=null; //collection of useless pages
  foreach (page_i in page_set)
  {
     j = argmax_j sim(page_i, seed_j)  //see equation 4
    if (j>t4)
      add pagei into cluster_j;
    else
      add page_i into unknown_set;
  }
}
```

6 Experiments and Discussion

Grouping web pages about PnOs is a pivotal component in our Web-based Information Extraction System (see Figure 1) [15]. Here search results about PnOs are segmented into different clusters according to their target entities, and then the pages in different clusters are used to fill in different templates which are related to different entities.

Experiment of web information processing is a time-consuming task, where each search typically returns hundreds, or even thousands of pages. Moreover, evaluating the effectiveness of clustering is notorious even though there are many guidelines such as entropy [5], clustering error [16], and average precision [6] to measure the quality of clustering. We obtained the primary results according to following steps. Because of lack of comparable results and standard test data, we just provide our preliminery results.

a) We collect the names of 30 persons and 30 names of organization (such as companies, governments and schools) in English from Yahoo (www.yahoo.com), MSN (www.msn. com). We control PnOs that belong to large companies and famous persons (such as *Microsoft or George W. Bush*), since there would be too many pages in the search results. For example, Google returns 2,880,000 pages for

Microsoft, and first hundreds of pages are about only one special target. To ensure sufficient data for the analysis, we also excluded those PnOs whose returning pages are less than 30.

b) We use every PnO name in the above collection as query string and submit to Google. The results of the searches are downloaded. If the returning pages are more than 1,000, we downloaded just the first 1,000 pages. We also filter out the files whose formats are not HTML and plain, such as PDF, PS, PPT formats, and those whose lengths are less than 100 or more than 10,000. The average number of pages returned is 227.

c) NEs are detected from the downloaded Web pages. We remove the numbers that are used to list the items in the pages. We, however, include e-mail addresses and telephone number as a part of NEs. The average number of NEs for each page is 15.78.

d) Cluster seeds are then detected from each set of search results. The number of clusters depends on the parameters used in the algorithm detect_seed. If t_1 and t_2 are smaller, the algorithm will produce more candidates of seeds. However, most of them will be removed during the step of eliminating redundancy. In our experiment, the candidates vary from 1 to 30s, where t_1 is set to 0.64 and t_2 to 5. The seeds that detect_seed outputs vary from 0 to 11. The average number of seeds is 3.5. The number of seeds for person is larger than that for organization. This means that the number of persons with the same name is larger than that of organizations.

e) Indirect pages are added into different clusters. The average number of indirect pages in each cluster is 38. The indirect pages in each cluster about organizations are considerably more than that about persons.

f) The quality of seeds is pivotal because it controls the distribution of segmentation. Missing a seed will entail some indirect pages being assigned into wrong or unknown-set and a cluster missing. If there are redundant seeds, the direct pages about the same target may be delivered into different clusters. Fortunately, it is quite easy to differentiate between direct pages and indirect pages by using our algorithm.

The detailed performance of detecting seeds is shown in Table 1. Here the missing is the number of direct pages which should not be removed from seed-set. The average number of clusters for persons and organization is 4.57 and 2.17 respectively. Precision is correct / (correct + incorrect + redundant); recall is correct / (correct + missing + redundant + incorrect). The performance of assigning indirect pages to clusters is shown in Table 2. There are only more than 50% indirect pages could be delivered to clusters and the other are discarded into unknown set. The latter are dispersed pages or lack the evidence to group their seeds. Because the number of indirect pages is large, we do not check the quality of their delivering, such as missing and incorrect. We will focus this in the further research.

We evaluated the performance of our clustering approach according to two aspects. One important factor is the quality of seeds. Missing a seed will reduce the number of clusters. While conserving a redundant seed will incur the pages about the same target entity scattering in more than one clusters. As shown in Table 1, the average ratio of the missing clusters and the redundant clusters is lower than 10%, which indicates

that the seeds are stable and reliable. The other factor is the quality of entire clusters, as listed in Table 2. There are about 40% of pages that cannot be clustered into exact targets and are assigned to unknown-set. This may be caused by missing heuristic information, when some target entities do not have direct pages and the contents related to them are sparse in the Web.

Table 1. the performance of detecting seeds or direct pages for distinct target entities.

Seeds	*total*	*correct*	*incorrect*	*missing*	*redundant*	*precision*	*recall*
Person	137	127	8	3	2	92.7%	94%
Org.	65	61	4	3	4	84.7%	89.7%
Overall	202	188	12	6	6	88.7%	91.8%

Table 2. the performance of assigning indirect pages.

page	*total num.*	*avg. num. in clusters*	*unknown*	*ratio of delivering*
Person	3,600s	70s(*30)	1,500s	58.3%
Org.	9,800s	220s (*30)	3,200s	67.3%

The performance of this PnOs grouping approach is discussed as following:

- **Effectiveness**: Compared with the results of manual clustering, outputs of computer do not always match human expectation[17]. Different people may produce different segmentations, and the case by computers is worse. For many clustering, though we could find out the reason for such segmentation, some of them still are specious. However, our approach is natural and its measurement is clear and relatively objective: pages about the same target should be grouped together. The seeds of clusters (direct pages) cover many items of targets and the relevance between direct page and query is highest. According to this measurement, there is little variable when search results are segmented. Hence, users could comprehend our clusters very well and accept it. Of course, once errors occur, especially when pages about different targets are arranged together, they could figure them out explicitly. In spite of that, the extensibility of this clustering approach is limited. For pages beyond PnOs, such as research paper, financial report, etc., the frequency of NEs is much lower. If there were represented by NEs, only few features should be non-zero, most of them could not be grouped together. At the same time, it is difficult to distinguish direct pages from indirect pages, so the number of clusters is unstable. On the contrary, for PnOs, there is obvious difference among them and detecting seeds is not sensitive to parameters. In a word, the testing users are satisfied with the segmentation by our approach.
- **Snippet or document**: If the clustering result on snippets returned by search engine is as good as that on document, we could save quite a little time and cost for downloading original pages. Like most clustering algorithm, our approach is sensitive to length of source and improper for snippets. This may be caused by snippets being short of NEs and fake NEs emerging. For example, if there is string "*…University of Singapore*" in the boundary of a snippet, a NE *University of Singapore* will be detected. In fact, it should be *National University of Singapore*.
- **Computational cost**: Account for response to users in time, the speed of clustering Web pages is pivotal. The computing time of our approach is nearly linear to the

number of Web pages, the number of NEs among them (including URL and links) and the number of clusters, when the number of clusters is further less than that of pages. Furthermore, NEs could be recognized in the procedure of downloading pages. Determining direct page or indirect page could be proceeded at the same time if NEs are not normalized. In our experiment, the average time for clustering 1000 pages about PnOs using non-optimal algorithm is 34 seconds (PIII 900 and Men 512M).

7 Conclusion

PnOs are common query questions during users surfing Internet. Our clustering approach based on NEs could be used to group search results according to their targets where distinctive direct pages play the seeds of clusters. Although it is difficult to extend this approach to domain besides PnOs, it is certainly an effective approach for users to summarize information about special target, and track the activity of entities in which they are interested.

Further research is centered on: (a) Improve the effectiveness of the clustering method. (b) Incorporate learning to tune the parameters of the clustering approach. (c) Use the clustering results to perform information extraction for PnOs, and to facilitate user browsing. (d) Extract the techniques to extract information in other domains.

References

[1] Elaine Marsh, Dennis Perzanowski, MUC-7 Evaluation of IE Technology: Overview of Result, (1998), http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

[2] G. Salton, Automatic Text Processing. Addison-Wesley, New York, (1989)

[3] Pirolli, P. & Card, S. Information foraging in information access environments. In: Proc. of the Conf. on Human Factors in Computing Sys., (1995) 51-58

[4] Daniel Boley, et al, Partitioning-based Clustering for Web Document Categorization, in: Decision Support System 27, (1999) 329-341

[5] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In: Proc. ACM SIGIR'98, (1998) 46-54

[6] Oren Zamir, Oren Etzioni, Grouper: a dynamic clustering interface to Web search results, in: Computer Networks l31, (1999)1361-174

[7] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, (1983)

[8] Bikel D.M., Schwartz R. & Weischedel R.M. An Algorithm that Learns What's in a Name, in: *Machine Learning*, 34(1-3), (1999)211-231

[9] Sekine S. NYU: Description of The Japanese NE System Used for MET-2, in: MUC-7, (1998)

[10] Lin D. Using collocation statistics in information extraction, in: MUC-7, (1998)

[11] Borthwick A. *A Maximum Entropy Approach to Named Entity Recognition.* Ph.D. Thesis, New York Univ. (1999)

[12] Shiren Ye, Tat-seng Chua, Jimin Liu, Learning Pattern Rules for Chinese Named Entity Extraction, COLING 02, Taiwan, (2002)
[13] Tat-seng, Jimin Liu, Learning Pattern Rules for Chinese Named Entity Extraction, AAAI02, Canada (2002)
[14] Mark Craven, Dan DiPasquo, et al, Learning to Extract Symbolic Knowledge from the WWW, in: Proc. of AAAI-98, Madison, USA, (1998) 509-516
[15] http://www.comp.nus.edu.sg/~pris, (2002)
[16] Dmitri Roussinov and J. Leon Zhao, Automatic Discovery of Similarity Relationships through Web Mining, in: Decision Support Systems: Special Issue on Web Retrieval and Mining, (2002)
[17] D. Roussinov, H. Chen, Document Clustering for Electronic Meetings: An Experimental Comparision of Two Techniques, in: Decision Support Sys., (27)1-2, (1999) 67-79