ISSN 23499842(Online), Volume 1, Special Issue 1(NCFCSIT 15), March 2015
International Journal of Innovative Trends and Emerging Technologies

THE MINING OF TEXT AND IMAGE DATA USING SIDE INFORMATION

*Priya Dharshini K, *Lakshmi.S.P, *UG STUDENTS
** Ms.S.Divya, **ASSISTANT PROFESSOR
Department of Computer Science and Engineering
Dhanalakshmi College of Engineering, Chennai
pridharsh93@gmail.com
lakshmisubramanian.cse@gmail.com
divyasrini0@gmail.com

***Abstract*— In most of the mining applications may be a text mining or image mining the content contains some side information along with the file content. Those side-information might be of various kinds, such as document provenance information, the links in the document, user-access behavior from web logs, or other non-textual attributes which are embedded into the text document where as in image it might be the color, shape, size, pixels or other image oriented features. Such attributes may contain a tremendous amount of information for clustering purposes. However, the relative importance of this side-information may be a risk to estimate the features, especially when some of the information is noisy. In such cases, it can be risky to incorporate side-information into the mining process, because it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, we need a principled way to perform the mining process, so as to maximize the advantages from using this side information. In this paper, we design an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach. Image data mining can be done manually by slicing and dicing data or it can be done with programs that analyze the data automatically. Color, texture and shape of an image have been primitive image descriptors in Content Based Image Retrieval (CBIR) system. We also will present experimental results on a number of real data sets in order to illustrate the advantages of using such an approach.**

Keywords: Side Information, Metadata, Clustering, data sets.

I. INTRODUCTION

Data Mining is the process of exploration and analysis, by automatic or semi-automatic, of large amounts of data to discover meaningful patterns and rules. It is used in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. Most often the problem of managing large data is done using the Clustering algorithm. The major problems in maintaining and retrieving data occur in application domain such as web browsers, social network and other digital collections. This happens mainly because of the absence of other attributes. A lot of side-information is available along with the text documents.

Such side-information may be of different kinds, such as the links in the document, user-access behavior from web logs, or other non-textual attributes which are embedded into the text document. Such attributes may contain a tremendous amount of information for clustering purposes. The basic side information available in the text are as follows,

- Web log: In many application, the behavior of the user is been registered in a log that is called as user log. This user log serves a side information to enhance the quality of the mining process.
- Links: Nowadays many documents contains links

in between them. The links includes the meaning of the context or it might lead to another page. These links might also serve as side information.

Other side information such as metadata (information about data) and other attributes contributes a lot in clustering. The image side information includes color, shape, pixel and shades of the image. The side information might be sometimes associated noisy corrupts which may worsen the quality of the mining process in many ways.

In order to achieve a good clustering, the side information and the attributes must provide similar hints to perform the clustering operations.

II. PROBLEM STATEMENT

Given large data sets with hundreds of thousands or millions of entries, computing all pairwise similarities between objects is often intractable, and more efficient methods are called for. Also, increasingly, people are trying to fit complex models such as mixture distributions or HMMs to these large data sets. Computing global models where all observations can affect all parameters is again intractable, and methods for grouping observations (similarly to the grouping of objects above) are

needed. The problem of text clustering arises in the context of many application domains such as the web, social networks, and other digital collections. This method of processing the side information may lead to inconvenience in extracting the complete data. The need of similar side information and attributes for the clustering process is a complex thing to calculate exactly. And also the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy.

MERITS AND DEMERITS

1. Side information such as pairwise constraints is useful to improve the clustering performance in general. However constraints are not always error free.

2. When erroneous constraints are specified as side information, treating them as hard constraints could have the disadvantages since strengthening incorrect or erroneous constraint may lead to performance degradation.

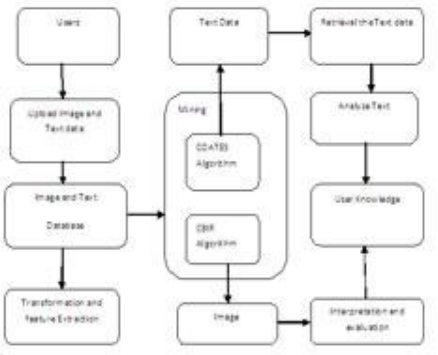

Fig 1: System Architecture

III. PROPOSED SYSTEM

In this project, we will discuss the efficient way that is used to retrieve the text and the image data extraction. The clustering of text is done by using the algorithm "Content and Auxiliary based Text Clustering" (COATES) Algorithm and the image is extracted using the "Content Based Image Retrieval" (CBIR) algorithm. At first the user have to login to upload the file that has to be retrieved by another person. The users who had registered are only allowed to upload data and they are only allowed to download the data. The user id is used to identify that particular content, in other words each text and image must be uploaded with an unique user id. The user id here serves as a primary key to identify the content that has to be retrieved. The user on the other end, when he opened the file, the file might be corrupted due to the noisy side information. At this time our algorithms are used to cluster the file in such a way excluding the noise. After clustering, it finds a centroid for each cluster. This process is done both in the received file and the file that is been stored in the web database. Now these two centroids are compared and found the difference between them. This difference is then applied in K-means clustering algorithm for calculation. This process is proceeded as a loop until the difference becomes zero. Then the result is been displayed on the user screen.

MERITS AND DEMERITS

1. A probabilistic model on the side information uses the partitioning information for the purpose of estimating the coherence of different clusters with side attributes. This helps in omitting out the noise in the membership behavior of different attributes.

2. By using the K-means clustering algorithm, it is easy for us to find the difference and damages that has occurred in the received file.

3. In addition it also takes care of the smoothing issues and also the time complexity that occur frequently during the data extraction.

IV. ALGORITHM

The clustering of text and the image is commonly done as follows, We assume that we have a corpus S of text documents. The total number of documents is N , and they are denoted by $T_1 \ldots T_N$. It is assumed that the set of distinct words in the entire corpus S is denoted by W. Associated with each document T_i, we have a set of side attributes $\overline{X_i}$. Each set of side attributes $\overline{X_i}$ has d dimensions, which are denoted by $(x_{i1} \ldots x_{id})$. We refer to such attributes as *auxiliary* attributes. For ease in notation and analysis, we assume that each side-attribute x_{id} is binary, though both numerical and categorical attributes can easily be converted to this format in a fairly straightforward way. This is because the different values of the categorical attribute can be assumed to be separate binary attributes, whereas numerical data can be discretized to binary values with the use of attribute ranges.

A. The COATES Algorithm:

In this section, we will describe our algorithm for text clustering that is done based on the side information. Content and Auxiliary based Text Clustering algorithm is referred as COATES algorithm in this entire paper. We assume that an input to the algorithm is the number of clusters k. Usually in all the clustering methods, that stopwords have been removed, and stemming has been performed in order to improve the discriminatory power of the attributes.

The algorithm requires two phases:

Initialisation: It is a lightweight weight process in which a standard text clustering approach is used without any side-information. The centroids and the partitioning created by the clusters formed in the initialization phase provide an initial starting point for the second phase which makes our algorithm simpler than others. This is based on text only, and does not use the auxiliary information.

Main Phase: The main phase of the algorithm is executed after the first phase. This phase starts off with these initial groups, and iteratively reconstructs these clusters with the use of both the text content and the auxiliary information. This phase performs alternating iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering. We call these iterations as content iterations and auxiliary iterations respectively. The combination of the two iterations is referred to as a major iteration.

The algorithm maintains a set of seed centroids, which are subsequently refined in the different iterations. The centroids for the k clusters created during this phase are denoted by L1 . . . Lk. We assume that the k clusters associated with the data are denoted by C1 . . . Ck. In each auxiliary phase, we create a probabilistic model, which relates the attribute probabilities to the cluster-membership probabilities. The goal of this modeling is to examine the coherence of the text clustering with the side-information attributes. We assume that each auxiliary iteration has a prior probability and a posterior probability of assignment of documents to clusters with the use of auxiliary variables in that iteration. We denote the prior probability that the document Ti belongs to the cluster Cj by P(Ti ∈ Cj) and the posterior probabilities P(Ti ∈ Cj|Xi) we use the auxiliary attributes Xi which are associated with Ti. Therefore, we would like to compute the conditional probability P(Ti ∈ Cj|Xi). Since we are focusing on sparse binary data, the value of 1 for an attribute is a much more informative event than the

Algorithm COATES(NumClusters: k, Corpus: T1 ...TN, Auxiliary Attributes: X1 ...XN);

```
begin
  Use content-based algorithm in [27] to create initial set of k
  clusters C1 ...Ck;
  Let centroids of C1 ...Ck be denoted by L1 ...Lk;
  t =1 ;
  while not(termination criterion) do
  begin
    { First minor iteration }
      Use cosine-similarity of each document Ti to centroids L1
      ...Lk in order to determine the closest cluster to Ti and
      update the cluster assignments C1 ...Ck;
    Denote assigned cluster index for document Ti by qc(i,t);
    Update cluster centroids L1 ...Lk to the centroids of updated
    clusters C1 ...Ck;
    { Second Minor Iteration }
    Compute gini-index of Gr for each auxiliary attribute r with
    respect to current clusters C1 ...Ck;
    Mark attributes with gini-index which is γ standard-
    deviations below the mean as non-discriminatory;
  { for document Ti let Ri be the set of attributes which take on
  the value of 1, and for which gini-index is discriminatory;}
      for each document Ti use the method discussed in
    section 2 to determine the posterior probability Pn(Ti ∈C
    |Ri);
        Denote qa(i,t) as the cluster-index with highest
        posterior probability of assignment for document Ti;
        Update cluster-centroids L1 ...Lk with the use of
        posterior probabilities as discussed in section 2;
      t = t +1 ;
  end
end
```

Fig. 1. The COATES Algorithm

default value of 0. Therefore, it suffices to condition only on the case of attribute values taking on the value of 1.

Furthermore, in order to ensure the robustness of the ap- proach, we need to eliminate the noisy attributes. For this, The gini-index is computed as follows. Let f_{rj} be the fraction of the records in the cluster Cj (created in the last content- based iteration), for which the attribute r takes on the value of 1. Then, we compute the relative presence p_{rj} of the attribute r in cluster j as follows:

$P_{rj} = f_{rj} \div \sum_{m=1}^{k} f_{rm}$

The values of p_{rj} are defined, so that they sum to 1 over a particular attribute r and different clusters j. We note that when all values of p_{rj} take on a similar value of 1/k, then the attribute values are evenly distributed across the different clusters. Therefore, we would like the values of p_{rj} to vary across the different clusters. We refer to this variation as skew. The level of skew can be quantified with the use of the gini-index. The gini-index of attribute r is denoted by Gr, and is defined as follows:

$Gr = \sum_{j=1}^{k} p_{rj}^2$

The value of Gr lies between 1/k and 1. The more dis- criminative the attribute, the higher the value of Gr. In each iteration, we use only the auxiliary

attributes for which the gini-index is above a particular threshold γ. The value of γ is picked to be 1.5 standard deviations below the mean value of the gini-index in that particular iteration. The total running time is given by O(N ·k·(d + dt)).

B. The CBIR Algorithm:

An image retrieval system is a system which allows us to browse, search and retrieve the images. Content Based Image Retrieval is the process of retrieving the desired image from a huge number of databases based on the contents of the image. The term "content" in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself which contributes to be the side information of the image. CBIR is desirable because searches that rely purely on metadata are dependent on annotation quality and completeness

We are going to implement the CBIR along with the K-means clustering algorithm. k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriority. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V)=\sum_{i=1}^{c}\cdot\sum_{j=1}^{ci}(||x_i-v_j||)^2$$

where,

'$||x_i - v_j||$' is the Euclidean distance between the x_i and v_j

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

Algorithmic steps for k-means clustering

Let $X = \{x_1,x_2,x_3,\ldots\ldots\ldots,x_n\}$ be the set of data points and $V = \{v_1,v_2,\ldots\ldots\ldots,v_c\}$ be the set of center.

1)Randomly select 'c' cluster centers.
2) Calculate the distance between each data point and cluster centers.
3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers. 4) Recalculate the new cluster center using:

$$V_i=(1/c_i)\sum_{j=1}^{ci} x_i$$

where, 'c_i' represents the number of data points in i^{th} cluster.

5) Recalculate the distance between each data point and new obtained cluster centers.
6) If no data point was reassigned then stop, otherwise repeat from step 3).

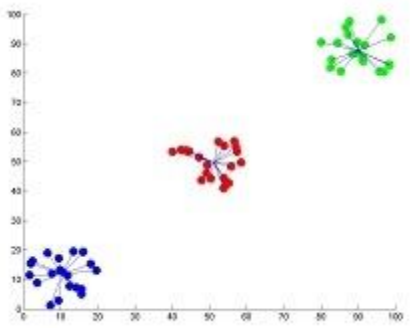

Fig I: Showing the result of k-means for *'N'* = 60 and *'c'* = 3

V. EXTENSION TO CLASSIFICATION

In this section, we will discuss how to extend the approach to classification. As before, we assume that we have a text corpus S of documents. The total number of documents in the training data is N, and they are denoted by T1 ...TN. Associated with each text document Ti, we also have a training label i, which is drawn from{1...k}. As before, we assume that the side information associated with the ith training document is denoted by Xi. It is assumed that a total of N' test documents are available, which are denoted by T'_1 ...T'_N. It is assumed that the side information associated with the ith document is denoted by X'i. We refer to our algorithm as the COLT algorithm throughout the paper, which refers to the fact that it is a COntent and auxiLiary attribute-

based Text classification algorithm. The steps used in the training algorithm are as follows:

• **Feature Selection:** In the first step, we use feature selection to remove those attributes, which are not related to the class label. This is performed both for the text attributes and the auxiliary attributes.

• **Initialization:** In this step, we use a supervised k-means approach in order to perform the initialization, with the use of purely text content. The main difference between a supervised k-means initialization, and an unsupervised initialization is that the class memberships of the records in each cluster are pure for the case of supervised initialization. Thus, the k-means clustering algorithm is modified, so that each cluster only contains records of a particular class.

• **Cluster-Training Model Construction:** In this phase, a mixture of the text and side-information is used for creating a cluster-based model. When the set of supervised clusters is been created or constructed, these are used for the purposes of classification. We will discuss each of these steps in some detail below. Next, we will describe the training process for the COLT algorithm. The first step in the training process is to create a set of supervised clusters, which are then leveraged for the classification. The first step in the supervised clustering process is to perform the feature selection, in which only the discriminative attributes are retained. Here, we compute the gini-index for each attribute in the data with respect to the class label. If the gini index is γ standard deviations (or more) below the average gini index of all attributes, then these attributes are pruned globally, and are

Algorithm COLT(NumClusters: k, Corpus: T1 ...TN, Auxiliary Attributes: X1 ...XN);

```
begin
  Use content-based algorithm in [27] to create initial set of k
  clusters C1 ...Ck;
  Let centroids of C1 ...Ck be denoted by L1 ...Lk;
  t =1 ;
  while not(termination criterion) do
  begin
    { First minor iteration }
      Use cosine-similarity of each document Ti to centroids L1
      ...Lk in order to determine the closest cluster to Ti and
      update the cluster assignments C1 ...Ck;
    Denote assigned cluster index for document Ti by qc(i,t);
    Update cluster centroids L1 ...Lk to the centroids of updated
    clusters C1 ...Ck;
    { Second Minor Iteration }
    Compute gini-index of Gr for each auxiliary attribute r with
    respect to current clusters C1 ...Ck;
    Mark attributes with gini-index which is γ standard-
    deviations below the mean as non-discriminatory;
  { for document Ti let Ri be the set of attributes which take on
  the value of 1, and for which gini-index is discriminatory;}
    for each document Ti use the method discussed in
    section 2 to determine the posterior probability Pn(Ti ∈C
    |Ri);
      Denote qa(i,t) as the cluster-index with highest
      posterior probability of assignment for document Ti;
      Update cluster-centroids L1 ...Lk with the use of
      posterior probabilities as discussed in section 2;
    t = t +1 ;
  end
end
```

Fig. 2. The COLT Training Process

Algorithm COLTClassify(Clusters: C1 ...Ck, Test Instance: T'_i, Auxiliary Attributes of Test Instance: X'_i)

```
begin
  Determine top r closest clusters in C1 ...Ck to T' i based on
  cosine similarity with the text attributes;
  Derive the set R'i from Xi, which is the set of non-zero
  attributes in X'i;
  Compute Ps(T'i ∈C j|R'i) with the use of Equation 8;
  top r Determine clusters in C1 ...Ck to X'i based on the largest
  value of Ps(T'i ∈C j|R' i);
  Determine the majority class label from the 2·r labeled
  clusters thus determined;
  return majority label;
end
```

Fig. 3. The COLT Classification Process with the use of the Supervised Clusters

never used further in the clustering process.

Then the initialization of the training procedure is performed only with the content attributes. Each cluster is associated with a particular class and all the records in the cluster belong to that class. This goal is achieved by first creating unsupervised cluster centroids, and then adding supervision to the process. Therefore, in each iteration, for a given document, its distance is computed only to clusters which have the same label as the document. The document is then assigned to that cluster. This approach is continued to convergence.

After initialization, the main process of creating supervised clusters with the use of a combination of content and auxiliary attributes is started. For the case of the auxiliary minor iteration, we compute the prior probability Pa(Ti ∈C j) and the posterior probability Ps(Ti ∈C j|Ri), as in the previous case, except that this is done only for cluster indices which belong to the same class label. The document is assigned to one of the cluster indices with the largest posterior probability.

Once the supervised clusters have been created, they can be used for the purpose of classification. . In order to perform the classification, we separately compute the r closest clusters to the test instance T'i with the use of both content and auxiliary

attributes.The time complexity to perform this classification operation is given by O(N ·k·(d + dt)).

V.EXPERIMENTAL RESULTS

Here we compare our clustering and classification methods against a number of baseline techniques on real and synthetic data sets. As the baseline, we used two different methods: (1) An efficient projection based clustering approach which adapts the k-means approach to text. (2) We adapt the kmeans approach with the use of both text and side information directly. We refer to this baseline as K-Means [text+side] in all figure legends.

For the case of the classification problem, we tested the COLT methods against the following baseline methods: (1) We tested against a Naive Bayes Classifier which uses only text. (2) We tested against an SVM classifier which uses only text. (3) We tested against a supervised clustering method which uses both text and side information. We will show that our approach has significant advantages for both the clustering and classification problems.

A. Data Sets We used three real data sets in order to test our approach. The data sets used were as follows:

(1) Cora Data Set: The Cora data set1 contains 19,396 scientific publications in the computer science domain. Each research paper in the Cora data set is classified into a topic hierarchy. On the leaf level, there are 73 classes in total. We used the second level labels in the topic hierarchy, and there are 10 class labels. We further obtained two types of side information from the data set: citation and authorship. These were used as separate attributes in order to assist in the clustering process. There are 75,021 citations and 24,961 authors. One paper has 2.58 authors in average, and there are 50,080 paper-author pairs in total.

(2) DBLP-Four-Area Data Set: The DBLP-Four-Area data set is a subset extracted from DBLP that contains four data mining related research areas, which are database, data mining, information retrieval and machine learning. This data set contains 28,702 authors, and the texts are the important terms associated with the papers that were published by these authors. There are 20 conferences in these four areas and 44,748 author-conference pairs. Besides the author conference attribute, we also used co-authorship as another type of side information, and there were 66,832 co author pairs in total.

(3) IMDB Data Set: The Internet Movie Data Base (IMDB) is an online collection2 of movie information. We obtained ten-year movie data from 1996 to 2005 from IMDB in order to perform text clustering. We extracted movies from the top four genres in IMDB which were labeled by Short, Drama, Comedy, and Documentary. We removed the movies which contain more than two above genres. There were 9,793 movies in total, which contain 1,718 movies from the Short genre, 3,359 movies from the Drama genre, 2,324 movies from the Comedy genre and 2,392 movies from the Documentary genre. The names of the directors, actors, actresses, and producers were used as categorical attributed corresponding to side information. The IMDB data set contained 14,374 movie-director pairs, 154,340 movie actor pairs, 86,465 movie-actress pairs and 36,925 movie producer pairs.

B. Evaluation Metrics The aim is to show that our approach is superior to natural clustering alternatives with the use of either pure text or with the use of both text and side information. In each data set, the class labels were given, but they were not used in the clustering process. The average cluster purity over all clusters (weighted by cluster size) was reported as a surrogate for the quality of the clustering process. Let the number of data points in the k clusters be denoted by n1 . . . nk. We denote the dominant input cluster label in the k clusters by l1 . . . lk. Let the number of data points with input cluster label li be denoted by ci. Then, the overall cluster purity P is defined by the fraction of data points in the clustering which occur as a dominant input cluster label in the k clusters by l1 . . . lk.

C. Sensitivity Analysis We also tested the sensitivity of the COATES algorithm with respect to two important parameters. We will present the sensitivity results on the Cora and DBLP-Four Area data sets. As mentioned in the algorithm in Fig. 1, we used threshold γ to select discriminative auxiliary attributes. While the default value of the parameter was chosen to be 1.5.The results are constant for both baseline methods because they do not use this parameter. It is evident from both figures that setting the threshold γ too low results in purity degradation, since the algorithm will prune the auxiliary attributes too aggressively in this case. On both data sets, the COATES algorithm achieves good purity results when γ is set to be 1.5. Further increasing the value of γ will reduce the purity slightly because setting γ too high will result in also including noisy attributes. Typically by picking γ in the range of (1.5, 2.5), the best results were observed. Therefore, the algorithm shows good performance for a fairly large range of values of γ. This suggests that the approach is quite robust.

VI. CONCLUSION AND SUMMARY

In this paper, we presented simplest ways and methods for mining image and text data with the use of side-information. Many forms of image databases and text databases contain a large amount of side-information or meta information, which may be used in order to improve the clustering process. In order to design the clustering method, here we have combined an iterative partitioning technique with a probability estimation process which computes the importance of various types of side-information. This general approach is used in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. From the results that we have shown it is clear that with the help of side-information, we can greatly enhance the quality of image and text clustering and classification, with a high level of efficiency throughout the approach.

REFERENCES

1. C. C. Aggarwal and H. Wang, Managing and Mining Graph Data Springer, 2010.
2. C. C. Aggarwal, Social Network Data Analytics, Springer, 2011.
3. C. C. Aggarwal and C.-X. Zhai, Mining Text Data, Springer, 2012
4. C. C. Aggarwal and C.-X. Zhai, "A Survey of Text Classification Algorithms," Book Chapter in Mining Text Data, Springer, 2012.
5. C. C. Aggarwal and P. S. Yu, "A Framework for Clustering Massive Text and Categorical Data Streams," in SIAM Conf. on Data Mining, pp. 477–481, 2006.
6. C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," in IEEE TKDE, vol. 16(2), pp. 245–255, 2006.
7. C. C. Aggarwal, and P. S. Yu., "On text clustering with side information,"in IEEE ICDE Conference, 2012.
8. R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in *CIKM Conf., pp. 778–* 779, 2006.
9. A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *SDM Conf., pp. 437–442,*\2007.
10. J. Chang and D. Blei, "Relational Topic Models for Document Networks, in *AISTASIS, pp. 81–88, 2009.*