

Spectral Co-Clustering for Dynamic Bipartite Graphs

Derek Greene, Pádraig Cunningham

School of Computer Science & Informatics, University College Dublin
{derek.greene,padraig.cunningham}@ucd.ie

Abstract. A common task in many domains with a temporal aspect involves identifying and tracking clusters over time. Often dynamic data will have a feature-based representation. In some cases, a direct mapping will exist for both objects and features over time. But in many scenarios, smaller subsets of objects or features alone will persist across successive time periods. To address this issue, we propose a dynamic spectral co-clustering method for simultaneously clustering objects and features over time, as represented by successive bipartite graphs. We evaluate the method on a benchmark text corpus and Web 2.0 bookmarking data.

1 Introduction

In many domains, where the data has a temporal aspect, it will be useful to analyse the formation and evolution of patterns in the data over time. For instance, researchers may be interested in tracking evolving communities of social network users, such as clusters of frequently interacting authors in the blogosphere, or circles of users with shared interests on social media sites. In the case of online news sources, producing large volumes of articles on a daily basis, it will often be useful to chart the development of individual news stories over time.

For many of these problems it may be of interest to simultaneously identify clusters of both data objects and features. This task, often referred to as *co-clustering*, has been formulated as the problem of partitioning a bipartite graph, where the two types of nodes correspond to objects and features [1]. However, to the best of our knowledge, this work has been limited to static applications, where temporal information is unavailable or has been disregarded.

A popular recent approach to the problem of clustering dynamic data has been to use an “offline” strategy, where the dynamic data is divided into discrete *time steps*. Sets of *step clusters* are identified on the individual time steps using a suitable clustering algorithm, and these step clusters are associated with one another over successive time steps [2]. However, clusters may change considerably between time steps. This can be problematic, both for the purpose of matching clusters between time steps, and for supporting users to follow and understand how groups are changing over time. To address this problem, both current and historic information can be incorporated into the objective of the

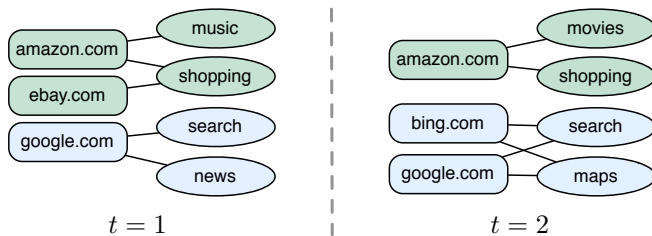


Fig. 1. A dynamic co-clustering scenario where 2 clusters appear in 2 time steps. Note that a subset of both objects (bookmarks) and features (tags) persists across time.

clustering process [3]. Benefits of this approach include increasing the smoothness of transitions between clusterings over time, and improving cluster quality by incorporating historic information to reduce the effects of noisy data.

A number of additional considerations arise when tracking dynamic data represented in feature spaces. Notably, a set of objects or features will not always persist in the data across steps. In general, three different scenarios are possible:

1. Data objects alone persist across time steps. For instance, in bibliographic networks, papers are only published at a single point in time, whereas authors will generally be present in the network over an extended period of time.
2. Features alone persist across time. In a news collection, articles will appear once, whereas terms may continue to appear as topics extend over time.
3. Both objects and features persist across time. For example, in the case of Web 2.0 tagging portals, both the individual tags and the objects being tagged (*e.g.* bookmarks, images) will appear in multiple time steps. A simple example with just two clusters is shown in Figure 1.

Here we consider the problem of tracking nodes in multiple related dynamic bipartite graphs. In Section 3 we describe the main contribution of this paper – a dynamic spectral co-clustering algorithm for simultaneously grouping objects and features over time, in any of the above scenarios. This algorithm takes into account both information from the current time step, together with historic information from the previous step. In our evaluations in Section 4 we show that the proposed algorithm works both in the case where features alone persist over time, and when objects **and** features persist. These evaluations are performed on a labelled benchmark news corpus and Web 2.0 tagging data.

2 Related Work

2.1 Co-clustering

In certain problems it may be useful to perform *co-clustering*, where both objects and features are assigned to groups simultaneously. One approach to the co-clustering problem is to view it as the task of partitioning a weighted bipartite graph. Dhillon [1] proposed a spectral approach to approximate the optimal

normalised cut of a bipartite graph, which was applied for document clustering. This involved computing a truncated singular value decomposition (SVD) of a suitably normalised term-document matrix, constructing an embedding of both terms and documents, and applying k -means to this embedding to produce a simultaneous k -way partitioning of both documents and terms. Mirzal & Furukawa [4] provided a further theoretical grounding for spectral co-clustering, demonstrating that simultaneous row and column clustering is equivalent to solving the separate row and column clustering problems.

2.2 Dynamic Clustering

The general problem of identifying clusters in dynamic data has been studied by a number of authors. Early work on the unsupervised analysis of temporal data focused on the problems of topic tracking and event detection in document collections [5]. More recently, Chakrabarti *et al.* [3] proposed a general framework for “evolutionary clustering”, where both current and historic information was incorporated into the objective of the clustering process. The authors used this to formulate dynamic variants of common agglomerative and partitional clustering algorithms. In the latter case, related clusters were tracked over time by matching similar centroids across time steps. Two evolutionary versions of spectral partitioning for classical (unipartite) graphs were proposed by Chi *et al.* [6]. The first version (PCQ) involved applying spectral clustering to produce a partition that also accurately clusters historic data. The second version (PCM) involved measuring historic quality based on the chi-square distance between current and previous partition memberships.

The application of dynamic clustering methods has been particularly prevalent in the realm of social network analysis, where the goal is to identify communities of users in dynamic networks. Palla *et al.* [7] proposed an extension of the popular CFinder algorithm to identify community-centric evolution events in dynamic graphs, based on an offline strategy. This extension involved applying community detection to composite graphs constructed from pairs of consecutive time step graphs. Another life-cycle model was proposed in [2], where the dynamic community finding approach was formulated as a graph colouring problem. The authors proposed a heuristic solution to this problem, by greedily matching pairs of node sets between time steps. The problem of clustering data over time has also been considered in the temporal analysis domain. Kalnis *et al.* [8] described a density-based clustering approach where clusters persist over time, despite continuous changes in cluster memberships. This corresponds closely to the “assembly line” dynamic clustering scenario described in [2].

3 Methods

3.1 Problem Definition

We represent a dynamic feature-based dataset as a set of l bipartite graphs $\{G_1, \dots, G_l\}$. Each *step graph* G_t consists of two sets of nodes, representing the

n_t data objects, and m_t features present in the data at time t . Edges exist only between nodes of different types, corresponding to non-zero feature values. We can conveniently represent each step graph using a feature-object matrix \mathbf{A}_t of size $m_t \times n_t$.

In the offline formulation of the dynamic co-clustering problem, the overall goal is to identify a set of *dynamic clusters* of objects and features, which appear in the data across one or more time steps. We refer to *step clusters* that are identified on individual step graphs, which represent specific observations of dynamic clusters at a given point in time. The formulation therefore has two key requirements: a suitable clustering algorithm to cluster individual time step graphs (ideally in a way that incorporates historic information), and an approach to track these clusters across time steps. While our primary focus here is on the former aspect, in Section 3.3 we also briefly discuss the latter aspect.

3.2 Dynamic Spectral Co-clustering

We now introduce a dynamic co-clustering algorithm that considers both historic information from the previous time step, and the internal quality of the clustering in the current time step. The algorithm consists of three phases: bipartite spectral embedding, cluster initialisation, and a cluster assignment phase.

Spectral embedding. Following normalised cut optimisation via spectral co-clustering described in [1], for a given time step feature-object matrix \mathbf{A}_t , we construct the degree-normalised matrix $\hat{\mathbf{A}}_t = \mathbf{D}_1^{-\frac{1}{2}} \mathbf{A}_t \mathbf{D}_2^{-\frac{1}{2}}$, where \mathbf{D}_1 and \mathbf{D}_2 are diagonal column and row degree matrices respectively. We then apply SVD to $\hat{\mathbf{A}}_t$, computing the leading left and right singular vectors corresponding to the largest singular values. Following the choice made by many authors in the spectral clustering literature, we use k_t dimensions corresponding to the expected number of clusters. Although the issue of selecting the number of clusters is not discussed in this paper, one potential approach is to choose k_t based on the eigengap method [9]. The truncated SVD yields matrices \mathbf{U}_{k_t} and \mathbf{V}_{k_t} . A unified embedding of size $(m_t + n_t) \times k_t$ is constructed by normalising and stacking the truncated factors as follows:

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{D}_1^{-1/2} \mathbf{U}_{k_t} \\ \mathbf{D}_2^{-1/2} \mathbf{V}_{k_t} \end{bmatrix} \quad (1)$$

Prior to clustering, the rows of \mathbf{Z}_t are subsequently re-normalised to have unit length, as proposed for spectral partitioning in [9]. This process provides us with a k_t -dimensional embedding of all nodes of both types in G_t .

Cluster initialisation. At time $t = 1$, we have no historic information. Therefore to seed the clustering process, we use a variant of orthogonal initialisation as proposed by Ng *et al.* [9] for spectral graph partitioning. This operates using a “farthest-first” strategy as follows. The first cluster centroid is chosen to be the

mean vector of the rows in \mathbf{Z}_t . We then repeatedly select the next centroid to be the row in \mathbf{Z}_t that is closest to being 90° from those that have been previously selected. This process continues until k_t centroids have been chosen.

For each time step $t > 1$, we initialise using clusters from the previous time step. A simple approach is to map the clusters generated on the embedding for time $t - 1$ to \mathbf{Z}_t . However, as noted previously, not all features and objects will persist between time steps. To produce an initial clustering at time t , we identify the intersection of the sets of nodes present in the graphs G_{t-1} and G_t . The clusters containing these are mapped to the embedding \mathbf{Z}_t , and we compute the resulting centroids. If less than k_t centroids are produced, the remaining centroids are chosen from the rows of \mathbf{Z}_t using orthogonal selection as above. We can then predict memberships for each unassigned row z_i of \mathbf{Z}_t , using a simple nearest centroid classifier to maximise the similarity:

$$\max_{C \in \mathcal{C}_t} z_i^\top \mu_c \quad (2)$$

where μ_c is the centroid of cluster C_c . This classification procedure yields a predicted clustering for rows in \mathbf{Z}_t (*i.e.* a co-clustering of all objects and features present at time t), which we denote \mathcal{P}_t .

Cluster assignment. To recover a clustering from \mathbf{Z}_t , we apply a constrained version of k -means clustering to the rows of the embedding, which takes into account both the internal quality of the current partition, and agreement with the predicted partition \mathcal{P}_t . We distinguish the latter from the membership preservation objective described in [6] – here we use predicted memberships for missing objects and features missing from the previous step.

As a measure of current cluster quality, we use vector-centroid similarities as in Eqn. 2. Historical quality is calculated based on the quantity $\text{pred}(\mathcal{P}_t, \mathcal{C}_t)$, which denotes the degree to which the predicted cluster assignments in \mathcal{P}_t agree with those in the current clustering \mathcal{C}_t . To quantify this agreement, we use a variant of pairwise *prediction strength* [10]:

$$\text{pred}(\mathcal{P}_t, \mathcal{C}_t) = \sum_{C \in \mathcal{C}_t} \frac{1}{|C|(|C| - 1)} \sum_{(z_i, z_j) \in C} co(z_i, z_j) \quad (3)$$

where $co(z_i, z_j) = 1$ if both rows were predicted to be coassigned in \mathcal{P}_t , or $co(z_i, z_j) = 0$ otherwise.

To combine both sources of information, the clustering objective then becomes a weighted combination of two objectives:

$$J(\mathcal{C}_t) = (1 - \alpha) \cdot \left(\sum_{c=1}^k \sum_{z_i \in C_c} z_i^\top \mu_c \right) + \alpha \cdot (\text{pred}(\mathcal{P}_t, \mathcal{C}_t)) \quad (4)$$

This type of aggregation approach has been widely used for combining sources of information, such as in dynamic clustering [3] and semi-supervised learning [11].

-
1. Build spectral embedding
 - Construct the normalised feature-object matrix $\hat{\mathbf{A}}_t = \mathbf{D}_1^{-\frac{1}{2}} \mathbf{A} \mathbf{D}_2^{-\frac{1}{2}}$.
 - Compute \mathbf{Z}_t from the truncated SVD of $\hat{\mathbf{A}}_t$ according to Eqn. 1.
 - Normalise the rows of \mathbf{Z}_t to unit length.
 2. Initialisation and prediction
 - If $t = 1$, apply orthogonal initialisation to select a set of k_t representative centroids from the representations of the objects in the embedded space.
 - For $t > 1$, recompute the k_{t-1} centroids based on last clustering but including only the embedding of the relevant set of objects/features in the current space.
 - If not all rows of the embedding have been assigned, apply nearest centroid classification to compute the predicted clustering \mathcal{P}_t .
 3. Compute clustering
 - Apply constrained k -means to rows in \mathbf{Z}_t , initialised by centroids from \mathcal{P}_t .
-

Fig. 2. Dynamic spectral co-clustering process, as applied for each time step t .

The parameter $\alpha \in [0, 1]$ controls the balance between the influence of historical information and the information present in the current spectral embedding. A higher value of α allows information from the previous time step to have a greater influence, yielding a smoother transition between clusterings at successive time steps. Naturally at time $t = 1$, the right-hand term in Eqn. 4 will be zero.

Eqn. 4 can be viewed as the standard spherical k -means objective [12], augmented by a constraint reward term. We can find a local solution for this problem by using an approach analogous to the semi-supervised PCKMeans algorithm proposed by Basu *et al.* [11] for clustering with pairwise constraints. Specifically, we apply an iterative k -means-like assignment process, re-assigning each row vector z_i from \mathbf{Z}_t to maximise:

$$\max_{C \in \mathcal{C}_t} (1 - \alpha) \cdot z_i^\top \mu_c + \alpha \cdot \text{pred}(z_i, C) \quad (5)$$

where the quantity $\text{pred}(z_i, C)$ represents the degree to which the predicted assignment for the row z_i in P_t agrees with the assignment of z_i to cluster C . This is given by the proportion of rows in C that were co-assigned with z_i in P_t :

$$\text{pred}(z_i, C) = \frac{1}{|C|(|C| - 1)} \sum_{(z_i, z_j) \in C} \text{co}(z_i, z_j) \quad (6)$$

Once the algorithm has converged to a local solution, \mathcal{C}_t provides us with a k -way partitioning of all nodes in the graph G_t (*i.e.* features and objects). An overview of the complete co-clustering process is shown in Figure 2.

3.3 Tracking Clusters Over Time

In the previous section we proposed an approach for co-clustering individual time step graphs. The second aspect of the offline approach to dynamic clustering

involves identifying *dynamic clusters* composed from clusters associated across time steps. We suggest that previous frameworks for tracking evolving dynamic communities [2, 13] can be readily adapted to the dynamic bipartite case. In brief, we construct a set of dynamic cluster timelines, each consisting of a set of clusters identified at different time steps and ordered by time. At each step in the dynamic co-clustering process, we match the predicted clusters (corresponding to clusters from the previous time step) with the actual output of the co-clustering process outlined in Figure 2. Matches are made based on the step cluster memberships for subsets of objects and/or features persisting between pairs of consecutive steps. This matching process will result in a set of dynamic clusters persisting across multiple steps.

4 Evaluation

4.1 Benchmark Evaluation

To evaluate the performance of the algorithm proposed in Section 3.2, we required an annotated dataset with temporal information. For this purpose we consider the bipartite document clustering problem, and use a subset of the widely-used Reuters RCV1 corpus [14]. The *RCV1-5topic* dataset¹ consists of 10,116 news articles covering a seven month period. Each article is annotated with a single ground truth topical label: health, religion, science, sport, weather. These topics are present across the entire time period of the corpus. We considered a number of different time step durations to split the seven month period – one month, a fortnight, and one week – yielding 7, 14, and 28 step graphs respectively. Naturally for this type of data, a subset of features (terms) will persist across time, while objects (documents) appear in only one time step.

Our evaluations focused on the performance of the dynamic spectral co-clustering algorithm on each time step graph in the *RCV1-5topic* dataset, using a range of values $\alpha \in [0.1, 0.5]$ for the balance parameter. As a baseline competitor, we used multi-partition spectral co-clustering as proposed by Dhillon [1]. To provide a fair comparison, we use orthogonal initialisation for both algorithms, and set the number of clusters k_t at time t to the number of ground truth topics.

Temporal smoothness. One of the primary motivations for dynamic co-clustering is to increase smoothness in the transitions between time step clusterings. To quantify the degree to which the proposed algorithm can enforce temporal smoothness, we measure the agreement between successive clusterings in terms of their *normalised mutual information* (NMI) [15]. Note that NMI values were calculated only over the terms common to each pair of consecutive time steps – documents are not considered as they do not persist.

Figure 3 shows a comparison of agreement values for the three different time window sizes. Dynamic co-clustering leads to a higher level of agreement than

¹ Datasets for this paper are available at <http://mlg.ucd.ie/datasets/dynak.html>

standard spectral co-clustering for all three time window sizes. The effect becomes significantly more pronounced as α increases. This is to be expected, as increasing the parameter leads to a higher weighting for the historic information in Eqn. 4. For $\alpha \geq 0.5$, the resulting co-clusterings are often almost identical to the predicted co-clustering \mathcal{P}_t , with the constrained k -means process converging to a solution after 2-5 iterations.

Clustering accuracy. To quantify algorithm accuracy, we calculated the NMI between clusterings and the relevant annotated document label information for

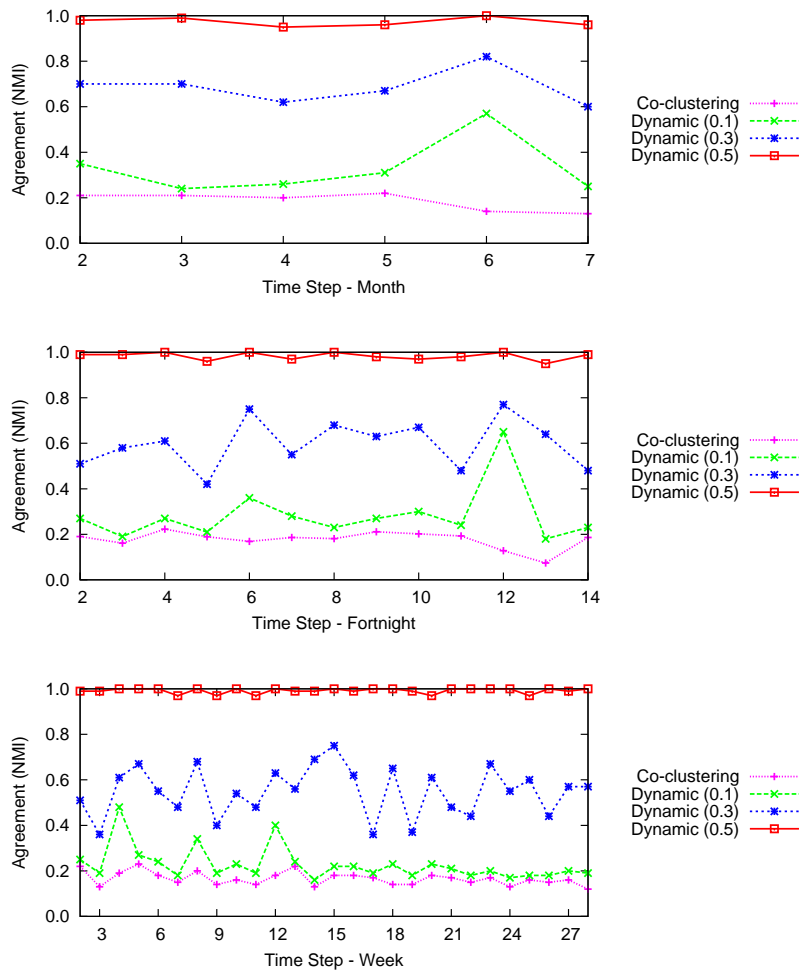


Fig. 3. Comparison of agreement (in terms of NMI) between successive feature clusterings, generated by spectral co-clustering and dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *RCV1-5topic* dataset for monthly, fortnightly, and weekly time steps.

each time step. Figure 4 illustrates a comparison of the accuracy achieved by traditional spectral co-clustering and dynamic co-clustering on the *RCV1-5topic* dataset for the three different time step sizes. We observed that, for monthly and fortnightly time steps, the accuracy achieved by dynamic co-clustering was not significantly higher. However, for the weekly case, there was a noticeable increase in accuracy. In the case of $\alpha = 0.5$, dynamic co-clustering lead to higher accuracy on 21 out of 28 of the weekly graphs.

These results could appear surprising given the increases in temporal smoothness demonstrated Figure 3. However, on closer inspection, it is apparent that there is a strong *concept drift* effect in the data, as the composition of topics

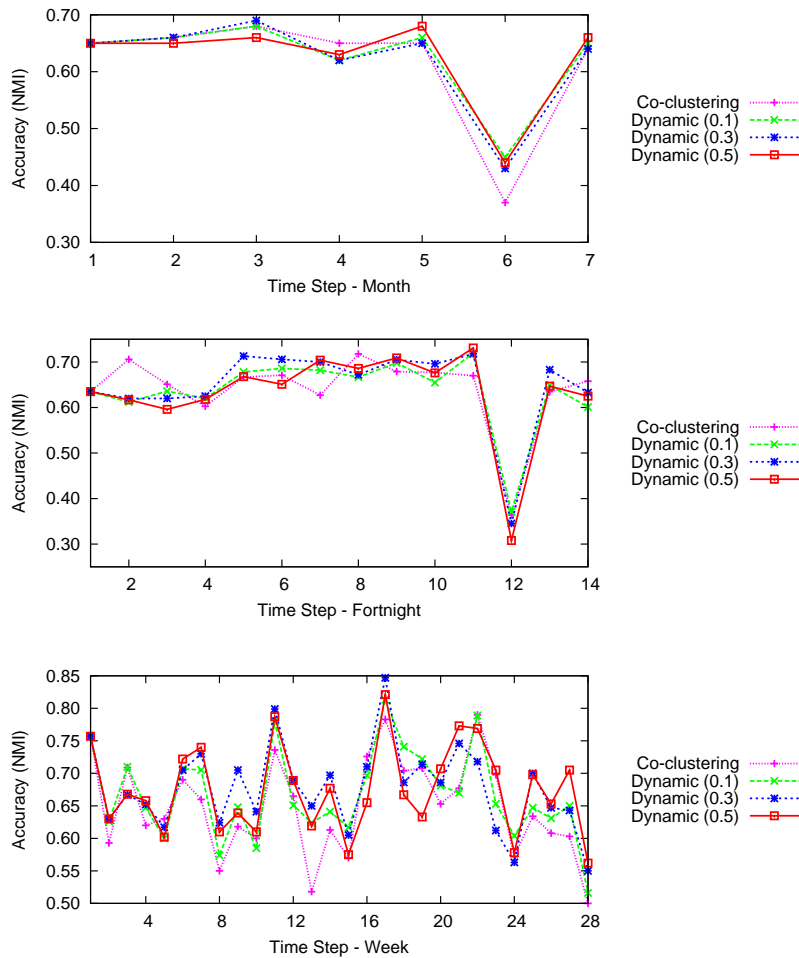


Fig. 4. Comparison of accuracy (in terms of NMI) for document clusterings generated by spectral co-clustering and dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *RCV1-5topic* dataset for monthly, fortnightly, and weekly time steps.

changes over seven months. Therefore, for longer time periods, there is a greater change in the clusters identified in successive time periods. In such cases we expect historic information to be less useful. For the shorter weekly time windows, where there is less scope for drift between steps, we expect the use of historic information to improve accuracy. These results highlight the importance of selecting an appropriate time step size for offline dynamic clustering.

4.2 Evaluation on Web 2.0 Data

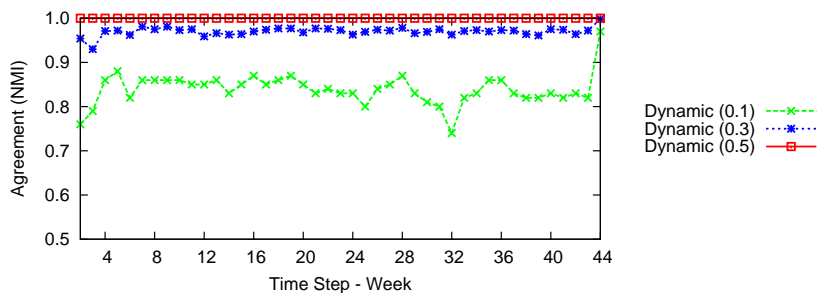
For the second phase of our evaluation, we applied the proposed co-clustering algorithm to a Web 2.0 data exploration problem. Unlike the RCV1 data, subsets of **both** objects (bookmarks) and features (tags) persist over time. We use a subset of the most recent data from a collection harvested by Görlitz *et al.* [16] from the *Del.icio.us* web bookmarking portal. The subset covers the 2000 top tags and 5000 top bookmarks across an eleven month period from January-November 2006. We divided this period into 44 weekly time steps, and for each time step we constructed a bipartite graph – the nodes represent tags and bookmarks, and the edges between them denote the number of times each bookmark was assigned a given tag during the time step. On average, each graph contained approximately 3750 bookmarks and 1760 tags. For each time step, we applied dynamic co-clustering for $k_t = 20$ to identify high-level topical clusters.

Figure 5 illustrates the agreement between both tag and bookmark clusterings identified by dynamic co-clustering for a balance parameter range $\alpha \in [0.1, 0.5]$. As with the *RCV1-topic* data, an increase in the value of α leads to clusters that are considerably more similar to those produced in the previous step, yielding smoother transitions between both feature and object clusters across time. In the extreme case of $\alpha = 0.5$, there is effectively no change between the predicted memberships and the final output of the co-clustering algorithm.

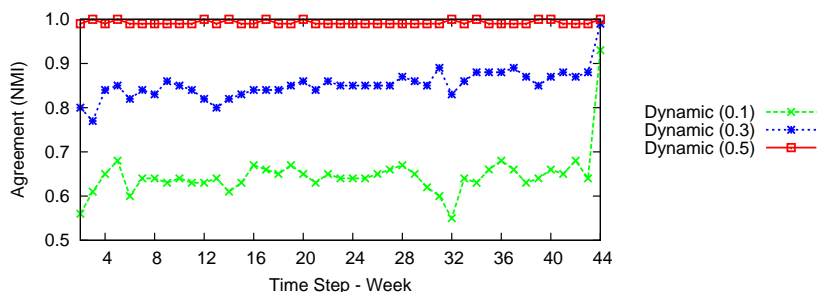
A number of authors (*e.g.* [17]) have suggested analysing the stability or “loyalty” of object member-cluster memberships across time. In the bipartite case, we can quantify this for both objects and features – we suggest the latter can be used to generate meaningful labels for dynamic clusters. For the tagged data, we score the fraction of time steps at which a tag is assigned to a given dynamic cluster. Over a sufficiently large number of steps, for each dynamic cluster we can produce a robust ranking of tags based on their respective membership stability scores. Examining the range of α parameters, we found the trade-off afforded by $\alpha = 0.1$ lead to the most interpretable label sets. In Table 1 we show the resulting descriptive labels selected for the dynamic clusters that exhibited the highest average tag membership stability, together with a suggested topic based on the tags. These descriptions highlight a range of general areas of interest covering sites frequently bookmarked by users during 2006.

5 Conclusion

In this work, we have described a spectral co-clustering algorithm for simultaneously clustering both objects and features in dynamic feature-based data,



(a) Object (Bookmark) Clustering Agreement



(b) Feature (Tag) Clustering Agreement

Fig. 5. Agreement between successive object and feature clusterings, identified by dynamic co-clustering ($\alpha \in [0.1, 0.5]$), on the *Del.icio.us* dataset across 44 weeks.

Topic	Top 10 Tags
IT	shortcuts, tweaks, opensource, security, troubleshooting, system, livecd, keyboard, sysadmin, ssh
Education	academic, school, mathematics, education, spanish, grammar, elearning, learning, math, slang
Music & Video	podcasts, mp3blog, youtube, television, movie, bittorrent, divx, torrent, p2p, npr
News & Media	newspapers, culture, opinion, society, iraq, news, journalism, environment, activism, political
Web Browsing	explorer, thunderbird, browser, opera, firefox, extensions, mozilla, greasemonkey, computing, plugin

Table 1. Top 10 tags for 5 most stable clusters (in terms of tag memberships over time) identified on the *Del.icio.us* dataset by dynamic co-clustering ($\alpha = 0.1$).

represented as a sequence of bipartite graphs. The co-clustering algorithm incorporates both current and historic information into the clustering process. A key aspect of the approach is that it is applicable in domains where objects or

features alone persist across time steps. In applications on both dynamic text and bookmark tagging data, the proposed approach was successful in identifying coherent clusters, while also ensuring a consistent transition between clusterings in successive time steps.

Acknowledgments. This work is supported by Science Foundation Ireland Grant No. 08/SRC/I140 (Cliques: Graph & Network Analysis Cluster)

References

1. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. (2001) 269–274
2. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proc. 13th International conference on Knowledge Discovery and Data mining (KDD '07). (2007) 717–726
3. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proc. 12th Int. Conf. on Knowledge Discovery and Data Mining. (2006) 554–560
4. Mirzal, A., Furukawa, M.: Eigenvectors for clustering: Unipartite, bipartite, and directed graph cases. arXiv (2010)
5. Yang, Y., Pierce, T., Carbonell, J.: A study of retrospective and on-line event detection. In: Proc. 21st International ACM SIGIR Conference on Research and development in information retrieval. (1998) 28–36
6. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.: Evolutionary spectral clustering by incorporating temporal smoothness. In: Proc. 13th SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. (2007) 153–162
7. Palla, G., Barabási, A., Vicsek, T.: Quantifying social group evolution. *Nature* **446**(7136) (2007) 664
8. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. Proc. SSTD 2005 (2005) 364–381
9. Ng, A., Jordan, M., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. *Advances in Neural Information Processing* **14**(2) (2001) 849–856
10. Tibshirani, R., Walther, G., Botstein, D., Brown, P.: Cluster validation by prediction strength. Technical report, Dept. Statistics, Stanford University (2001)
11. Basu, S., Banerjee, A., Mooney, R.: Active semi-supervision for pairwise constrained clustering. In: Proc. SIAM Int. Conf. on Data Mining. (2004) 333–344
12. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. *Machine Learning* **42**(1-2) (January 2001) 143–175
13. Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: Proc. International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10). (2010)
14. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. *JMLR* **5** (2004) 361–397
15. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *JMLR* **3** (December 2002) 583–617
16. Görlitz, O., Sizov, S., Staab, S.: Pints: Peer-to-peer infrastructure for tagging systems. In: Proc. 7th International Workshop on Peer-to-Peer Systems. (2008)
17. Berger-Wolf, T., Saia, J.: A framework for analysis of dynamic social networks. In: Proc. 12th Int. Conf. on Knowledge Discovery and Data Mining. (2006) 523–528