HIERARCHICAL CO-CLUSTERING OF MUSIC ARTISTS AND TAGS

Jingxuan Li
School of Computer Science
Florida International University
Miami, FL USA
jli003@cs.fiu.edu

Tao Li
School of Computer Science
Florida International University
Miami, FL USA
taoli@cs.fiu.edu

Mitsunori Ogihara
Department of Computer Science
University of Miami
Coral Gables, FL USA
ogihara@cs.miami.edu

ABSTRACT

The user-assigned tag is a growingly important research topic in MIR. Noticing that some tags are more specific versions of others, this paper studies the problem of organizing tags into a hierarchical structure by taking into account the fact that the corresponding artists are organized into a hierarchy based on genre and style. A novel clustering algorithm, *Hierarchical Co-clustering Algorithm (HCC)*, is proposed as a solution. Unlike traditional hierarchical clustering algorithms that deal with homogeneous data only, the proposed algorithm simultaneously organizes two distinct data types into hierarchies. HCC is additionally able to receive constraints that state certain objects "must-be-together" or "should-be-together" and build clusters so as to satisfying the constraints.

HCC may lead to better and deeper understandings of relationship between artists and tags assigned to them. An experiment finds that by trying to hierarchically cluster the two types of data better clusters are obtained for both. It is also shown that HCC is able to incorporate instance-level constraints on artists and/or tags to improve the clustering process.

1. INTRODUCTION

The user-defined tags are becoming an essential component in web databases and social network services. The tags assigned to events and data objects as a whole represent how they are received by the community and provide keys to other users accessing them. In music information retrieval some recent papers study how to incorporate tags effectively for fundamental data retrieval tasks such as clustering, recommendation, and classification (see, e.g., [4, 19, 21, 22]).

An important characteristic of the tags is that sometimes tags are extensions of others and thus more specific than those they extend, e.g., "Soft Metal" extending "Metal", "Dance Pop" extending "Pop", and "Extremely Provocative" extending "Provocative". Since there is no limit in the length of tags, a tag can be an extension of another one, which is an extension of yet another one. This suggests that the music tags can be not only clustered as has been done before but *hierarchically* clustered.

Many approaches have been developed to produce hierarchical organizations of words and of documents, and so organizing tags into a hierarchy is a problem that is already-solved. However, we observe that the artists, to which tags are assigned, too can be organized into a hierarchical structure based on their prominent genres and styles. In fact, these hierarchies are much related to each other, since style labels often appear as tags. This leads to the questions of whether an attempt to build simultaneously hierarchical organizations of tags and of artists will lead to better organizations of both and of whether such organizations can be effectively and efficiently built.

In data mining the problem of developing hierarchical organization of data is referred to as *hierarchical clustering* and the problem of clustering two data types is referred to as *co-clustering*. While co-clustering essentially aims at simultaneously clustering rows and columns of a matrix, where the rows and the columns correspond to separate data types (e.g., terms and documents), hierarchical clustering aims at building a tree-like structure of the rows based on the columns a tree-like structure of the columns based on the rows. While both organizations have their own advantages, such as natural facilitation of data navigation and browsing in hierarchical clustering [6], few algorithms simultaneously build both [2].

In this paper, we develop a novel method, called HCC, for simultaneously clustering two data types, and we use HCC for building hierarchical co-clusters of tags and styles. HCC is designed based on the approaches in [10, 14]. HCC is essentially agglomerative hierarchical clustering: it starts with singleton clusters and then repeatedly merging two nearest clusters into one until there remains only one cluster. However, it may *merge groups from different data types at any point*. In our case, this means that at each step of the merging process, HCC can merge a subset of the artists with a subset of the tags based on their internal heterogeneity. In practice, one sometimes observes that a group of artists and a group of tags are exclusively correlated with each other (i.e., not correlated with any other artists or tags). HCC aims at, in such a situation, merging them into a single group at the earliest possible stage.

Our hope is that such artist-tag mixed clusters will be used for better retrieval when both artists and tags are specified in a query.

Figure 1 shows a sample output dendrogram of HCC while Figure 2 shows a sample output dendrogram of a traditional hierarchical clustering method. We show that such mixed-data-type hierarchical clusters can be generated by HCC and empirically better clusters are generated by concurrent use of two data types. Furthermore, we show that HCC can be extended to incorporate instance-level constraints that specify certain tags must be or must not be together or certain artists must be or must not be together for better organization.

The rest of the paper is organized as follows: Section 2 discusses the related work; Section 3 describes the details of HCC and the techniques for incorporating instance-level constraints; Section 4 presents experimental results; and finally Section 5 provides our conclusions.

2. RELATED WORK

Hierarchical Clustering is generation of tree-like cluster structures without user supervision. Hierarchical clustering algorithms organize input data either bottom-up (agglomerative) or top-down (divisive) [20]. **Co-clustering** refers to clustering of more than one data type. Dhillon [7] proposes bipartite spectral graph partitioning approaches to co-cluster words and documents. Long et al. [15] proposed a general principled model, called Relation Summary Network, for co-cluster heterogeneous data presented as a k-partite graph. While hierarchical clustering deals with only one type of data and the organization that co-clustering produces consists of just one level, **Hierarchical Co-clustering** aims at simultaneously construction of two or more hierarchies [12, 13].

Recently much work has been done on the use of background information in the form of instance level must-link and cannot-link constraints. This topic is referred to as **Constrained Clustering**. Here a must-link constraint enforces that two instances must be placed in the same cluster and a cannot-link constraint enforces that two instances must not be placed in the same cluster. Most of these constraint-based algorithms are developed for partitional clustering (e.g, K-means clustering, spectral clustering, and non-negative matrix factorizations) [1], and little has been done on utilizing constraints for hierarchical clustering.

3. HIERARCHICAL CO-CLUSTERING (HCC)

3.1 Problem Formulation

Suppose we are given a set of m artists $\boldsymbol{A}$=$\{a_1, a_2, \ldots, a_m\}$, and a set of n unique tags that are assigned to the music of these artists $\boldsymbol{T}$=$\{t_1, t_2, \ldots, t_n\}$. Suppose we are also given an $m \times n$ artist-tag relationship matrix $X = (x_{ij}) \in \mathbb{R}^{m \times n}$, such that x_{ij} represents the relationship between the i-th artist in $\boldsymbol{A}$ and the j-th tag in $\boldsymbol{T}$. Our goal is to simultaneously generate a hierarchical clustering of $\boldsymbol{A}$ and of $\boldsymbol{T}$ based on matrix X.

3.2 HCC

Like agglomerative hierarchical clustering algorithms, HCC starts with singleton clusters and then successively merges the two nearest clusters until only one cluster is left. However, unlike traditional algorithms, it may unify classes from two different data types. This means that the cluster left at the end consists of all the rows and columns and so if there are m rows and n columns exist, HCC executes $m+n-1$ rounds. The output of HCC is thus a single tree where the leaves are the rows and the columns of the input matrix, where nodes having both rows and columns as descendants may appear at any non-leaf level. Note that, in Figure 1, at the third layer the artist A3 - Led Zeppelin is joined with the tag B2 - Classic rock.

The algorithm of HCC is presented in Algorithm 1. The

Algorithm 1 HCC Algorithm Description

```
Create an empty hierarchy H
List ← Objects in A + Objects in B
N ← size[A] + size[B]
Add List to H as the bottom layer

for i = 0 to N − 1 do
  p, q = PickUpTwoNodes(List)
  o = Merge(p, q)
  Remove p, q from List and add o to List
  Add List to H as the next layer
end for
```

central part in the design of Algorithm 1 is the method PickUpTwoNodes, which is for selecting two nodes (corresponding to two clusters) to merge. For the purpose of creating groups consisting of two different data types, we use cluster heterogeneity measurement, denoted by CH. Given a group C consisting of r rows, P, and s columns, Q, we define $CH(C)$ as

$$CH(C) = \frac{1}{rs} \sum_{i \in P, j \in Q} (x_{ij} - \mu)^2, \qquad (1)$$

where μ is the average of entries over rows P and columns Q; i.e., $\mu = \frac{1}{rs} \sum_{i \in P, j \in Q} x_{ij}$. For a merger, we choose the two nodes whose merging would result in the least increase in the total cluster heterogeneity [10].

3.3 Incorporating Instance-level Constraints

In practice, one may observe pairs of artists that should be clustered into the same cluster. Similarly, one may observe pairs of tags that must be always in the same tag cluster. These observations are represented as the aforementioned "must-link" and "cannot-link" constraints. We design HCC so as to incorporate such constraints.

There are two issues in incorporating these constraints. One is how to use them for grouping data points of the same type; i.e., how to use artist constraints for grouping artists and tag constraints for grouping tags. The other is how to transfer constraints on one data type to the other data type. To address the first issue, we use Dunn's Index

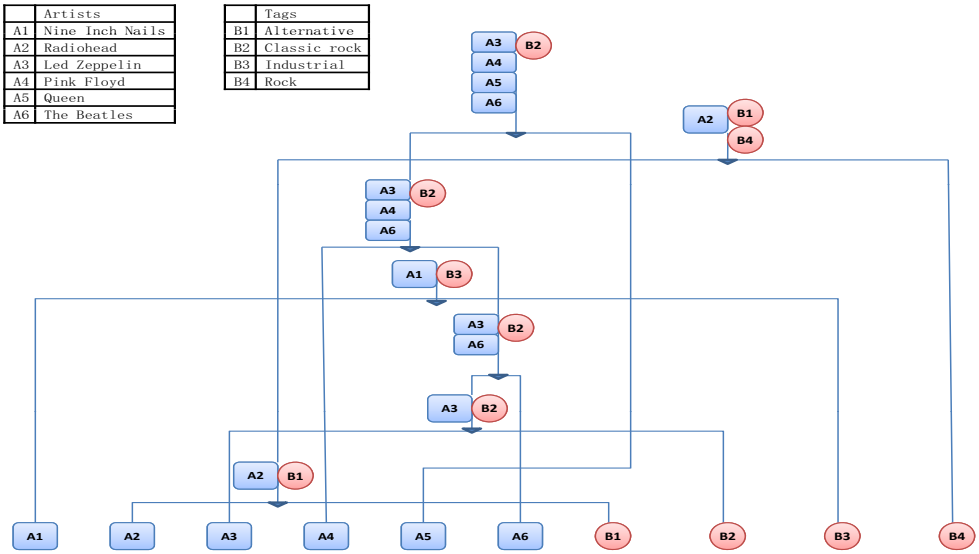

Figure 1. Part of HCC dendrogram. Rectangles represent artists and ellipses represent tags assigned to these artists. The nodes containing both rectangles and ellipses are clusters containing both an artist and a tag.

to determine the best layer for cutting the HCC-generated dendrogram and then apply the constrained K-Means to incorporate the constraints of the same data type. To address the second issue, we use an alternating exchange algorithm.

3.3.1 Best Layer

Since HCC produces a list of clustering results and each clustering corresponds to one layer of the dendrogram, we use Dunn's Validity Index [9] to measure and compare these clusterings. This validity measure is based on the idea that good clustering produces well-separated compact clusters. Given a clustering layer consisting of r clusters $c_1, \ldots, c_r$, Dunn's Index is given by:

$$D = \frac{\min_{1 \le i < j \le r} d(c_i, c_j)}{\max_{1 \le k \le r} d'_k}, \tag{2}$$

where $d(c_i, c_j))$ is the inter-cluster distance between the i-th and the j-th clusters and d'_k is the intra-cluster distance of the k-th cluster. Generally, the larger Dunn's Index, the better the clustering.

After determining the best layer to cut the dendrogram, we can easily make use of the constraints of the same data type. In particular, we perform constrained K-Means on the best layer with the parameter K set to the number of clusters in that layer. For this purpose, we use the MPCK-Means algorithm in [3].

3.3.2 Alternating Exchange

Here we show how to transfer the constraints between different data types. Specifically, at the best layer of the dendrogram generated by HCC, if some artist (or tag) data points of certain node are being re-assigned to another node at the same layer after using the instance-level constraints, we can use an alternating exchange algorithm [11] to improve tag (or artist) clustering. The objective function of clustering can be written as [11]:

$$Z = \sum_{k=1}^{r} \sum_{l=1}^{m} \sum_{i \in A_k} \sum_{j \in T_l} (x_{ij} - w_{kl})^2, \tag{3}$$

with

$$w_{kl} = \frac{1}{a_k t_l} \sum_{i \in A_k} \sum_{j \in T_l} x_{ij}. \tag{4}$$

Here r is the number of type A clusters, m is the number of type T clusters, A_k is the k-th cluster contains data points of type A, T_l is the l-th cluster contains data points of type T, a_k and t_l respectively denote data points of type A and T. As before, x_{ij} is the value representing the relationship between the i-th type-A data point and the j-th type-T data point.

To transfer constraints from tags to artists, we do the following: Suppose we have just obtained a clustering of artists, C_A, and a clustering of tags, C_T, by cutting the HCC dendrogram using Dunn's index, as described before. We first incorporate into these clusterings the tag constraints using the techniques described in Section 3.3.1 thereby obtain an improved tag clustering, C'_T. Then we execute the greedy algorithm shown in Algorithm 2 to make changes on artist class assignments. The greedy algorithm is aimed at minimizing the quantity Z in (3) and in each round one artist is moved from the current cluster to another if that move decreases the value of Z. Transferring constraints backward (i.e., from artists to tags) could be done by simplying switch the role of tags and artists. In our implemenation, we transfer only from tags to artists.

Algorithm 2 Alternating Exchange Algorithm

Input: clusterings C_A and C'_T, and normalized A-T matrix X, where C'_T is obtained by using the MPCK-Means on the output of HCC with respect to tag constraints.

while There is an artist whose relocation from the current cluster to another decreases the value of Z **do**
  pick an artist-destination pair that maximizes the decrease and relocate the artist to the destination
end while
Output the resulting artist clustering C'_A

4. EXPERIMENT

4.1 Data Set

We use the data set in [22] consisting of 403 artists. For each artist, tags and styles are collected from Last.fm (http://www.last.fm). There are 8,529 unique tags and 358 unique style labels. Note that an artist may receive the same tag more than once. By counting the number of assignments by the same tag, each artist is represented by a 8,529-dimensional integer vector. We scale these tag vectors so that the total of the 8,529 entries is equal to a fixed constant. We will use X to denote the artist-tag frequency matrix thus generated.

As to the style labels, each artist belongs to at least one style and each style contains at least one artist. We generate an artist-style incident matrix from the data, so that the entry at coordinate (i, j) is 1 if the i-th artist has the j-th style label and 0 otherwise.

4.2 Hierarchies Generated from HCC

We use HCC to generate a dendrogram of the artists and the tags. Figure 1 is part of the dendrogram generated by HCC in our experiment. In the dendrogram, each leaf represents one artist or one tag, each internal node contains subsets of artists and tags, and the top layer is the cluster contains all artists and tags. Because many people assign a tag "Industrial" to artist *Nine Inch Nails*, "Industrial" and *Nine Inch Nails* are clustered together. The novelty here is that artists and tags are jointly organized into a hierarchical structure. Once such a hierarchical organization has been generated, an artist can be described by the tags that appear in its cluster. The more representative are the tags for certain artists, the larger possibility for them to be clustered together.

We compare the HCC-generated dendrogram with one generated by single linkage hierarchical clustering (SLHC). This is the standard hierarchical clustering method and thus serves as our baseline. Since SLHC can cluster only one type of data, we provide SLHC with the normalized artist-tag matrix by viewing each row as the feature vector of the corresponding artist and produce hierarchical clustering of artists. The artist dendrogram generated by SLHC is shown in Figure 2. To evaluate and compare these two artist dendrograms, we utilize CoPhenetic

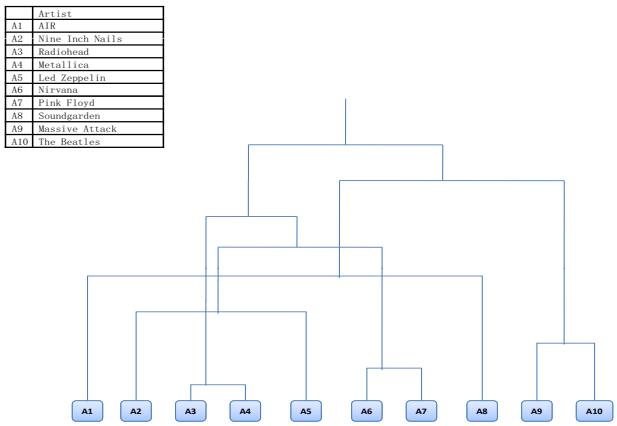

Figure 2. Part of the dendrogram generated by SLHC.

Correlation Coefficient (CPCC) [17] as evaluation measure. Intuitively CPCC measures how faithfully a dendrogram preserves the pairwise distances between the original data points. CoPhenetic Correlation Coefficient (CPCC) is given as:

$$\frac{\sum_{i<j}(d(i,j)-d)(t(i,j)-t)}{\sqrt{(\sum_{i<j}(d(i,j)-d)^2)(\sum_{i<j}(t(i,j)-t)^2)}} \quad (5)$$

Here $d(i, j)$ and $t(i, j)$ are respectively the ordinary Euclidean distance and the dendrogrammatic distance between the i-th and the j-th data points, and d and t are their respective averages. The CPCC for HCC was 3.71 while that for SLHC was 3.69, and so we can say that our HCC method generates faithful dendrogram with reasonable clustering performance on artist-tag dataset. Through the coupled dendrogram, one can observe the relationship between artists and tags, also make use of the tags within the same cluster as some artists to explain why these artists are clustered together.

4.3 Clustering Performance Comparisons

We also evaluate the artist clustering performance of HCC, by comparing it with three co-clustering algorithms including Information-Theoretic Co-clustering (ITCC) [8], Euclidean Co-clustering (ECC), and Minimum Residue Co-clustering (MRC) [5] on the artist-tag dataset.

Using style labels we obtain artist clusters and cluster labels. We first cluster the styles using KMeans clustering based on the artist-style matrix (that is, clustering of the columns, where each column is the 403-dimensional $0/1$ vector that shows assignments of the style corresponding to the column to the 403 artists). We then treat each cluster as a label and assign to each artist one label in the following manner:

- If all the styles assigned to an artist a belongs to a single cluster, we use that cluster as the label of a. Otherwise, choose the cluster with the largest number of styles assigned to a. If there is a tie, choose the one with the larger total number of styles, and if that doesn't break the tie, break it arbitrarily.

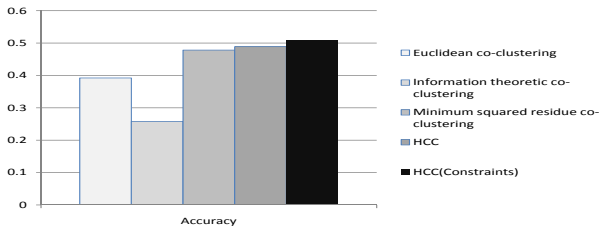

Figure 3. Accuracy of various clustering methods. HCC(constraints) represents HCC with 10 artist constraints.

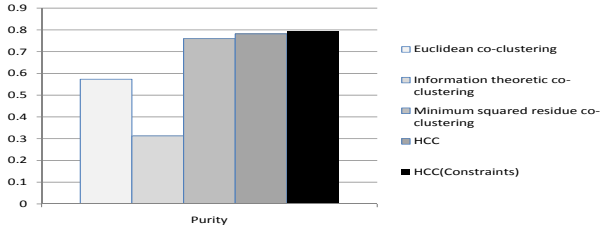

Figure 4. Purity of various different clustering methods. HCC(constraints) represents HCC with 10 artist constraints.

We use these labels as our ground truth class labels in the clustering performance measurements presented below.

4.3.1 Evaluation Measures

We use Accuracy, Normalized Mutual information (NMI), Purity, and Adjusted Rand Index (ARI) as performance measures. These measures have been widely used in clustering evaluation and we hope they would provide insights on the performance of our HCC method. For all these measures, the higher the value, the better the clustering.

Suppose we are given clusters $C_1, \ldots, C_k$ of size $c_1, \ldots, c_k$, respectively and we are comparing this clustering against the ground-truth clustering $E_1, \ldots, E_k$ of size $e_1, \ldots, e_k$. Let n be the total number of data points and for all i and j, let μ_{ij} denote the number of data points in both C_i and E_j.

Accuracy measures the extent to which each cluster contains the entities from corresponding class and is given by:

$$Accuracy = \max_{\pi} \frac{\sum_{i,\pi(i)} \mu_{i\pi(i)}}{n}, \tag{6}$$

where π ranges all permutations of $1, \ldots, k$. **Purity** measures the extent to which a cluster contains entities of a single class and is given by:

$$Purity = \frac{1}{n} \sum_{i=1}^{k} \mu_{i\rho(i)}, \tag{7}$$

where $\rho(i)$ is the j that maximizes μ_{ij}. **Adjusted Rand Index** is the corrected-for-chance version of Rand Index, and measures the similarity between two clusterings [16]. It is given by:

$$ARI = \frac{a - \frac{2bc}{n(n-1)}}{\frac{b+c}{2} - \frac{2bc}{n(n-1)}}. \tag{8}$$

Here $a = \sum_{i,j} \frac{\mu_{ij}(\mu_{ij}-1)}{2}$, $b = \sum_i \frac{c_i(c_i-1)}{2}$, and $c = \sum_j \frac{e_j(e_j-1)}{2}$. **NMI** is the normalized version of mutual information and measures how much information the two clusterings share [18] and is given by:

$$NMI = \frac{\sum_{i,j} \mu_{ij} \log(\frac{n\mu_{ij}}{c_i e_j})}{\sqrt{(\sum_i c_i \log \frac{c_i}{n})(\sum_j e_j \log \frac{e_j}{n})}}. \tag{9}$$

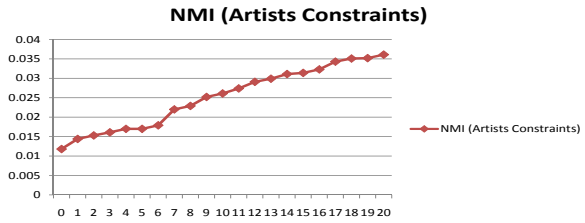

Figure 5. NMI on HCC with artists constraints range from 0 - 20.

4.3.2 Experimental Results

As we mentioned in Section 3.3.1, Dunn's Index can be used to find the best layer of the dendrogram generated by HCC. After computing Dunn's Index on the clustering of each layer, it is found that there are 11 clusters in the best layer. Since we have already obtained the best layer, the clustering of this layer is compared against Co-clustering algorithms. This clustering is based on artist data points, we applied co-Clustering algorithms for clustering artists.

Figures 3 and Figure 4 show the experiment results on the clustering methods using accuracy and purity as the performance measures, respectively. The results in both figures demonstrate that our HCC method outperforms the co-clustering methods. Similar behaviors can be observed when using ARI and NMI measures. Due to space limitation, we do not include the figures for ARI or NMI. Figures 3 and 4 also show that using the artist constraints improves the clustering performance.

We also evaluate NMI on HCC with increasing number of constraints. The result in Figure 5 shows that the artist clustering performance improves with the increasing number of artist constraints. In other words, the artist constraints improves the clustering performance of HCC. Figure 6 shows that artist clustering performance improves as the number of tag constraints increases.

5. CONCLUSION

In this paper, we propose a novel clustering method, HCC, to hierarchically cluster artists and tags simultaneously. With the dendrogram generated by HCC one can have a picture of all artists and tags, so as to find the relationship between the artists and tags within the same cluster. Furthermore, we perform experiments on artist-tag dataset, the results show that HCC outperforms its competitors, provid-

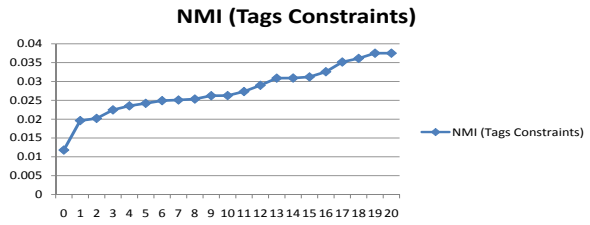

Figure 6. NMI on HCC with tags constraints range from 0 - 20.

ing reasonable dendrograms with clusterings in each layer. In the future, we would try out HCC on larger datasets to further confirm its ability in MIR area.

6. ACKNOWLEDGMENT

The work is partially supported by NSF grants IIS-0546280, CCF-0939179, and CCF-0958490 and an NIH Grant 1-RC2-HG005668-01.

7. REFERENCES

[1] S. Basu, I. Davdison, K. Wagstaff (eds): "Constrained Clustering: Advances in Algorithms, Theory, and Applications," Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2008.

[2] P. Berkhin: "A survey of clustering data mining techniques," *Grouping Multidimensional Data*, pp. 25–71, Springer, Heidelberg, 2006.

[3] M. Bilenko, S. Basu, and R. J. Mooney: "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. 21st International Conference on Machine Learning*, pp. 81–88, 2004.

[4] K. Bosteels, E. Pampalk, and E. E. Kerre: "Music retrieval based on social tags: A case study," in *Proc. 9th International Conference on Music Information Retrieval*, 2008.

[5] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra: "Minimum sum-squared residue co-clustering of gene expression data," in *Proc. 4th SIAM International Conference on Data Mining*, pp. 114–125, 2004.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey: "Scatter/gather: a cluster-based approach to browsing large document collections," in *Proc. 15th ACM SIGIR Conference*, pp. 318–329, 1992.

[7] I. S. Dhillon: "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 269–274, 2001.

[8] I. S. Dhillon, S. Mallela, and D. S. Modha: "Information-theoretic co-clustering," in *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 89–98, 2003.

[9] J. C. Dunn: "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybernetics*, 3(3):32–57, 1974.

[10] T. Eckes and P. Orlik: "An error variance approach to two-mode hierarchical clustering," *J. Classification*, 10(1):51–74, 1993.

[11] W. Gaul and M. Schader: "A new algorithm for two-mode clustering," in *Data analysis and information systems*, pp. 15–23, Springer, Heidelberg, 1996.

[12] M. Hosseini and H. Abolhassani: "Hierarchical co-clustering for web queries and selected urls,". in *Proc. Web Information Systems Engineering (WISE)*, pp. 653-662, 2007.

[13] D. Ienco, R.G. Pensa, and R. Meo: "Parameter-Free Hierarchical Co-clustering by n-Ary Splits,". in *Proc. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pp. 580–595, 2009.

[14] J. Li and T. Li: "HCC: A Hierarchical Co-Clustering Algorithm," in *Proc. 33rd ACM SIGIR Conference*, 2010.

[15] B. Long, X. Wu, Z. M. Zhang, and P. S. Yu: "Unsupervised learning on k-partite graphs," in *Proc. 12th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 317–326, 2006.

[16] G. W. Milligan and M. C. Cooper: "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behavioral Research*, 21(4):441–458, 1986.

[17] R. R. Sokal and F. J. Rohlf: "The comparison of dendrograms by objective methods," *Taxon*, 11:33–40, 1962.

[18] A. Strehl and J. Ghosh: "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *J. Machine Learning Research*, 3:583–617, 2003.

[19] P. Symeonidis, M. M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos: "Ternary semantic analysis of social tags for personalized music recommendation," in *Proc. 9th International Conference on Music Information Retrieval*, pp. 219–224, 2008.

[20] P. Tan, M. Steinbach, and V. Kumar: "Introduction to Data Mining," Addison-Wesley, 2005.

[21] D. Turnbull, L. Barrington, and G. Lanckriet: "Five approaches to collecting tags for music," in *Proc. 9th International Conference on Music Information Retrieval*, pp. 225–230, 2008.

[22] F. Wang, X. Wang, B. Shao, T. Li, and M. Ogihara: "Tag integrated multi-label music style classification with hypergraphs," in *Proc. 10th International Society for Music Information Retrieval*, pp. 363–368, 2009.