Clustering Social Event Images using Kernel Canonical Correlation Analysis

Unaiza Ahsan
uahsan3@gatech.edu

Irfan Essa
irfan@cc.gatech.edu

Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Sharing user experiences in form of photographs, tweets, text, audio and/or video has become commonplace in social networking websites. Browsing through large collections of social multimedia remains a cumbersome task. It requires a user to initiate textual search query and manually go through a list of resulting images to find relevant information. We propose an automatic clustering algorithm, which, given a large collection of images, groups them into clusters of different events using the image features and related metadata. We formulate this problem as a kernel canonical correlation clustering problem in which data samples from different modalities or 'views' are projected to a space where correlations between the samples' projections are maximized. Our approach enables us to learn a semantic representation of potentially uncorrelated feature sets and this representation is clustered to give unique social events. Furthermore, we leverage the rich information associated with each uploaded image (such as usernames, dates/timestamps, etc.*) and empirically determine which combination of feature sets yields the best clustering score for a dataset of 100,000 images.*

1. Introduction

The recent growth of social media/networking sites such as Facebook, YouTube, Flickr and Instagram have led to new ways in which people share their experiences. Events ranging from social or political occurrences to natural disasters result in a large amount of multimedia being uploaded to social media platforms. We seek to develop an automatic approach to group images from different sources and users, who are at the same event. We leverage image features and available metadata to determine clusters of unqiue events. A unique event cluster is one that comprises all the images captured at that particular event. We cast this problem as a multi-view clustering problem, where each 'view' corresponds to each source of information such as titles of images and descriptions.

Clustering in high dimensions is a challenging problem due to the 'curse of dimensionality' and has resulted in proposals for projecting data samples onto fewer dimensions (dimensionality reduction) and clustering in the new reduced space. Principal Component Analysis (PCA) [10] addresses this by projecting data points to a lower-dimensional space where the points' variance is maximized. Random projection addresses this by projecting data samples to a lower dimensional space using a random matrix with unit length columns [4]. Canonical Correlation Analysis (CCA) [8] addresses this by projecting data samples to a lower-dimensional space such that the projected data samples' correlations are maximized. The disadvantage of using standard dimensionality reduction methods like PCA or random projections is that they don't take into account multiple modalities or 'views' of the data; they only preserve the pairwise distances/variances between the samples. We propose the use of kernel CCA [8] to reduce the dimensionality of social multimedia and learn a *reduced semantic representation of social events*. This learned space can then be clustered to produce unique social events.

CCA computes a set of canonical variates which are the orthogonal linear combinations of the features from two sources of information. The computed canonical variates (lower dimensional features) are representative of 'two views' since the computation is based on mutual correlations between data samples in the views. Thus the reduced feature space represents the underlying semantics of the data [8]. Any standard clustering algorithm such as k-means [12] can be applied in this space to determine groups of similar data points. Our approach to address the event clustering problem in social multimedia uses each image's visual content, user-provided data (such as titles, descriptions, usernames *etc.*), and automatically generated content (metadata from the camera) with social multimedia in order to group unique events.

Real world datasets are highly non-linear. Linearly projecting feature sets of social multimedia to a lower dimensional space and then clustering in that space do not yield meaningful results. Thus, we use kernel CCA [8] in order to map the feature sets to a high dimensional space, and then

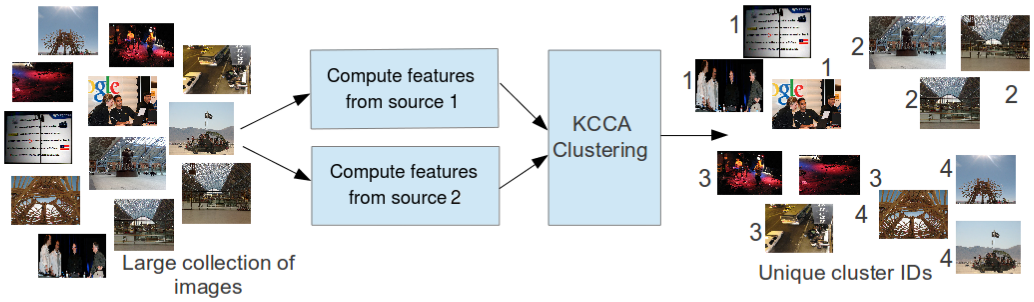
Large collection of images
Compute features from source 1
Compute features from source 2
KCCA Clustering
Unique cluster IDs
1
1
1
2
2
2
3
3
3
4
4
4

Figure 1. Overview of our proposed approach to aggregate unique event clusters

apply linear CCA in that space to obtain the lower dimensional feature space.

Specifically, we propose using kernel CCA to formulate social event aggregation as a multi-view clustering problem where we consider five different 'views' of the data: visual content, dates uploaded, usernames of people who shared the image, tags and titles/descriptions associated with an image. We perform kernel CCA on two views at a time and present an empirical comparison between different feature combinations. We do not rely on pairwise similarity metrics to group different event images because they do not scale well. To the best of our knowledge, this technique has not been applied in the social event clustering context to group unique event images.

Our main contributions are:

- Learning a semantic representation of social multimedia content via kernel CCA and clustering in the learned space.
- Determining the most discriminative feature representations which combine to give the best clustering score.

In the next section we describe the related work, followed by a brief overview of CCA and kernel CCA. We proceed to describe our approach and experiments and conclude with a discussion of our results.

2. Related Work

CCA was first proposed by [9], and then Hardoon *et al.* formulated canonical correlation analysis to solve an image retrieval problem and extended it to use kernels [8]. Multiview clustering [3] approaches have also been explored for data which can be divided into two (or more) attributes or 'views'.

Event clustering on social web has been addressed in a number of ways on many different platforms using text, such as Twitter and blogging platforms. Social event clustering using images and associated metadata has also been approached using different techniques. Becker et al. [2] learn document similarity metrics for different attributes of Flickr data to cluster social events. We also pose the social event detection problem as a clustering one, but do not attempt to learn similarity metrics for all attributes.

Kernel CCA with a clustering algorithm has been used in very few domains. Chaudhuri et al. [6] use CCA to cluster data samples generated from a mixture of Gaussian distributions and apply this to clustering Wikipedia articles and audio-visual speaker data. Trivedi *et al.* [17] use the same technique to cluster similar webpages by exploiting textual content and tags. Our work is different from these applications of kernel CCA-based clustering as we tackle a different problem; that of aggregating social multimedia to identify unique events, and we exploit a rich set of information along with images and text.

Blaschko and Lampert [5] have used kernel CCA with spectral clustering to perform image categorization using images with captions. They cluster images belonging to 9 categories, so the number of clusters is already known. In this paper, we have a completely unsupervised kernel CCA-based clustering approach. Furthermore, we address a problem where the number of clusters can potentially reach tens of thousands.

3. Kernel Canonical Correlation Analysis

We briefly present the theoretical foundations of kernel CCA and also introduce our notation. For more details, please see [8].

3.1. CCA

Canonical correlation analysis solves an eigenvector problem to determine the lower dimensional subspace on which the multiview data is projected [7]. Let $M = \{m_1, m_2, ..., m_n\}$ and $T = \{t_1, t_2, ..., t_n\}$ be the feature matrices corresponding to data from source 1 (say, visual data) and data from source 2 (say, textual data) where n is the number of event images. CCA determines two vectors

Figure 2. Sample event images from Social Event Detection dataset [15]

$\mathbf{v_m}$ and $\mathbf{v_t}$ with the constraint that the projected matrices' correlations with the vectors i.e. $\mathbf{v_m}^\top M$ and $\mathbf{v_t}^\top T$ are mutually maximized. Thus from [8],

$$\rho = \underset{\mathbf{v_m},\mathbf{v_t}}{\operatorname{argmax}} \left(\frac{\mathbf{v_m}^\top C_{mt} \mathbf{v_t}}{\sqrt{\mathbf{v_m}^\top C_{mm} \mathbf{v_m} \mathbf{v_t}^\top C_{tt} \mathbf{v_t}}} \right) \tag{1}$$

such that the following constraints are satisfied:

$$\mathbf{v_m}^\top C_{mm} \mathbf{v_m} = 1, \text{ and } \mathbf{v_t}^\top C_{tt} \mathbf{v_t} = 1$$

Here, C_{mm} and C_{tt} denote the within-sets covariance matrices of M and T respectively and C_{mt} refers to the between-set covariance matrix of M and T. Hence, the maximum canonical correlation is the maximum of ρ with respect to the two sets of directions $\mathbf{v_m}$ and $\mathbf{v_t}$. Forming the Lagrangian and computing the derivatives of it w.r.t $\mathbf{v_m}$ and $\mathbf{v_t}$ lead to the solution:

$$\mathbf{v_t} = \frac{C_{tt}^{-1} C_{tm} \mathbf{v_m}}{\lambda} \tag{2}$$

Here, we assume C_{tt} to be invertible. The solution for $\mathbf{v_m}$ is obtained by solving the generalized eigenproblem of the form $A\mathbf{m} = \lambda B\mathbf{m}$,

$$C_{mt} C_{tt}^{-1} C_{tm} \mathbf{v_m} = \lambda^2 C_{mm} \mathbf{v_m} \tag{3}$$

Thus, by solving Eq. 3 and substituting the values of $\mathbf{v_m}$ in Eq. 2 we can obtain the sequence of $\mathbf{v_t}$ and obtain the two canonical correlation vectors $\mathbf{v_m}$ and $\mathbf{v_t}$. These vectors correspond to the directions in which the image feature matrix M and text feature matrix T are projected to, such that their mutual correlations in the projected space is maximized.

3.2. Kernel CCA

Since linear CCA does not address the nonlinearities present in real world datasets, kernel CCA is used to first project the data onto a potentially infinite dimensional space and then linear CCA is applied in that space.

Now, $\mathbf{v_m}$ and $\mathbf{v_t}$ can be written as the projections onto the new lower dimensional space with the directions being α and β. Hence $\mathbf{v_m} = M^\top \alpha$ and $\mathbf{v_t} = T^\top \beta$.

Substituting the above expressions into Eq. 1 gives the following equation:

Algorithm 1 - Computing cluster IDs of event images

```
Input: Matrices $M$ and $T$ of size $n \times D_1$ and $n \times D_2$ respectively,
       Precision parameter $\eta$,
       Gaussian kernel parameter $\sigma$
       Number of clusters $k$ and
       Number of canonical variates $d$
Output: Cluster IDs for each event image
1: Project $M$ and $T$ to higher dimensional space $(n \times n)$ using Gaussian kernel
2: Perform kernel CCA on the two matrices to obtain the canonical variates $\alpha$'s and $\beta$'s.
3: Take $d$ canonical variates of both views and concatenate them.
4: Run k-means on the resulting matrix varying $k$, the number of clusters.
5: Choose the parameters $k$, $\eta$, $\sigma$ and $d$ with the best clustering score.
```

$$\rho = \max_{\alpha,\beta} \left(\frac{\alpha^\top M M^\top T T^\top \beta}{\sqrt{\alpha^\top M M^\top M M^\top \alpha . \beta^\top T T^\top T T^\top \beta}} \right) \tag{4}$$

If $K_m = MM^\top$ and $K_t = TT^\top$ are the kernel matrices corresponding to the two views, we can substitute their expression into Eq. 4, formulate it as an optimization problem and get the final form as:

$$\left(K_m + \kappa I\right)^{-1} K_t \left(K_t + \kappa I\right)^{-1} K_m \alpha = \lambda^2 \alpha \tag{5}$$

where κ is the regularization parameter.

To address computational issues with real world datasets, [8] decompose the kernel matrices using Partial Gram-Schmidt Orthogonalization (PGSO) [1] or equivalently, Incomplete Cholesky Decomposition (ICD). We tested both methods (PGSO and ICD) in early experiments. But in final experiments we use ICD only due to the computational complexities we faced using PGSO. Our multiview clustering algorithm is described in Algorithm 1

4. Social Event Clustering via kernel CCA

We aim to aggregate unique social events into event clusters. In our approach, the algorithm has to differentiate between similar event categories (such as concerts) happening at different times and places. We use the theoretical framework from kernel CCA and cluster in the resulting space.

4.1. Computing Image Features

Several image feature descriptors can be computed for each image in order to have a representation of image content and this feature descriptor matrix is one of the 'views' of the social event data. We obtain image feature descriptors via Scale Invariant Feature Transform (SIFT) descriptors [11]. One can also experiment with color- or texture-based features or holistic features such as GIST (to detect landmarks in order to identify location-based events as in [14].

4.2. Computing Textual Features

A simple way to represent text data is term frequencies. Text data associated with an image includes titles, descriptions, usernames of people who uploaded the image, dates and timestamps (if treated as text strings). If commonly occuring words are to be weighted less than unique words, one can also use term frequency inverse document frequency (TFIDF).

4.3. Kernel CCA and Clustering

The outputs of the feature building steps are two matrices which are input to the kernel CCA algorithm (See Figure 1) which outputs two sets of canonical variates. We do a parameter search and select the top d canonical variates for each feature combination.

4.4. Evaluation Metric

We evaluate the clustering algorithm's performance via normalized mutual information (NMI) [16].

4.4.1 Normalized Mutual Information (NMI)

Mutual information refers to the shared information between two random variables X and Y. If X and Y are independent, their mutual information will be zero.

Given two random variables X and Y, NMI is given by [16]:

$$NMI = \frac{I(X,Y)}{H(X)H(Y)}$$

where $I(X,Y)$ is the mutual information between X and Y, $H(X)$ is the entropy of X and $H(Y)$ is the entropy of Y. Concretely, the NMI formula can be re-written as [16]:

$$NMI = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} m_{ij} log\left(\frac{m m_{ij}}{m_i (\hat{m})_j}\right)}{\sqrt{\left(\sum_{i=1}^{m} m_i log\left(\frac{m_i}{m}\right)\right)\left(\sum_{j=1}^{m} (\hat{m})_j \, log\left(\frac{(\hat{m})_j}{m}\right)\right)}} \tag{6}$$

where m_i is the number of elements in ith cluster, $(\hat{m})_j$ is the number of elements in jth ground truth cluster and m_{ij} is the number of elements that are in both ith cluster as well as jth ground truth cluster.

We choose to use this metric because evaluating clustering algorithms based on information-theoretic methods is more reliable than methods based on counting pairs [18].

5. Experiments

5.1. Dataset

We test our approach on the Social Event Detection dataset provided by MediaEval2013 [15] and consists of around 300,000 publicly shared Flickr images uploaded between Janurary 2006 and December 2012. Each image represents a snapshot of an event which the user attended and then uploaded on Flickr (see Figure 2).

Each image has the following metadata: username, date taken/uploaded, title, description, tags and geotags (provided for only a quarter of the dataset). We do not use geotags in our experiments.

The ground truth of each image is a unique label which MediaEval2013 has provided. The dataset does not contain images that are associated with multiple social events.

5.2. Experimental setup

In experimenting with this dataset, our goal is to apply a multiview clustering technique which would yield competitive results in terms of cluster quality. Hence, we choose a subset of this dataset (100,000 images and their associated metadata) to test our clustering algorithm. To compute image features, we use the standard Bag of Words (BoW) model and build a visual vocabulary by selecting ~30,000 images as training images for computing the SIFT feature vector. Hence, we obtain each image's feature vector and compute the matrix of image features vectors to be input to the kernel CCA algorithm. We compute standard TFIDF vectors for each line of text associated with an image. The text is preprocessed by removing punctuation marks, stop words and HTML tags. We also take the dates and time stamps of the images and concatenate them as strings and use term frequency as features. Thus, we test the clustering performance for the following combinations: usernames and tags, text and tags, dates and text, dates and usernames, tags and dates, text and usernames, visual and text, visual and tags, visual and usernames, and visual and dates

The input to the algorithm are two matrices (for two views) and the output is two sets of canonical variates (vectors) to which the two matrices have been projected. We concatenate the top d directions obtained from the two views and perform standard k-means clustering on it to produce the final clustering result (See Algorithm 1).

5.3. Parameter Selection

We use the Gaussian kernel function in order to implicitly map the data samples to a high dimensional space, and perform CCA in that space. Besides that, we have a precision parameter which we set to 10^{-6} according to the suggestion of [8]. Furthermore, we need to choose number of clusters k and number of canonical variates d. In our experimental setup, we run a heuristic parameter search to select

Table 1. Clustering results using Gaussian kernel with best parameters for each feature combination

Feature types	k	σ	d	NMI
usernames+tags	5000	0.6	22	**0.9166**
text+tags	5000	2.1	12	0.8176
dates+text	3800	3.6	22	0.6110
dates+usernames	5000	3.6	22	0.7480
tags+dates	5000	3.6	22	0.6470
text+usernames	5000	4.1	12	0.8913
visual+text	5000	2.0	12	0.6230
visual+tags	5000	2.0	22	0.8117
visual+usernames	2000	1.5	02	**0.8933**
visual+dates	3500	0.5	22	0.5000

the most optimal parameters which gives the best clustering performance.

The Gaussian kernel, also known as Radial Basis Function (RBF) kernel is a non-linear mapping, parameterized by σ and given by:

$$K(\mathbf{X},\mathbf{Y}) = \exp\left(-\frac{\|\mathbf{X}-\mathbf{Y}\|_2^2}{2\sigma^2}\right)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are two sets of data samples.

The parameters we need to choose with Gaussian kernel based CCA clustering are: Gaussian parameter sigma (σ), number of clusters (k) and number of canonical variates (d). The best parameters for each combination of features are given in Table 1.

6. Results and Discussion

Table 1 shows the clustering performance (in terms of NMI scores) for different combinations of features computed on this dataset using Gaussian kernel. The results give us important insights about this problem and the proposed solution.

Choice of features We combine features from several different sources to leverage as much available information as possible. Also, we aim to empirically discover the most discriminative features in the social event data that yield a good clustering score. Table 1 is divided into two parts. The upper part shows results based on textual features, and the bottom part shows results based on the combination of visual and textual features. We note that textual feature combinations yield the best NMI score of 0.92. Visual features combined with text-based features give us the best NMI score of 0.89. There are several reasons for this performance.

Images shared on Flickr may have very different visual content, yet, may be taken at the same event, thus belonging to the same cluster (see Figure 3). Hence, local descriptors such as SIFT keypoints may be very different for the same event. This makes it challenging to decide which visual descriptor can work sufficiently well to discriminate between a majority of clusters. We also experiment with GIST [13] descriptors on this dataset, but do not get useful results, hence they are not included. Our intuition was that SIFT, being robust to changes in illumination, rotation and scale of the image will help in identifying unique event images, especially if images are captured of a stage where various artists perform their act and this scene is captured from different angles. However, social events are not restricted to performances on stage. Hence using SIFT or any visual descriptor that depends on the localized image content is not going to be highly useful for identifying unique events. Textual features are necessary for good clustering performance.

Having said that, in our experiments, combining SIFT with textual content (usernames in this dataset) associated with each event image yield the second highest NMI score when using Gaussian kernel compared to all other combinations. (See Table 1). Usernames in this dataset proved to be the most discriminative feature even though multiple users can upload images of the same event and vice versa. We believe that there is a certain level of certainty that a group of images are captured at the same event if the usernames associated with this group of images is the same. Dates and time stamps with the images do not yield good scores since this may not be a reliable source of information.

Titles/descriptions with the images and tags are also discriminative sources of information in the social event clustering task. In our experiments, tags are a better feature choice than text. This is not surprising as users find it easy to upload images, and add a few words as tags, rather than write a whole description and long titles with event images. Hence we see a higher NMI score with tags as one of two features as opposed to text.

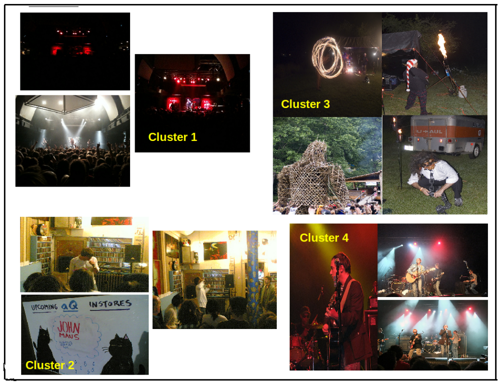

Figure 3. Images belonging to the same cluster (social event)

Choice of kernel Choosing the kernel function on a particular dataset determines the outcome of a learning problem. In this paper, we used Gaussian kernel because using a kernel which intuitively represents a similarity between the kernelized features is highly effective in a clustering task. It is simple to experiment with the Gaussian kernel since it has one parameter to tune. We determined σ through a search over possible choices of values.

Number of clusters 'k' Our approach is a completely unsupervised one and hence, choosing the number of clusters as a parameter is not a trivial task. We run a search through a range of possible values of k starting from 1000 for each feature combination. The optimal value for most of our experiments turns out to be roughly equivalent to the true total number of clusters. Our experimentally determined value for k is 5000 (for most feature combinations), and the true number of clusters (social events) in our data is around 4700.

Number of canonical variates 'd' Kernel CCA maps the original data points first to an infinite dimensional space and then to a reduced lower dimensional space. The reduced space is a set of vectors or directions to which the original data is mapped. The number of these vectors correspond to d. To determine the optimal d, we again run a search over possible values and experimentally determine that for obtaining optimal clustering scores, the top 12-22 canonical variates are sufficient to yield the social event clusters.

7. Conclusion and Future Work

In this paper, we propose the technique of kernel CCA to partition a set of event images into unique social events, via two sets of features or 'views' of the data. We show results for various combinations of features and conclude that the usernames, tags and visual features combined with the textual, are discriminative feature combinations for obtaining unique social event clusters.

We also empirically determine that visual features alone of an image are not discriminative enough to differentiate between two different social events (especially, if the two events belong to the same category such as concerts).

We aim to take this work further and scale it to millions of images for a potential improvement in clustering results. For scaling this alogrithm, we will experiment with incremental clustering algorithms so that this technique can be applied to social streaming multimedia.

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2003.

[2] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300. ACM, 2010.

[3] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

[4] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[5] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[6] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136. ACM, 2009.

[7] C.-L. Chen, Y.-C. Gong, and Y.-J. Tian. Kck-means: A clustering method based on kernel canonical correlation analysis. In *Computational Science–ICCS 2008*, pages 995–1004. Springer, 2008.

[8] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[9] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

[10] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] J. MacQueen. Some methods for classification and analysis of multivariate observations, 1967.

[13] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[14] S. Papadopoulos, C. Zigkolis, Y. Kompatsiaris, and A. Vakali. Cluster-based landmark and event detection on tagged photo collections. *IEEE Multimedia*, 2010.

[15] T. Reuter, S. Papadopoulos, G. Petkos, V. Mezaris, Y. Kompatsiaris, P. Cimiano, C. de Vries, and S. Geva. Social event detection at mediaeval 2013: Challenges, datasets, and evaluation. In *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop Barcelona, Spain, October 18-19, 2013*, 2013.

[16] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[17] A. Trivedi, P. Rai, S. L. DuVall, and H. Daumé III. Exploiting tag and word correlations for improved webpage clustering. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 3–12. ACM, 2010.

[18] S. Wagner and D. Wagner. *Comparing clusterings: an overview*. Universität Karlsruhe, Fakultät für Informatik, 2007.