Leveraging Social Bookmarks from Partially Tagged Corpus for Improved Webpage Clustering

Anusua Trivedi
School of Computing, University of Utah, Salt Lake City - 84112 (UT)
Piyush Rai
School of Computing, University of Utah, Salt Lake City - 84112 (UT)
Hal Daumé III
Department of Computer Science, University of Maryland, College Park - 20742 (MD)
and
Scott L. DuVall
VA SLC Health Care System & University of Utah, Salt Lake City - 84112 (UT)

Automatic clustering of webpages helps a number of information retrieval tasks, such as improving user interfaces, collection clustering, introducing diversity in search results, etc. Typically, webpage clustering algorithms only use features extracted from the page-text. However, the advent of social-bookmarking websites, such as StumbleUpon.com and Delicious.com, has led to a huge amount of user-generated content such as the social tag information that is associated with the webpages. In this paper, we present a subspace based feature extraction approach which leverages the social tag information to complement the page-contents of a webpage for extracting beter features, with the goal of improved clustering performance. In our approach, we consider page-text and tags as two separate views of the data, and learn a shared subspace that maximizes the correlation between the two views. Any clustering algorithm can then be applied in this subspace. We then present an extension that allows our approach to be applicable even if the webpage corpus is only partially tagged, i.e., when the social tags are present for not all, but only for a small number of webpages. We compare our subspace based approach with a number of baselines that use tag information in various other ways, and show that the subspace based approach leads to improved performance on the webpage clustering task. We also discuss some possible future work including an active learning extension that can help in choosing which webpages to get tags for, if we only can get the social tags for only a small number of webpages.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Clustering—*documentation*

General Terms: Algorithms

Additional Key Words and Phrases: Webpage Clustering, Social Bookmarking, Information Retrieval, Canonical Correlation Analysis, Kernel Methods

Author's address: Anusua Trivedi, 50 S Central Campus Drive, School of Computing (Room 3190), University of Utah, Salt Lake City, UT 84112.

1. INTRODUCTION

The world-wide-web contains a wealth of information in amounts so enormous that it may seem daunting at first to be able to mine any useful information one is looking for. Fortunately, web mining techniques such as clustering help to organize the web content into appropriate subject-based categories so that their efficient search and retrieval becomes manageable.

Traditional webpage clustering typically uses only the page content information (usually, just the page text) in an appropriate feature vector representation such as Bag of Words, TF-IDF, etc., and then applies standard clustering algorithms (e.g., K-means algorithm [McQueen 1967], spectral clustering [von Luxburg 2007], etc.). Another approach somewhat related to clustering is to mine topic information from documents collections (e.g., Latent Dirichlet Allocation [Blei et al. 2003]), which can be seen as clustering words occurring in each document (instead of clustering documents directly).

On the one hand, the proliferation of the world-wide-web presents ever increasing challenges for the search engines to cope with task of mining the humongous wealth of available information on the web nowadays. On the other hand, the increasing amounts of user-generated content nowadays nicely complements this information and can help in an effective mining of the data present on the web. For example, users can provide captions for images on the internet, provide tags to webpages and other media content they regularly browse on the internet, etc. Therefore such user-generated content can provide useful information in various form such as meta-data, or in more explicit ways such as tags.

User specified *social tags*, in particular, have proven to be extremely effective in browsing, organizing, and indexing of webpages. Various social bookmarking websites such as StumbleUpon and Delicious allow users to tag webpages with keywords or short text snippets that can provide a description of the webpages. Users can collaboratively tag webpages and this has made organizing, sharing, navigating, and retrieving web content much easier than ever before. In this work, we aim to exploit the tag information for a web-mining task, namely webpage clustering.

Since user provided tags can often provide high-level, contexual information for webpages, we want to exploit them by treating the tag information as an alternate *view* of the data. Motivated by the success of multi-view learning algorithms [Blum and Mitchell 1998; Brefeld and Scheffer 2004; Muslea et al. 2002; Bickel and Scheffer 2004; Ando and Zhang 2007; Kakade and Foster 2007] in various machine learning tasks, we use two views of the data (page-text and social tags) to extract highly discriminative features and perform clustering using these features. The feature extraction amounts to performing clustering in a lower dimensional subspace which is also effective in dealing with the problem of overfitting when we only have a small number of documents having a very large number of features. In particular, we use a regularized variant of the Kernel Canonical Correlation Analysis [Hotelling 1936; Gestel et al. 2001; Hardoon et al. 2004] (KCCA) algorithm to learn this subspace. KCCA (and Canonical Correlation Analysis - CCA - in general) has received tremendous attention due to its ability for effectively extracting useful features from heterogeneous or parallel data sources, such as images and text [Socher and Fei-Fei

2010], or features and labels (supervised dimensionality reduction [Rai and Daumé III 2009; Ji et al. 2008]). Therefore such an approach is expected to be useful for extracting useful features in the case of webpage clustering as well since such data natually comes with multiple views (page-text and social tags in our case).

One problem with the most existing multiview learning algorithms is that they require all the views to be *complete*, i.e., present for all the examples. This may however not always be the case. For example, in the context of social bookmarking datasets, user tags may be available only for a small subset of webpages. One way to apply multiview learning algorithms in this setting would be to first try to predict (using some classification algorithm) the set of social tags for each non-tagged webpage. This can however be very expensive since the set of possible tags ("labels") can be really large (equal to the tag vocabulary size).

This limitation makes it necessary to develop multiview algorithms that can work even with *incomplete* view information. With this motivation, we also present an extension of our kernel CCA based approach approach to deal with missing views. Our approaches to deal with missing views is based on the fact that the similarity between a pair of examples should be the same across all the views. In particular, we show how the kernel CCA based approach to multiview clustering [Chaudhuri et al. 2009] can be used in situations when only one view (the primary view) is complete whereas the other view(s) could potentially be *incomplete*, i.e., features from such view(s) are available only for a small number of examples. Our approach does not require computing the explicit features in the incomplete views (e.g., we do not require the tags to be predicted for the non-tagged webpages). In particular, we take the kernel variant of CCA [Hardoon et al. 2004] which works on the kernel matrices defined over each view, and propose a way to construct the *full* kernel matrix corresponding to the incomplete view, given the other complete view. This is followed by applying the kernel CCA based multiview clustering algorithm. Our presentation is based on the kernel CCA based multiview clustering but our approach can also be applied to other kernel based multiview clustering algorithms [de Sa 2005].

Rest of the paper is organized as follows. In Section 2, we describe the general framework we are considering in this paper. Section 3 briefly describes multi-view learning algorithms. Section 3.1 and Section 3.2 describe CCA and kernel CCA respectively. Section 4 describes our approach for dealing with the incomplete views in the kernel CCA setting. Our results are described in Section 5. We discuss related work in Section 6. In Section 7, we briefly describe some possible future work, including an active learning [Settles 2009] extension that can help in choosing which webpages to get the social tags for, if we can get the social tags for only a small number of webpages. We conclude with Section 8.

2. WEBPAGE CLUSTERING USING TAGS

Our problem setting consists of a collection of webpages where each page also has a set of user-specified tags (e.g., from social bookmarking websites such as Delicious or StumbleUpon). The goal is to obtain a clustering of the webpages into semantically relevant categories. To assess the relevance and coherency of the discovered clusters, one can use hierarchical web directories such as the Open Directory Project (ODP) as the gold standard. Web directories such as ODP are widely acceptable gold standards because they usually provide an agreed-upon clustering of webpages by human users, and have been used for evaluations in various recent works [Ramage et al. 2009; Lu et al. 2009].

In this paper, we study vector space models for clustering in which each document (a webpage) is represented using a feature vector derived from the page-text (and, if available, other contextual information, such as tags, which we consider in this paper). The K-means algorithm is a popular vector space model for flat-clustering which works iteratively by assigning each data point to its nearest cluster center, recomputing the cluster centers, and repeating the process until convergence. In this paper, we use the K-means algorithm for our evaluations. Our approach, however, is applicable to any vector space clustering algorithm.

Formally, for our clustering task, we are given a collection of N webpages, with each webpage consisting of a bag of words from a word vocabulary W, and a bag of tags from a *tag vocabulary* T. The goal is to cluster the webpages in K clusters where K is the desired number of clusters. It is also to be noted that the tag vocabulary is expected to have very little overlap with the word vocabulary since the social tags assigned to a webpage are usually words conveying contexual and semantic information about the webpage. Therefore, in most cases, the words used for social tags are not part of the webpages.

There are a number of ways in which the vector space algorithms such as K-means can exploit the tag information to improve clustering of webpages. Some of the common choices are [Ramage et al. 2009; Lu et al. 2009]:

(1) Words Only: Discard the tag information (use only bag of words in page-text).

(2) Tags Only: Discard the word information (use only bag of tags).

(3) Words + Tags: Form a combined bag of both words and tags, and use it to derive feature vectors for each document

(4) Word Vector + Tag Vector: Form two separate feature vectors (e.g., in bag of words representations) for words and tags using word vocabulary W and tag vocabulary T respectively, and concatenate the two feature vectors (with appropriate weighing of the two parts [Ramage et al. 2009]).

It turns out [Ramage et al. 2009; Lu et al. 2009] that the concatenation of word and tag feature vectors (4) outperforms approaches that use feature vectors derived from the word (1) vocabulary, the tag vocabulary (2), or vocabulary derived from a union of words and tags (3).

However, the concatenation approach inflates the feature vector size of each document, and therefore the approach tends to not do well if the number of webpages is small as compared to the feature dimensionality [Kriegel et al. 2009]. The reason can be attributed to the fact that clustering, and density estimation in general, can yield poor parameter estimates if the number of features far exceeds the number of data points. Furthermore, one would expect that there would be a significant correlation between the words and the tags for a given webpage and the concatenation based approach fails to exploit this correlation. Also, the relative importance of features in the tags and words views of the concatenated vector can be different which may require an explicit weighting of features in the two views [Ramage et al.

2009].

A number of efficient clustering algorithms deal with high data dimensionality by first projecting the high dimensional data onto a lower dimensional subspace, and then performing clustering in that subspace. The projection step is usually performed using standard dimensionality reduction techniques such as principal component analysis [Vempala and Wang 2002] (PCA), or random projections [Dasgupta 1999]. However, PCA or random projections only preserve the data variances or pairwise distances and fail to take advantage of multiple views of the data (if such information is available). Also note that even if PCA is performed on the joint words + tags vector, it would only maximize the variances of word and tag feature spaces individually, without capturing their correlations.

3. MULTI-VIEW LEARNING

In multi-view learning, the features can be split into two subsets such that each subset alone is sufficient for learning. By exploiting both views of the data, multi-view learning can result in improved performance on various learning tasks, both supervised and unsupervised [Brefeld and Scheffer 2004; Muslea et al. 2002; Bickel and Scheffer 2004; Ando and Zhang 2007; Kakade and Foster 2007; Foster et al. 2008]. Multi-view approaches help supervised learning algorithms by being able to leverage unlabeled data [Blum and Mitchell 1998], whereas, for unsupervised learning algorithms, multiple views of the data can often help in extracting better features [Foster et al. 2008].

Canonical Correlation Analysis [Hotelling 1936] (CCA) is an unsupervised feature extraction technique for finding dependencies between two (or more) views of the data by maximizing the correlations between the views in a shared subspace. This property makes CCA a suitable choice for multi-view learning algorithms. In our settings, the two views are words in the page-text, and the set of tags for each webpage. CCA is then applied as a projection technique to extract features from webpage data, with projection direction guided by the tag information. Final clustering is then performed using the features extracted by CCA.

3.1 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a technique for modeling the relationships between two (or more) set of variables. CCA computes a low-dimensional *shared* embedding of both sets of variables such that the correlations among the variables between the two sets is maximized in the embedded space. CCA has been applied with great success in the past on a variety of learning problems dealing with multi-modal data [Hardoon and Shawe-taylor 2003; Hardoon et al. 2004; Rustandi et al. 2009].

More formally, given a pair of datasets $\mathbf{X} \in \mathbb{R}^{D_1 \times N}$ and $\mathbf{Y} \in \mathbb{R}^{D_2 \times N}$, CCA seeks to find linear projections $\mathbf{w}_x \in \mathbb{R}^{D_1}$ and $\mathbf{w}_y \in \mathbb{R}^{D_2}$ such that, after projecting, the corresponding examples in the two datasets are maximally correlated in the projected space. The correlation coefficient between the two datasets in the embedded space is given by

$$\rho = \frac{\mathbf{w}_x^T \mathbf{X} \mathbf{Y}^T \mathbf{w}_y}{\sqrt{(\mathbf{w}_x^T \mathbf{X} \mathbf{X}^T \mathbf{w}_x)(\mathbf{w}_y^T \mathbf{Y} \mathbf{Y}^T \mathbf{w}_y)}} \tag{1}$$

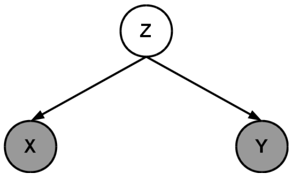


Fig. 1. The dependency view of CCA: Coupled datasets X and Y, and their shared subspace defined by Z. In our webpage clustering setting, X corresponds to the features derived from the page-text and Y corresponds to the features derived from the tags. Z represents the semantic subspace shared by both words and tags.

Since the correlation is not affected by rescaling of the projections $\mathbf{w}_x$ and $\mathbf{w}_y$, CCA is posed as a constrained optimization problem.

$$\max_{\mathbf{w}_x,\mathbf{w}_y} \mathbf{w}_x^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_y \tag{2}$$

subject to:

$$\mathbf{w}_x^T \mathbf{X}\mathbf{X}^T \mathbf{w}_x = 1, \mathbf{w}_y^T \mathbf{Y}\mathbf{Y}^T \mathbf{w}_y = 1$$

It can be shown [Hardoon et al. 2004] that the above formulation is equivalent to solving the following generalized eigen-value problem:

$$\begin{pmatrix} 0 & \mathbf{\Sigma_{xy}} \\ \mathbf{\Sigma_{yx}} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w_x} \\ \mathbf{w_y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{\Sigma_{xx}} & 0 \\ 0 & \mathbf{\Sigma_{yy}} \end{pmatrix} \begin{pmatrix} \mathbf{w_x} \\ \mathbf{w_y} \end{pmatrix}$$

where $\mathbf{\Sigma_{xx}}$ and $\mathbf{\Sigma_{yy}}$ denotes the covariances of data samples $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ respectively, and $\mathbf{\Sigma_{xy}}$ denotes the cross-covariance between $\mathbf{X}$ and $\mathbf{Y}$.

3.2 Kernel CCA

Canonical Correlation Analysis is a linear feature extraction algorithm. Many real world datasets, however, exhibit nonlinearities, and therefore a linear projection may not be able to capture the properties of the data. Kernel methods [Shawe-Taylor and Cristianini 2004] give us a way to deal with the nonlinearities by mapping the data to a higher (potentially infinite) dimensional space and then applying linear methods in that space (e.g., Support Vector Machines [Burges 1998] for classification, Kernel Principal Component Analysis [Schölkopf et al. 1998] for dimensionality reduction). The attractiveness of kernel methods is attributed to the fact that this mapping need not be computed explicitly, via the technique call the *kernel trick* [Shawe-Taylor and Cristianini 2004].

The kernel variant of CCA (called Kernel Canonical Correlation Analysis - KCCA) can be thought of as first (implicitly) mapping each D dimensional data point $\vec{x}$ to a higher dimensional space $\mathcal{F}$ defined by a mapping ϕ whose range is in an inner product space (possibly infinite dimensional), followed by applying linear CCA in the feature space $\mathcal{F}$.

To get the kernel formulation of CCA, we switch to the dual representation [Hardoon et al. 2004] by expressing the projection directions in Equation 1 as $\mathbf{w}_x = \mathbf{X}\boldsymbol{\alpha}$ and $\mathbf{w}_y = \mathbf{Y}\boldsymbol{\beta}$ where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are vectors of size N. The dual formulation of Equation 1 is given by:

$$\rho = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\boldsymbol{\alpha}^T \mathbf{X}^T \mathbf{X} \mathbf{Y}^T \mathbf{Y} \boldsymbol{\beta}}{\sqrt{\boldsymbol{\alpha}^T \mathbf{X}^T \mathbf{X} \mathbf{X}^T \mathbf{X} \boldsymbol{\alpha} \times \boldsymbol{\beta}^T \mathbf{Y}^T \mathbf{Y} \mathbf{Y}^T \mathbf{Y} \boldsymbol{\beta}}} \tag{3}$$

Now using the fact that $\mathbf{K}_x = \mathbf{X}^T\mathbf{X}$ and $\mathbf{K}_y = \mathbf{Y}^T\mathbf{Y}$ are the kernel matrices for $\mathbf{X}$ and $\mathbf{Y}$, kernel CCA amounts to solving the following problem:

$$\rho = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\boldsymbol{\alpha}^T \mathbf{K}_x \mathbf{K}_y \boldsymbol{\beta}}{\sqrt{\boldsymbol{\alpha}^T \mathbf{K}_x^2 \boldsymbol{\alpha} \times \boldsymbol{\beta}^T \mathbf{K}_y^2 \boldsymbol{\beta}}} \tag{4}$$

subject to the following constraints $\boldsymbol{\alpha}^T \mathbf{K}_x^2 \boldsymbol{\alpha} = 1$ and $\boldsymbol{\beta}^T \mathbf{K}_y^2 \boldsymbol{\beta} = 1$.

KCCA works by using the kernel matrices $\mathbf{K}_x$ and $\mathbf{K}_y$ of the examples in the two views $\mathbf{X}$ and $\mathbf{Y}$ of the data. This is in contrast with linear CCA which works by doing an eigen-decomposition of the covariance matrix. The eigenvalue problem for kernel CCA is given by:

$$\begin{pmatrix} 0 & \mathbf{K}_x\mathbf{K}_y \\ \mathbf{K}_y\mathbf{K}_x & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{K}_x^2 & 0 \\ 0 & \mathbf{K}_y^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \tag{5}$$

For the case of linear Kernel, KCCA reduces to the standard CCA. However, working under the kernel formalism has the additional advantage of being computationally efficient if the number of features greatly exceeds the number of examples because KCCA works on $N \times N$ kernel matrices, whereas CCA works on $D \times D$ covariance matrices. The former would be much more efficient than the latter if $D \gg N$, which is usually the case with document clustering where the vocabulary size often far exceeds the number of documents.

3.3 Regularization in KCCA

To avoid overfitting and trivial solutions (non-relevant solutions), CCA literature [Shawe-Taylor and Cristianini 2004; Hardoon et al. 2004] suggests regularizing the projection directions $\mathbf{w}_x$ and $\mathbf{w}_y$ by penalizing them using Partial Least Squares (PLS) which basically means that their high weights are penalized. This is achieved by adding regularization terms corresponding to $\mathbf{w}_x$ and $\mathbf{w}_y$ in the denominator of Equation 4.

$$\begin{aligned} \rho &= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\boldsymbol{\alpha}^T \mathbf{K}_x \mathbf{K}_y \boldsymbol{\beta}}{\sqrt{(\boldsymbol{\alpha}^T \mathbf{K}_x^2 \boldsymbol{\alpha} + \kappa||\mathbf{w}_x||^2)(\boldsymbol{\beta}^T \mathbf{K}_y^2 \boldsymbol{\beta} + \kappa||\mathbf{w}_y||^2)}} \\ &= \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\boldsymbol{\alpha}^T \mathbf{K}_x \mathbf{K}_y \boldsymbol{\beta}}{\sqrt{(\boldsymbol{\alpha}^T \mathbf{K}_x^2 \boldsymbol{\alpha} + \kappa \boldsymbol{\alpha}^T \mathbf{K}_x \boldsymbol{\alpha})(\boldsymbol{\beta}^T \mathbf{K}_y^2 \boldsymbol{\beta} + \kappa \boldsymbol{\beta}^T \mathbf{K}_y \boldsymbol{\beta})}} \end{aligned}$$

Since the above equation is invariant to scaling of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we impose the following constraints on the denominator terms of the above equation:

$$\boldsymbol{\alpha}^T \mathbf{K}_x^2 \boldsymbol{\alpha} + \kappa \boldsymbol{\alpha}^T \mathbf{K}_x \boldsymbol{\alpha} = 1$$

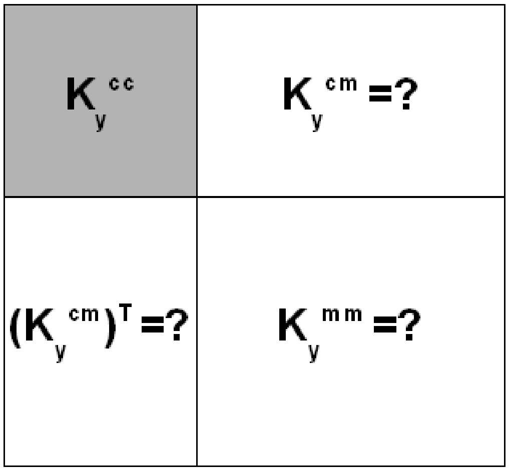
$$\boldsymbol{\beta}^T \mathbf{K}_y^2 \boldsymbol{\beta} + \kappa \boldsymbol{\beta}^T \mathbf{K}_x \boldsymbol{\beta} = 1$$

3.4 Computational Issues

Kernel CCA relies on the decomposition of kernel matrices which can be an expensive operation as the number of examples grows. To deal with this, one can use Incomplete Cholesky Decomposition [Bach and Jordan 2003] (ICD). We, on the other hand, use Partial Gram-Schmidt Orthogonalization (PGSO) as suggested in [Hardoon et al. 2004]. Incomplete Cholesky method can be seen as a dual implementation of PGSO. The advantage of PGSO over ICD is that the former does not require permutations of rows and columns unlike the latter.

4. KERNEL CCA WITH INCOMPLETE VIEWS

One shortcoming of both CCA and KCCA is that they assume that features across all views are available for each example. This may however not be the case with many multiview datasets. For example, not all webpages in a corpus might be tagged by users. Likewise, not all webpages can be expected to have hyperlinks pointing towards them. Therefore, although one view (i.e., page-text) would be available for all the webpages, the other view might be available only for a small number of webpages. To apply multiview clustering on such datasets, one needs a way to deal with the lack of data in the incomplete view(s). In this section, we present an approach to address this shortcoming for KCCA. The problem for standard CCA can also be dealt with by using KCCA with a linear kernel. Also, our approach is not limited to kernel CCA based multiview clustering. It can also be used for other kernel based multiview clustering algorithms such as the multiview spectral clustering [de Sa 2005].



K_y^{cc}	$K_y^{cm} =?$
$(K_y^{cm})^T =?$	$K_y^{mm} =?$

Fig. 2. Completing the full kernel matrix using the incomplete view $\mathcal{Y}$

Note that KCCA works by first constructing the kernel matrix for each view of the data. For simplicity, let us denote the two views by $\mathcal{X}$ and $\mathcal{Y}$. Generalization to more than two views with one complete and remaining incomplete views can be done in a likewise manner. Let us assume that view $\mathcal{X}$ is complete whereas view $\mathcal{Y}$ is incomplete, i.e., the features for this view are available for only a subset of the total examples. To formalize, we denote the set of webpages with features present in both the views $\mathcal{X}$ and $\mathcal{Y}$ (i.e., the fully paired or *complete*) as $\mathcal{C} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_c, \mathbf{y}_c)\}$ and the set of webpages with features present only in view $\mathcal{X}$ (i.e., unpaired or *missing*) as $\mathcal{M} = \{\mathbf{x}_{c+1}, \ldots, \mathbf{x}_{c+m}\}$. Let us denote by $\mathbf{K}_x$ the $(c+m) \times (c+m)$ kernel matrix defined over all the examples using features from view $\mathcal{X}$. The corresponding graph Laplacian defined as $\mathcal{L}_x = \mathbf{D}_x - \mathbf{K}_x$, where $\mathbf{D}_x$ is the diagonal matrix consisting of the row sums of $\mathbf{K}_x$ along its diagonals.

Likewise, for view $\mathcal{Y}$, we denote the kernel matrix by $\mathbf{K}_y$. However, since features for view $\mathcal{Y}$ are only available for a small number of examples, only an $c \times c$ subblock of the full kernel matrix $\mathbf{K}_y$ will be available for this view (see Figure 2). In order to apply kernel CCA, one must first construct the full kernel matrix $\mathbf{K}_y$. Using the ideas from Laplacian regularization, this can be achieved by solving the following optimization problem for kernel matrix completion:

$$\min_{\mathbf{K}_y \succeq 0} tr(\mathcal{L}_x \mathbf{K}_y) \qquad (2)$$
$$s.t. \mathbf{K}_y(i,j) = k(\mathbf{y}_i, \mathbf{y}_j),\ \forall 1 \leq (i,j) \leq c$$

The objective above optimizes w.r.t. $\mathbf{K}_y$ the alignment between $\mathbf{K}_x$ and $\mathbf{K}_y$, *given the known part of* $\mathbf{K}_y$. Here tr denotes the matrix trace. Note that although the multiview assumption requires the views to be conditionally independent, since both views are just expressing different representations of the *same* object, both kernel matrices $\mathbf{K}_x$ and $\mathbf{K}_y$ are still expected to have a high degree of alignment between them.

The positive semi-definite constraint on the kernel matrix $\mathbf{K}_y$ makes it a semi-definite program (SDP) [Boyd and Vandenberghe 2004], which can be solved using the existing SDP solvers. One problem with the SDP based solvers is their lack of scalability to a large number of examples. Although the scalability can still be dealt with using first order solvers such as SDPNAL [yuan Zhao et al. 2010], assessing convergence can be an issue with such approaches. In this paper, we take a different approach and, due to the special problem structure (i.e., upper left sub-block of $\mathbf{K}_y$ being known), we can in fact obtain a *closed-form* solution for $\mathbf{K}_y$. Furthermore, our approach is much less computationally intensive than having to solve an SDP since, as we will show, it only requires a couple of matrix multiplication and inverses. A similar approach was proposed in [Carreira-Perpinan and Lu 2007] to compute the embedding of out-of-sample datapoints in Laplacian eigenmaps.

We denote $\mathbf{K}_y(i,j) = k(y_i, y_j),\ \forall 1 \leq (i,j) \leq c$ in Equation (2) as $\mathbf{K}_y^{cc}$, the $c \times c$ kernel matrix for the set of examples with view $\mathcal{Y}$ available.

Since $\mathbf{K}_y$ is a positive semi-definite matrix, we can express it as $\mathbf{A}\mathbf{A}^T$ where $\mathbf{A}$ is a matrix of reals. Further, let us write $\mathbf{A}$ as $\mathbf{A} = \binom{\mathbf{A}_c}{\mathbf{A}_m}$, and $\mathcal{L}_x$ as:

$$\mathcal{L}_x = \begin{bmatrix} \mathcal{L}_x^{cc} & \mathcal{L}_x^{cm} \\ (\mathcal{L}_x^{cm})^T & \mathcal{L}_x^{mm} \end{bmatrix}$$

Using these, we can rewrite Equation (2) as follows:

$$\min_{\mathbf{A}} tr(\mathcal{L}_x \mathbf{A}\mathbf{A}^T) = \min_{\mathbf{A}} tr(\mathbf{A}^T \mathcal{L}_x \mathbf{A}) = \min_{\mathbf{A}_c, \mathbf{A}_m} tr\Big(\binom{\mathbf{A}_c}{\mathbf{A}_m}^T \begin{bmatrix} \mathcal{L}_x^{cc} & \mathcal{L}_x^{cm} \\ (\mathcal{L}_x^{cm})^T & \mathcal{L}_x^{mm} \end{bmatrix} \binom{\mathbf{A}_c}{\mathbf{A}_m} \Big)$$

Simplifying the above, and using the fact that $\mathbf{A}_c$ is a constant (since $\mathbf{A}_c\mathbf{A}_c^T = \mathbf{K}_y^{cc}$, a constant), gives:

$$\min_{\mathbf{A}_m} tr(\mathbf{A}_c^T\mathcal{L}_x^{cc}\mathbf{A}_c + \mathbf{A}_c^T\mathcal{L}_x^{cm}\mathbf{A}_m + \mathbf{A}_m^T(\mathcal{L}_x^{cm})^T\mathbf{A}_c + \mathbf{A}_m^T\mathcal{L}_x^{mm}\mathbf{A}_m)$$

Using the matrix trace property $tr(X) = tr(X^T)$, one can see that the above reduces to:

$$\min_{\mathbf{A}_m} tr(\mathbf{A}_c\mathbf{A}_c^T\mathcal{L}_x^{cc}) + 2 * tr(\mathbf{A}_c^T\mathcal{L}_x^{cm}\mathbf{A}_m) + tr(\mathbf{A}_u^T\mathcal{L}_x^{mm}\mathbf{A}_m)$$

Again, using the fact $\mathbf{A}_c\mathbf{A}_c^T = \mathbf{K}_y^{cc}$, we write the above as:

$$\min_{\mathbf{A}_m} tr(\mathbf{K}_y^{cc}\mathcal{L}_x^{cc}) + 2 * tr(\mathbf{A}_c^T\mathcal{L}_x^{cm}\mathbf{A}_m) + tr(\mathbf{A}_m^T\mathcal{L}_x^{mm}\mathbf{A}_m) = \min_{\mathbf{A}_m} 2 * tr(\mathbf{A}_c^T\mathcal{L}_x^{cm}\mathbf{A}_m) + tr(\mathbf{A}_m^T\mathcal{L}_x^{mm}\mathbf{A}_m)$$

Taking the derivative w.r.t. $\mathbf{A}_m$ and setting it to zero gives:

$$2 * (\mathcal{L}_x^{cm})^T\mathbf{A}_c + 2 * \mathcal{L}_x^{mm}\mathbf{A}_m = 0$$

Therefore $\mathbf{A}_m = -(\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{A}_c$, and $\mathbf{A} = \left(\begin{smallmatrix}\mathbf{A}_c \\ \mathbf{A}_m\end{smallmatrix}\right) = \left(\begin{smallmatrix}\mathbf{A}_c \\ -(\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{A}_c\end{smallmatrix}\right)$

Using $\mathbf{K}_y = \mathbf{A}\mathbf{A}^T$ gives us the closed-form expression for $\mathbf{K}_y$:

$$\mathbf{K}_y = \left(\begin{smallmatrix}\mathbf{A}_c\mathbf{A}_c^T & -\mathbf{A}_c\mathbf{A}_c^T\mathcal{L}_x^{cm}(\mathcal{L}_x^{mm})^{-1} \\ -(\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{A}_c\mathbf{A}_c^T & (\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{A}_c\mathbf{A}_c^T\mathcal{L}_x^{cm}(\mathcal{L}_x^{mm})^{-1}\end{smallmatrix}\right)$$

Finally, substituting back for $\mathbf{A}_c\mathbf{A}_c^T = \mathbf{K}_y^{cc}$ gives:

$$\mathbf{K}_y = \left(\begin{smallmatrix}\mathbf{K}_y^{cc} & -\mathbf{K}_y^{cc}\mathcal{L}_x^{cm}(\mathcal{L}_x^{mm})^{-1} \\ -(\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{K}_y^{cc} & (\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T\mathbf{K}_y^{cc}\mathcal{L}_x^{cm}(\mathcal{L}_x^{mm})^{-1}\end{smallmatrix}\right)$$

Having obtained the full kernel matrix $\mathbf{K}_y$ for all $c + m$ examples on view $\mathcal{Y}$, we can now apply kernel CCA on the two kernel matrices $\mathbf{K}_x$ and $\mathbf{K}_y$, and use the extracted features in any off-the-shelf clustering algorithm such as k-means.

5. EXPERIMENTS

For our experiments, we compare our CCA based approach against a number of baselines, and show that accounting for the correlations between tags and words helps in extracting better features which lead to improved clustering performance. The K-means algorithm is chosen as the base clustering algorithm for all the approaches considered in the paper. Any other vector-space clustering algorithm can also be used however. Since K-means is sensitive to initialization, we repeated each experiment 20 times and have reported the average scores with standard deviations. Section 5.2 describes the experiments with the fully tagged corpus and Section 5.3 describes the experiments with the partially tagged corpus.

5.1 Datasets

Our dataset consists of a collection of 2000 tagged webpages that we use for our webpage clustering task. All webpages in our collection were downloaded from

URLs that are present in both the Open Directory Project (ODP) web directory (so that their ground-truth clustering are available) and Delicious social bookmarking website (so that their tag information is available). The Delicious dataset of tags is available here: `http://kmi.tugraz.at/staff/markus/datasets/`

Each webpage that we crawled and downloaded was tagged by a number of users on Delicious. Therefore, for each webpage, we combine the tags assigned to it by all users who tagged that webpage.

After stemming and stop-word removal, we had a page text vocabulary of 70168 unique words and a tag vocabulary (set of all unique tags) of 4328 unique tags. These are essentially the sizes for the page-text based and tag based feature vectors respectively. We used the bag-of-words representation for the feature vectors. Our approach can however also be applied with other feature representations such as the term-frequency/inverse-document-frequency (TF/IDF).

5.2 Fully Tagged Corpus

Our first set of experiments are with a fully tagged corpus. To assess the efficacy of the inclusion of tag information for webpage clustering, we compare the following approaches in our experiments:

(1) **Word feature vector only:** For this, we only consider the words appearing in the webpages. We construct feature vector for each webpage using the bag of words representation, using the words extracted from the page-text.

(2) **Tag feature vector only:** For this, we only consider the tags associated with each webpage, and construct feature vector for each webpage using the bag of tags representation. The tag set for each webpage consists of the tags applied to it by *all* users in the Delicious dataset.

(3) **Word feature vector + Tag feature vector:** For this, we created an augmented feature vector by *concatenating* the tag feature vector with the word feature vector and normalized appropriately (as done in [Ramage et al. 2009]).

(4) **Kernel PCA on words + tags feature vector:** For this, we apply Kernel PCA on the concatenated word + tag feature vector (3) and use extracted features for the final clustering.

(5) **Kernel CCA on words and tags feature vectors:** For this, we treat features derived using (1) and (2) as two *views* of the data, and perform a CCA over both views to learn a shared subspace. Projections of the word feature vector in this subspace are then used as features for the final clustering.

In addition, we also experimented with Kernel PCA *separately* on word features and tag features, and found the performance in both cases to be lower than Kernel PCA on the joint vector. Therefore we skip those results from the presentation, and only report the results of Kernel PCA on the joint words + tags vector.

In our experiments with Kernel PCA and Kernel CCA, we have used linear, polynomial, and Gaussian (RBF) kernels. The hyperparameter for Gaussian kernel (the kernel width parameter) is set to the median pair-wise distance between examples. We note that it is also possible to learn a suitable kernel from the data [Weinberger et al. 2004] but that is not our focus in this paper.

We performed experiments both with full data available, and also with varying amount of data). In particular, the latter experiment was conducted to assess the performance of various approaches when the number of webpages is small but the feature vector associated with each webpage is high dimensional. The number of projection directions for PCA and CCA are kept sufficiently large - as the feature vector size is much larger than the number of webpages, we simply set the number of projection directions equal to the number of webpages so that it is reasonably large.

5.2.1 *Full Data.* In our first experiment, we run all the algorithms on the entire collection of the tagged webpages. Our results on the full data are shown in Table-I. As the results in the table indicate, inclusion of tag information in any form seems to improve the performance as compared to the case when only words from page-text are used. This is evidenced by the better results of words + tags as compared to words only and tags only (which has also been shown in some other recent works [Ramage et al. 2009; Lu et al. 2009]).

	F1-Score	**Precision**	**Recall**
Words Only	0.37(±0.025)	0.29(±0.013)	0.48(±0.021)
Tags Only	0.34(±0.014)	0.26(±0.011)	0.44(±0.023)
Words + Tags	0.40(±0.018)	0.35(±0.015)	0.49(±0.031)
Kernel PCA on Words + Tags (Linear)	0.39(±0.035)	0.32(±0.022)	0.51(±0.031)
Kernel PCA on Words + Tags (Polynomial)	0.44(±0.012)	0.35(±0.017)	0.61(±0.009)
Kernel PCA on Words + Tags (Gaussian)	0.40(±0.014)	0.30(±0.008)	0.53(±0.021)
Kernel CCA on Words and Tags (Linear)	0.42(±0.012)	0.33(±0.011)	0.62(±0.006)
Kernel CCA on Words and Tags (Polynomial)	0.48(±0.006)	0.36(±0.008)	0.79(±0.014)
Kernel CCA on Words and Tags (Gaussian)	0.46(±0.009)	0.34(±0.011)	0.73(±0.013)

Table I. Clustering performances of various methods on the full collection of tagged webpage data. Each experiment has been run 20 times.

Among the Kernel based approaches, Kernel PCA on words + tags performs mostly comparably with raw words + tags (although it did better for the Polynomial Kernel case). Finally, we observe that the Kernel CCA based approach does best overall, suggesting that taking into account the correlations between tags and words indeed leads to an improved performance. Among the kernel based approaches, the polynomial kernel (with degree 2) performed the best in all cases.

5.2.2 *Varying Data Amount.* In our second experiment, we looked at how the various approaches perform when the number of webpages is small. For this experiment, we gradually vary the number of webpages from 100 to 600 and monitor the F-scores reported by all the approaches. The results are shown in Figure 3.

As we can see in Figure 3 (left) that words only, tags only, and words + tags based approaches perform poorly when the number of webpages is small. Also, notice that words + tag performs worse than words only when the number of webpages is very small, possibly due to poor parameter estimation for high dimensional yet small sample size. The words + tags based approach does however begin to outperform the words only and tags only approaches as the number of webpages increases. On the other hand, we observe that both PCA and CCA based approaches consistently perform better than the other 3 baselines, with CCA being the best overall.

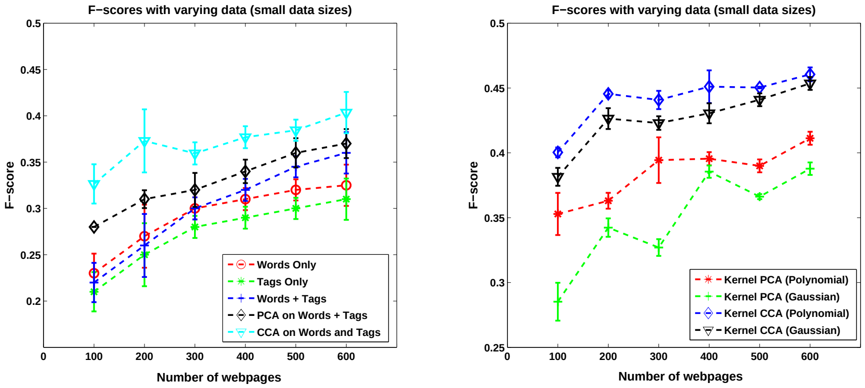


Fig. 3. Performance of the various approaches for the case of very small number of webpages and then varying the amounts of data: Top: Tag augmented PCA and CCA (with linear kernel) compared against other baselines (words only, tags only, words + tags). Bottom: Comparison between the kernel based approaches for non-linear kernels (polynomial kernel has degree 2; higher degrees did not lead to better performance)

Figure 3 (right) compares both kernel based feature extraction approaches - Kernel PCA and Kernel CCA for 2 choices of kernels, polynomial and Gaussian. Compared with the linear feature extraction (Figure 3 top), we see that the kernel based approaches yield better F-scores, with the Kernel CCA being better than Kernel PCA. The better performance of Kernel CCA over Kernel PCA can be attributed to the fact that although Kernel PCA performs a joint projection of words + tags feature vector, it maximizes the *variances* of the word feature vector and the tag feature vector *individually*. On the other hand, the Kernel CCA based approach maximizes their *correlations*, resulting in the better performance.

5.3 Partially Tagged Corpus

To simulate the partially tagged corpus setting, we provide our algorithm the tag features for only a small fraction of webpages in the corpus. For the remaining webpages, we only use the page-text based features. We call webpages with both page-text and tag information available as *paired*, and webpages with only page-text information available as *non-paired*. In our experiment, we vary the fraction of paired webpages from 10% to 60%.

We compare our kernel-completion-followed-by-KCCA based approach with two baselines. Our first baseline is KCCA with full view information, i.e., all the webpages are paired with their corresponding tag information. Our second baseline is an incomplete view setting like ours: KCCA on paired webpages but Kernel PCA on non-paired webpages (since only a single view, page-text, is available for non-paired webpages). The k-means algorithm was used as the base clustering algorithm in our approach as well as the other baselines. However, any other clustering algorithm can be used as well. Since k-means might be sensitive to initializations, we run it 20 times and report the mean and standard deviations. Gaussian kernel was used for all the kernel computations and the width parameter was set to the median pairwise distance between the examples. For the evaluation of clustering performance: we used the *average cluster entropy* which is based on the impurity of a cluster given the true classes in the data. If p_{ij} be the fraction of class j in obtained cluster i, N_i be the size of cluster i, and N be the total number of examples, then the average cluster entropy is defined as: $E = \sum_{i=1}^{K} \frac{N_i(-\sum_j p_{ij}\log(p_{ij}))}{N}$, where K is the number of clusters.

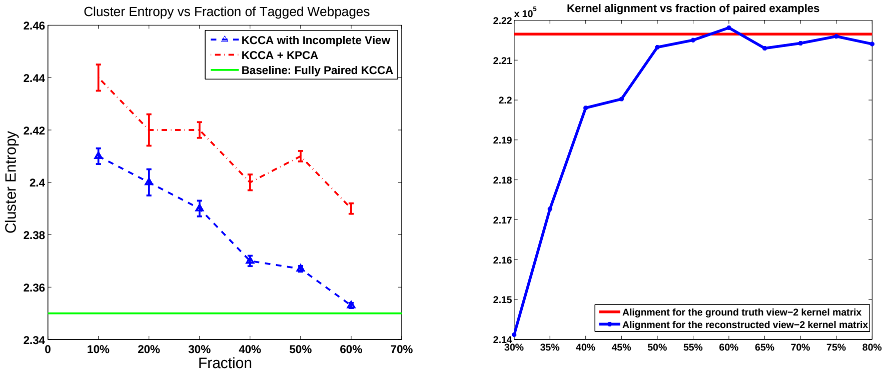


Fig. 4. Performance of the various approaches

The performance of our approach and the other baselines is shown in Figure 4(left). As we can see, our approach with incomplete view with just about 50% to 60% paired webpages achieves comparable results to the fully paired KCCA case. With fractions higher than that, we observed the performance to wiggle around and stay roughly the same (moderately better or moderately worse) to the fully paired case. On the other hand, it significantly outperforms the other baseline that uses KCCA on the paired webpages and kernel PCA on the non-paired webpages. The inferior performance of the KCCA+KPCA approach can be attributed to the fact that only a small subset of the webpages use the tag information for the low-dimensional projection.

We also experimented to see how good is the reconstructed tag view kernel matrix with respected to the ground truth kernel matrix of tag features. To do this, we vary the fraction of paired examples as did in the previous experiment and plot the alignment of matrices in both views. As we can see from Figure 4 (right), the alignment gets better as the fraction of paired examples increases, and with about 55-60% paired examples, the alignment is almost as high as the alignment obtained on the ground truth kernel matrix.

Note that merely having a high alignment between $\mathbf{K}_x$ and $\mathbf{K}_y$ does not ensure that the multiview clustering performance will be good. If the number of examples in view $\mathcal{Y}$ is very small, then the optimization could give a kernel matrix $\mathbf{K}_y$ that may be very similar to $\mathbf{K}_x$ and it may not give any useful information from view $\mathcal{Y}$. Therefore one needs a sufficient number of examples from view $\mathcal{Y}$ so that the obtained kernel matrix $\mathbf{K}_y$ actually is a good representative of the similarities between examples in view $\mathcal{Y}$. As shown in our experiments, about 50% to 60% tagged webpages gave close to optimal performance. Another thing to note here is that the reconstructed kernel matrix $\mathbf{K}_y$ in our approach depends on the kernel $\mathbf{K}_y^{cc}$ constructed using tagged set of webpages so the reconstruction accuracy (and hence the clustering performance) depends on how good is $\mathbf{K}_y^{cc}$. With a small but reasonably well tagged subset of the whole data, we expect the reconstructed $\mathbf{K}_y$ to be sufficiently close to the optimal kernel matrix in view $\mathcal{Y}$.

5.3.1 *A Tag-Prediction based Baseline for Partially Tagged Corpus.* Another way to deal with the case when the tags are available for only a small number of webpages is to use the tagged webpages for predicting the tags for the rest of them (akin to the framework proposed by [Hardoon et al. 2006] which automatically annotates images using annotations for similar images). Under this approach, one can perform a latent semantic analysis or CCA to discover a semantic subspace of webpages having tag information available. After that, each non-tagged webpage can be projected onto this subspace and can be assigned the same tags as that of the tagged webpage closest to it *in the semantic subspace*. We note here that although the similarities among documents can be compared in the original feature space, a closeness measure in the semantic subspace is a better measure of similarity between two documents, because we would be measuring *thematic similarities* in this subspace. Once we do this for all non-tagged webpages, we will have full information (i.e., tags with page-text for all webpages) to apply the CCA based approach we proposed in this paper. In our experiments, we found that this baseline did roughly similar to the KCCA+KPCA baseline.

6. RELATED WORK

A number of techniques have been proposed in the past to improve information retrieval tasks using auxiliary sources of information, e.g., anchor text for web search [Eiron and McCurley 2003], interconnectivity of webpages [Cohn and Hofmann 2001], captions for image retrieval [Blei and Jordan 2003], etc. Other recent works on exploiting social annotations, in particular, to improve various web mining tasks include annotation based approaches to web search [Bao et al. 2007], webpage classification [Zubiaga et al. 2009], and information retrieval in general [Zhou et al. 2008]. Similar in spirit to our work, using tag information for webpage clustering has earlier been proposed in [Ramage et al. 2009; Lu et al. 2009] using a concatenation of word and tag feature vectors. In [Ramage et al. 2009], the authors also proposed a probabilistic generative model based on an extension of the Latent Dirichlet Allocation [Blei et al. 2003]. Their model is essentially the same as the conditionally independent LDA (CI-LDA) which assumes separate sets of topics for words and tags. This assumption tends to loosen the coupling/correlations between the word topics and the tag topics [Newman et al. 2006]. Another issue is that exact inference in such models is intractable and therefore approximations are needed which require using Markov Chain Monte Carlo, or variational methods. In contrast, our CCA based approach reduces to solving an eigenvalue problem which can be solved efficiently using existing eigensolvers. Another benefit of using the kernel variant of CCA we use in this paper is that the complexity of solving the eigenvalue problem depends on the number of webpages rather than the vocabulary size which would be especially advantageous when the number of webpages is small as compared to the vocabulary size.

Among other works that use CCA, Chaudhury et al [Chaudhuri et al. 2009] used the CCA based approach for audio-visual speaker clustering and hierarchical Wikipedia document clustering by category, and showed that CCA based approach outperforms PCA based clustering approaches. In another work, Blaschko et al [Blaschko and Lampert 2008] use CCA for clustering images using the associated text as a second view. Both of these works assume that the views are complete, unlike the setting we considered in this paper.

7. FUTURE WORK

There are a number of possible extensions of our work. One direction in the partially tagged corpus setting would be to identify which of webpages one should get tags for so as to have the best performance on the learning task at hand (e.g., clustering with partially tagged corpus). Active Learning [Settles 2009] could be useful in such a setting. Here we briefly describe an active learning based approach to accomplish this.

7.1 An Active Learning Extension

In the partially tagged case, we have a set of tagged webpages and rest of the webpages are untagged. If there is a budget on the set of tagged webpages, one should get those webpages tagged which are the most informative about the rest of the corpus (especially about the webpages that are not tagged). The partially tagged corpus setting is like a transduction/semi-supervised learning setting where we want to learn with both labeled and unlabeled data. Let us denote by T, the binary matrix indicating with $T_{ij} = 1$ if a tagged webpage i has been assigned the tag j. Given T, just as in Section 4 where we compute $\mathbf{A}_m$, we can use the following equation to predict the tags for the untagged webpages (assuming that the tags for the untagged webpages come from the same tag vocabulary):

$$U = -(\mathcal{L}_x^{mm})^{-1}(\mathcal{L}_x^{cm})^T T \tag{6}$$

where $\mathcal{L}_x$ is the graph Laplacian of the data using the page-text view (Section 4). Using the tag predictions U on the untagged webpages, we can compute the estimated *risk* (expected tag prediction error on the untagged webpages) on the untagged webpages as is done in [Zhu et al. 2003]. In our active learning extension, as we choose a new webpage to tag, a new row is added to the matrix T, and we get a new estimate U for the tag predictions of the untagged webpages. As suggested in [Zhu et al. 2003], the chosen webpage should be such that it minimizes the estimated risk on the untagged set of webpages. This can be our criteria at each step to select webpages which we should get tags for. We leave the further details for future work.

7.2 Other Ways to Deal with Partially Tagged Corpus

In this paper, we used a kernel matrix completion approach. There exist a number of other possibilities which are worth investigating in our setting. We describe some of these here briefly. The first approach (Section 7.2.1) uses a semi-supervised version of CCA which can extract features using both tagged and non-tagged webpages, or can use a combination of CCA and LSA on the tagged and non-tagged webpages respectively. The second approach (Section 7.2.2) is based on first predicting the tags for non-tagged webpages using any of the several methods described, and then applying the Kernel CCA based clustering approach we have proposed in this paper.

7.2.1 *Semi-supervised Projections.* It is possible to apply the CCA based approach in a semi-supervised fashion using both tagged and non-tagged webpages. For example, one can take a probabilistic approach to CCA [Rai and Daumé III 2009] and treat the missing tags for non-tagged webpages as latent variables. In the non-probabilistic setting, one can use the semi-supervised variants of CCA [Blaschko et al. 2008; Kim and Pavlovic 2009] which do not require full information from both the views. Alternatively, a somewhat similar way of accomplishing this would be to write a combined eigenvalue problem with one part of it being CCA on the tagged webpages, and the other being LSA on the non-tagged webpages.

7.2.2 *Predicting Tags for Non-Tagged Webpages.* A number of approaches have also been proposed in the recent past that autopredict tags [Brooks and Montanez 2006] and such approaches can be also used for predicting tags for non-tagged webpages. Another rather naïve option could be to use the tagged corpus of webpages to train several prediction models, one for predicting each tag, and then use these models to predict the tags for non-tagged webpages. A problem with such an approach is the large number of tags which leads to scalability issues. Furthermore, tags can potentially come from an open-vocabulary and be sparse [Law et al. 2010]. Another issue could be synonymy where two different tags may have the same meaning. To address these issues in the context of music clip tag prediction, [Law et al. 2010] proposed a framework that organizes tags into semantically meaningful classes using topic models, and then predicts these classes given a non-tagged piece of music. Such an approach can be useful for webpage tag prediction as well.

7.3 A Note on Tag Relevance

Finally, not all tags are meaningful for a given webpage. Some spurious tags can hamper the discriminative power of the more relevant ones. One can filter such spurious tags before using them [Suchanek et al. 2008]. This roughly amounts to doing feature selection but here the feature selection for tags can benefit from the other sources of information (such as how many users applied a particular tag to some document). Incorporating such information can lead to identifying the tags that are most discriminative, and hence is expected to lead to even better performance.

8. DISCUSSION AND CONCLUSION

User generated content can be a very rich source of useful information for web-mining and information retrieval on the web. Intelligent ways of harnessing this rich source of information can greatly benefit the existing web-mining algorithms. Often the usefulness of user-generated content is due to the fact that it is small but structured (e.g., tags), in addition to being semantically precise, which can nicely complement the huge but unstructured information (e.g., page-text). As we have seen in this paper, tag information can be exploited in numerous ways to improve webpage clustering, both when tags for available for all webpages as well as in the case when the tag information is available only for a small subset of webpages. Although we have presented results for webpage clustering, due to the discriminative information provided by the tags, the features extracted by our CCA based approach can also be useful for webpage classification. In this paper we have considered the case when tags are the auxiliary source of information; the proposed approaches can also be useful for harnessing the benefits of other type of meta-data generated by users on the web.

Finally, future work will also investigate how considering meta-data such as tags associated with document can help in domains other than the Web. For example, in Medical Informatics, clustering patient records can be a difficult problem since these records often tend to be highly unstructured and noisy. However, often these records are marked with very specific tags which can be exploited in a manner similar to what we have presented in this paper. Also, the multiview clustering with incomplete views has natural applications in clustering with multilingual data (for cross-lingual information retrieval). Since machine translation is a hard problem and can also be error prone, one could consider obtaining good translations for a small fraction of the documents in the corpus and then apply our multiview clustering algorithm with incomplete views.

Acknowledgement

This work is supported by resources and facilities of the VA Salt Lake City Health Care System with funding support from the Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374 and the VA Informatics and Computing Infrastructure (VINCI), VA HSR HIR 08-204. This work was also partially supported by NSF grant IIS-0712764.

REFERENCES

Ando, R. K. and Zhang, T. 2007. Two-view feature generation model for semi-supervised learning. In *ICML '07*. 25–32.

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

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., and Su, Z. 2007. Optimizing web search using social annotations. In *WWW '07*. 501–510.

Bickel, S. and Scheffer, T. 2004. Multi-view clustering. In *ICDM '04*. IEEE Computer Society, Washington, DC, USA, 19–26.

Blaschko, M. B. and Lampert, C. H. 2008. Correlational spectral clustering. In *CVPR*.

Blaschko, M. B., Lampert, C. H., and Gretton, A. 2008. Semi-supervised laplacian regularization of kernel canonical correlation analysis. In *ECML PKDD '08*. Springer-Verlag, Berlin, Heidelberg.

Blei, D. M. and Jordan, M. I. 2003. Modeling annotated data. In *SIGIR '03*. 127–134.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

Blum, A. and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT' 98*. 92–100.

Boyd, S. and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Brefeld, U. and Scheffer, T. 2004. Co-em support vector learning. In *ICML '04*. 16.

Brooks, C. H. and Montanez, N. 2006. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06*. 625–632.

BURGES, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2,* 2, 121–167.

CARREIRA-PERPINAN, M. A. AND LU, Z. 2007. The Laplacian Eigenmaps Latent Variable Model. In *AISTATS*.

CHAUDHURI, K., KAKADE, S. M., LIVESCU, K., AND SRIDHARAN, K. 2009. Multi-view clustering via canonical correlation analysis. In *ICML '09*. 129–136.

COHN, D. AND HOFMANN, T. 2001. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS*.

DASGUPTA, S. 1999. Learning mixtures of gaussians. In *FOCS '99*. IEEE Computer Society, Washington, DC, USA.

DE SA, V. R. 2005. Spectral Clustering with two views. In *Proceedings of the Workshop on Learning with Multiple Views, ICML*.

EIRON, N. AND MCCURLEY, K. S. 2003. Analysis of anchor text for web search. In *SIGIR '03*. 459–460.

FOSTER, D. P., KAKADE, S. M., AND ZHANG, T. 2008. Multi-view dimensionality reduction via canonical correlation analysis. *Technical Report TTI-TR-2008-4*.

GESTEL, T. V., SUYKENS, J. A. K., BRABANTER, J. D., MOOR, B. D., AND VANDEWALLE, J. 2001. Kernel canonical correlation analysis and least squares support vector machines. In *ICANN '01*. Springer-Verlag, London, UK, 384–389.

HARDOON, D. R., SAUNDERS, C., SZEDMAK, O., AND SHAWE-TAYLOR, J. 2006. A correlation approach for automatic image annotation. In *Springer LNAI 4093*. 681–692.

HARDOON, D. R. AND SHAWE-TAYLOR, J. 2003. Kcca for different level precision in content-based image retrieval. In *Third International Workshop on Content-Based Multimedia Indexing, IRISA*.

HARDOON, D. R., SZEDMAK, S. R., AND SHAWE-TAYLOR, J. R. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation 16,* 12, 2639–2664.

HOTELLING, H. 1936. Relations Between Two Sets of Variables. *Biometrika*, 321–377.

JI, S., TANG, L., YU, S., AND YE, J. 2008. Extracting shared subspace for multi-label classification. In *KDD '08*. 381–389.

KAKADE, S. M. AND FOSTER, D. P. 2007. Multi-view regression via canonical correlation analysis. In *COLT'07*. 82–96.

KIM, M. AND PAVLOVIC, V. 2009. Covariance operator based dimensionality reduction with extension to semi-supervised settings. In *AIStats*.

KRIEGEL, H.-P., KRÖGER, P., AND ZIMEK, A. 2009. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data 3,* 1, 1–58.

LAW, E., SETTLES, B., AND MITCHELL, T. 2010. Learning to tag from open vocabulary labels. In *ECML PKDD '10*. Springer, Berlin, Heidelberg.

LU, C., CHEN, X., AND PARK, E. K. 2009. Exploit the tripartite network of social tagging for web clustering. In *CIKM '09*. 1545–1548.

MCQUEEN, J. 1967. Some Methods of Classification and Analysis of Multivariate Observations. In *Proc. Fifth Berkeley Symp. Math. Statistics and Probability*.

MUSLEA, I., MINTON, S., AND KNOBLOCK, C. A. 2002. Active + semi-supervised learning = robust multi-view learning. In *ICML '02*. San Francisco, CA, USA, 435–442.

NEWMAN, D., CHEMUDUGUNTA, C., AND SMYTH, P. 2006. Statistical entity-topic models. In *KDD '06*. ACM, New York, NY, USA, 680–686.

RAI, P. AND DAUMÉ III, H. 2009. Multi-label prediction via sparse infinite CCA. In *NIPS*. Vancouver, Canada.

RAMAGE, D., HEYMANN, P., MANNING, C. D., AND GARCIA-MOLINA, H. 2009. Clustering the tagged web. In *WSDM '09*. 54–63.

RUSTANDI, I., JUST, M. A., AND MITCHELL, T. M. 2009. Integrating multiple-study multiple-subject fmri datasets using canonical correlation analysis. In *MICCAI: fMRI data analysis workshop*.

Schölkopf, B., Smola, A., and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10,* 5, 1299–1319.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shawe-Taylor, J. and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis.* Cambridge University Press, New York, NY, USA.

Socher, R. and Fei-Fei, L. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*. San Francisco, CA.

Suchanek, F. M., Vojnovic, M., and Gunawardena, D. 2008. Social tags: meaning and suggestions. In *CIKM '08*. 223–232.

Vempala, S. and Wang, G. 2002. A spectral algorithm for learning mixtures of distributions. In *FOCS '02*. IEEE Computer Society, Washington, DC, USA.

von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing 17,* 4, 395 – 416.

Weinberger, K. Q., Sha, F., and Saul, L. K. 2004. Learning a kernel matrix for nonlinear dimensionality reduction. In *ICML '04*.

yuan Zhao, X., Sun, D., and chuan Toh, K. 2010. A newton-cg augmented lagrangian method for semidefinite programming. *SIAM Journal of Optimization*.

Zhou, D., Bian, J., Zheng, S., Zha, H., and Giles, C. L. 2008. Exploring social annotations for information retrieval. In *WWW '08*. 715–724.

Zhu, X., Lafferty, J., and Ghahramani, Z. 2003. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining.* 58–65.

Zubiaga, A., Martínez, R., and Fresno, V. 2009. Getting the most out of social annotations for web page classification. In *DocEng '09: Proceedings of the 9th ACM symposium on Document engineering*. 74–83.