Exploiting Tag and Word Correlations for Improved Webpage Clustering

Anusua Trivedi
School of Computing
University of Utah
Salt Lake City, Utah, USA
anusua@cs.utah.edu

Piyush Rai
School of Computing
University of Utah
Salt Lake City, Utah, USA
piyush@cs.utah.edu

Scott L. DuVall
VA SLC Healthcare System &
University of Utah
Salt Lake City, Utah, USA
scott.duvall@utah.edu

Hal Daumé III*
Dept. of Computer Science
Universty of Maryland
College Park, Maryland, USA
me@hal3.name

ABSTRACT

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[1] and Delicious[2], has led to a huge amount of user-generated content such as the tag information that is associated with the webpages. In this paper, we present a subspace based feature extraction approach which leverages tag information to complement the page-contents of a webpage to extract highly discriminative 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 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. Although our results here are on the webpage clustering task, the same approach can be used for webpage classification as well. In the end, we also suggest possible future work for leveraging tag information in webpage clustering, especially when tag information is present for not all, but only for a small number of webpages.

*Also holds an adjunct position with the School of Computing, University of Utah

[1]www.stumbleupon.com

[2]www.delicious.com

SMUC'10, October 30, 2010, Toronto, Ontario, Canada.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *clustering*; H.1.2 [**Models and Principles**]: User/Machine Systems - *human information processing*

General Terms

Algorithms

Keywords

Social Tagging, Webpage Clustering

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, Term-Frequency/Inverse-Document-Frequency, etc., and then applies standard clustering algorithms (e.g., K-means algorithm [30], spectral clustering [41], etc.). Another approach somewhat related to clustering is to mine topic information from documents collections (e.g., Latent Dirichlet Allocation [9]), 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 reg-

ularly 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 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 be very discriminative 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 [10, 11, 31, 5, 2, 25] in various machine learning tasks, we use two views of the data (page-text and 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 [23, 19, 22] (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 [38], or features and labels (supervised dimensionality reduction [33, 24]). Therefore such an approach is expected to be useful for extracting useful features in the case of webpage clustering as well since the data often does have multiple views (page-text and tags in our case).

Although in this paper, we consider webpage clustering, the tag directed feature extraction approach we propose here can also be useful for tasks other than clustering. For instance, if the task is webpage classification instead, the extracted features are expected to help the classification task as well.

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 algorithms respectively. Our results are described in Section 4. We discuss related work in Section 5. In Section 6, we briefly describe future work, geared towards settings when tag information is available for not all but only for a small number of webpages, and conclude with Section 7.

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 [1] (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 [34, 29].

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.

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 [34, 29]:

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 [34]).

It turns out [34, 29] 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 [27]. 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 [34].

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 [40] (PCA), or

random projections [16]. 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 [11, 31, 5, 2, 25, 18]. Multi-view approaches help supervised learning algorithms by being able to leverage unlabeled data [10], whereas, for unsupervised learning algorithms, multiple views of the data can often help in extracting better features [18].

Canonical Correlation Analysis [23] (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 [21, 22, 35].

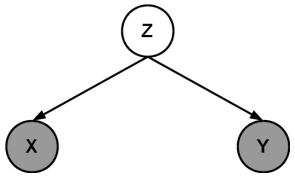

Figure 1: The dependency view of CCA: Coupled datasets X and Y, and their shared subspace defined by Z. In our webpage clustering setting, X corresponds to the features derived from the page-text and Y corresponds to the features derived from the tags. Z represents the semantic subspace shared by both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

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

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

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.

$$\max_{\mathbf{w}_x, \mathbf{w}_y} \mathbf{w}_x^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_y \quad (2)$$

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 [22] 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 [37] 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 [13] for classification, Kernel Principal Component Analysis [36] 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* [37].

The kernel variant of CCA (called Kernel Canonical Correlation Analysis - KCCA) can be thought of as first (implicitly) mapping each D dimensional data point $\boldsymbol{x}$ 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 [22] 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:

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

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:

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

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 [37, 22] 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 [3] (ICD). We, on the other hand, use Partial Gram-Schmidt Orthogonalization (PGSO) as suggested in [22]. 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. 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.

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 [34]).

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 chosen via cross-validation. We note that it is also possible to learn a suitable kernel from the data [42] but that is not our focus in this paper.

4.1 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).

4.2 Clustering Evaluation Metric

For our evaluation, we compare the obtained clusterings by all methods with the ground truths provided by the Open Directory Project [1] (ODP). The ODP is a human-maintained and edited hierarchical web directory which consists of 17 top-level directories, out of which we used 14 top-level categories. This is also the number of clusters K given as input to the clustering algorithms. Each node in the ODP directory has a label such as Arts, Games, Computers, Business, etc. To obtain the gold-clustering, we assign the same cluster label to a node (webpage) and all other pages that are its descendants in the directory tree.

The evaluation metric we use is the F1-score which is defined as the harmonic mean of Precision defined as

$$p = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

and Recall defined as

$$r = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Precision and recall scores indicate how the cluster assignments between the obtained clustering and the ground truth clustering agree or disagree, over pairs of documents.

4.3 Results

We performed a number of experiments both with full data available, and also with varying amount of data (especially when the number of webpages is much smaller than the feature dimensionality). 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.

4.3.1 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-1. 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 [34, 29]).

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.

4.3.2 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 2.

As we can see in Figure 2 (top) 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.

Figure 2 (bottom) 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 2 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. 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 [17], interconnectivity of webpages [15], captions for image retrieval [8], etc. Other recent works on exploiting social annotations, in particular, to improve various web mining tasks include annotation based approaches to web search [4], webpage classification [45], and information retrieval in general [44]. Similar in spirit to our work, using tag information for webpage clustering has earlier been proposed in [34, 29] using a concatenation of word and tag feature vectors. In [34], the authors also proposed a probabilistic generative model based on an extension of the Latent Dirichlet Allocation [9]. 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 [32]. Another issues 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 [14]

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

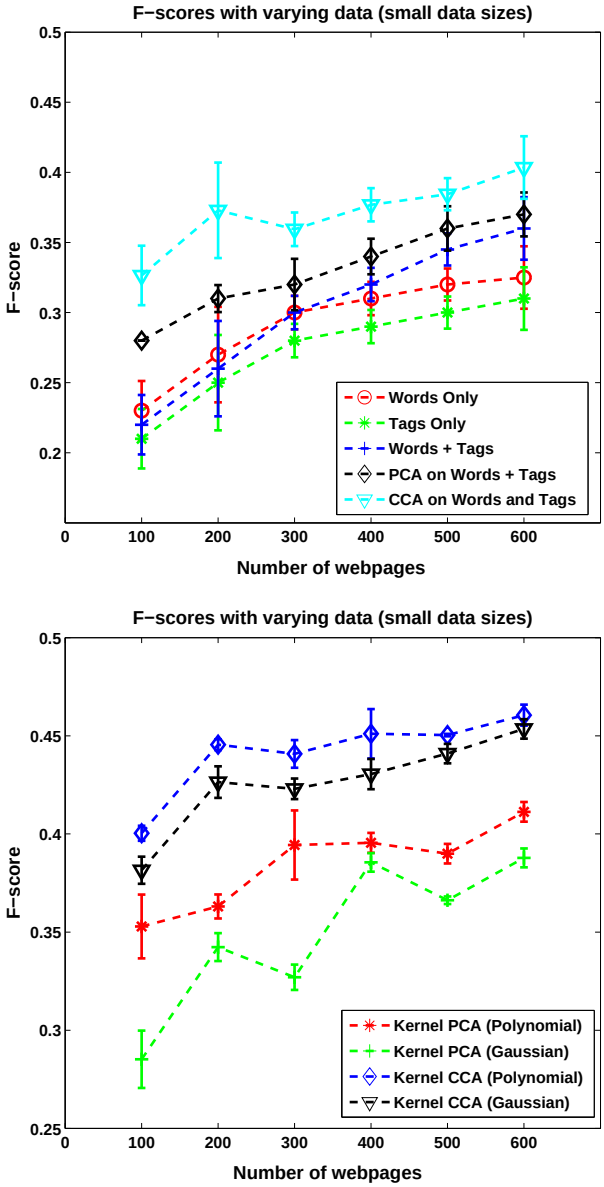

Figure 2: Performance of the various approaches for the case of very small number of webpages and then varying the amounts of data: Top: Tag augmented PCA and CCA (with linear kernel) compared against other baselines (words only, tags only, words + tags). Bottom: Comparison between the kernel based approaches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

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 [6] use CCA for clustering images using the associated text as a second view.

6. FUTURE WORK

To our knowledge, all the existing approaches exploiting tag information for webpage clustering (e.g., [34, 29] and the CCA based approach we proposed in this paper) assume that all the webpages are tagged, which is a somewhat restrictive assumption. In a more realistic setting, one can only expect that the tags will be available for only a small number of webpages. In this section, we suggest some alternatives which can make it possible to exploit tag information even when the tag information in available for only a small number of webpages.

The first approach (Section 6.1) is based on an annotation based probabilistic latent semantic analysis (LSA) [43] over document-word and tag-word co-occurrence matrices. The second approach (Section 6.2) 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 third approach (Section 6.3) 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.

We provide brief descriptions of all these approaches here and leave the detailed evaluations for future work.

6.1 Annotation based Probabilistic LSA

Assume that we are given two sets of webpages - one set $\mathcal{T}$ is tagged and the other set $\mathcal{U}$ is non-tagged. Further, $|\mathcal{T}| \ll |\mathcal{U}|$, and $N = |\mathcal{T}| + |\mathcal{U}|$ is the total number of webpages. The goal is to obtain a clustering of all N webpages. We define the following:

- A = document-word co-occurrence matrix (bag-of-words representation) of size $N \times |W|$ where N is the number of documents (webpages) in the corpus, and $|W|$ is the page-text vocabulary size. A_{ij} denotes the frequency of the word j appearing in document i. Note that the document-word co-occurence matrix is constructed using both tagged and non-tagged webpages.
- B = tag-word co-occurrence matrix (bag-of-words representation) of size $|T| \times |W|$ where $|T|$ is the total number of tags in the corpus, and $|W|$ is the page-text vocabulary size. B_{ij} denotes the number of times tag i is associated with word j. Note that the tag-word matrix is constructed using only the tagged webpages

Note that this is a more fine-granular association where we do not look for the associations between the tag and a webpage, but go a level further to consider the co-occurences of tags with the actual words appearing in the webpages (based on the tag-word co-occurrence matrix). Also, we would assume the same page-text vocabulary while constructing matrices A and B. To do this, we pool all N webpages (with and without tag information) and construct a common vocabulary of size $|W|$. The vocabulary would not include tags (unless some tags, coincidentally, are words in some webpages).

Having constructed the document-word and word-tag co-occurrence matrices A and B, *joint* PLSA can be applied using A and B in a manner similar to [15]. A similar framework was applied in [43] for the problem of clustering images on the social web using the image captions.

6.2 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 [33] 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 [7, 26] 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.

6.3 Predicting Tags for Non-Tagged Webpages

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 [20] 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.

A number of other approaches have also been proposed in the recent past that autopredict tags [12] 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 [28]. 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, [28] 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.

6.4 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 be-

fore using them [39]. 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.

7. 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.

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. PR was partially supported by NSF grant IIS-0712764.

8. REFERENCES

[1] *Open Directory Project (http://www.dmoz.org/).*

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

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

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

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

[6] Blaschko, M. B., and Lampert, C. H. Correlational spectral clustering. In *CVPR* (2008).

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

[8] Blei, D. M., and Jordan, M. I. Modeling annotated data. In *SIGIR '03* (2003), pp. 127–134.

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

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

[11] Brefeld, U., and Scheffer, T. Co-em support vector learning. In *ICML '04* (2004), p. 16.

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

[13] Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, 2 (1998), 121–167.

[14] Chaudhuri, K., Kakade, S. M., Livescu, K., and Sridharan, K. Multi-view clustering via canonical correlation analysis. In *ICML '09* (2009), pp. 129–136.

[15] Cohn, D., and Hofmann, T. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS* (2001).

[16] Dasgupta, S. Learning mixtures of gaussians. In *FOCS '99* (Washington, DC, USA, 1999), IEEE Computer Society.

[17] Eiron, N., and McCurley, K. S. Analysis of anchor text for web search. In *SIGIR '03* (2003), pp. 459–460.

[18] Foster, D. P., Kakade, S. M., and Zhang, T. Multi-view dimensionality reduction via canonical correlation analysis. *Technical Report TTI-TR-2008-4* (2008).

[19] Gestel, T. V., Suykens, J. A. K., Brabanter, J. D., Moor, B. D., and Vandewalle, J. Kernel canonical correlation analysis and least squares support vector machines. In *ICANN '01* (London, UK, 2001), Springer-Verlag, pp. 384–389.

[20] Hardoon, D. R., Saunders, C., Szedmak, O., and Shawe-taylor, J. A correlation approach for automatic image annotation. In *Springer LNAI 4093* (2006), pp. 681–692.

[21] Hardoon, D. R., and Shawe-taylor, J. Kcca for different level precision in content-based image retrieval. In *Third International Workshop on Content-Based Multimedia Indexing, IRISA* (2003).

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

[23] Hotelling, H. Relations Between Two Sets of Variables. *Biometrika* (1936), 321–377.

[24] Ji, S., Tang, L., Yu, S., and Ye, J. Extracting shared subspace for multi-label classification. In *KDD '08* (2008), pp. 381–389.

[25] Kakade, S. M., and Foster, D. P. Multi-view regression via canonical correlation analysis. In *COLT'07* (2007), pp. 82–96.

[26] Kim, M., and Pavlovic, V. Covariance operator based dimensionality reduction with extension to semi-supervised settings. In *AIStats* (2009).

[27] Kriegel, H.-P., Kröger, P., and Zimek, A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data 3*, 1 (2009), 1–58.

[28] Law, E., Settles, B., and Mitchell, T. Learning to tag from open vocabulary labels. In *ECML PKDD '10* (Berlin, Heidelberg, 2010), Springer.

[29] Lu, C., Chen, X., and Park, E. K. Exploit the tripartite network of social tagging for web clustering. In *CIKM '09* (2009), pp. 1545–1548.

[30] Manning, C. D., Raghavan, P., and Schtze, H. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[31] Muslea, I., Minton, S., and Knoblock, C. A. Active + semi-supervised learning = robust multi-view learning. In *ICML '02* (San Francisco, CA, USA, 2002), pp. 435–442.

[32] Newman, D., Chemudugunta, C., and Smyth, P. Statistical entity-topic models. In *KDD '06* (New York, NY, USA, 2006), ACM, pp. 680–686.

[33] Rai, P., and Daumé III, H. Multi-label prediction via sparse infinite CCA. In *NIPS* (Vancouver, Canada, 2009).

[34] Ramage, D., Heymann, P., Manning, C. D., and Garcia-Molina, H. Clustering the tagged web. In *WSDM '09* (2009), pp. 54–63.

[35] Rustandi, I., Just, M. A., and Mitchell, T. M. Integrating multiple-study multiple-subject fmri datasets using canonical correlation analysis. In *MICCAI: fMRI data analysis workshop* (2009).

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

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

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

[39] Suchanek, F. M., Vojnovic, M., and Gunawardena, D. Social tags: meaning and suggestions. In *CIKM '08* (2008), pp. 223–232.

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

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

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

[43] Yang, Q., Chen, Y., Xue, G.-R., Dai, W., and Yu, Y. Heterogeneous transfer learning for image clustering via the social web. In *ACL-IJCNLP '09* (2009), pp. 1–9.

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

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