A Flexible Framework for Projecting Heterogeneous Data

Aubrey Gress
Department of Computer Science
University of California, Davis
agress@ucdavis.edu

Ian Davidson
Department of Computer Science
University of California, Davis
davidson@cs.ucdavis.edu

ABSTRACT

In many real world settings the data to analyze is heterogeneous consisting of (say) images, text and video. An elegant approach when dealing with such data is to project all the data to a common space so standard learning methods can be used. However, typical projection methods make strong assumptions such as the multi-view assumption (datum in one data set are always associated with a single datum in the other view) or that the multiple data sets have an overlapping feature space. Such strong assumptions limit what data such work can be applied to. We present a framework for projecting heterogeneous data from multiple data sets into a common lower dimensional space using a rich range of guidance which does not assume any overlap between the instances or features in different data sets. Our work can specify inter-dataset (between instances in different data sets) guidance and intra-dataset (between instances in the same data set) guidance, both of which can be positively or negatively weighted. We show our work offers substantially more flexibility over related methods such as Canonical Correlation Analysis (CCA) and Locality Preserving Projections (LPP) and illustrate its superior performance for supervised and unsupervised learning problems.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

Dimensionality Reduction; Spectral Methods; Heterogeneous Data

CIKM'14, November 3–7, 2014, Shanghai, China.

ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2662030.

1. INTRODUCTION

A growing trend in data mining and machine learning is that the data to analyze does not consist of a single instance type or come from a single source, but instead contains fundamentally different instance types. This gives rise to multiple data sets which may not have overlapping features or common instances. Examples include personal information management (images, video, text and geographic tags) and health care records (medical scans, written-notes and medical records). For some special situations the multi-view setting, where each instance has multiple descriptions (views), is natural and many algorithms exist to exploit this extra information [10, 18, 27]. This effectively requires the different data sets to have overlapping instances. Similarly, other methods require the two data sets to have overlapping features [16].

However, a broader setting is required in many other situations. For example, consider a collection of personal data consisting of images, videos and emails. While there are multiple types of data, instances in the different data sets do not have common features nor do they represent different views of the same instance. As such, a more pragmatic model is required to learn a useful embedding of the data. We propose to use relationships between any pair of instances whether they be in the same or different data sets. We focus on the relations of `similarity` and `dissimilarity` between instances. This form of guidance has been used extensively in areas such as metric learning and constrained clustering [6, 22, 24]. It has been shown to be useful at making use of many forms of guidance including side information (i.e. from labels), personal preferences and even geometry [1]. We use this guidance to learn an embedding in which pairs of instances with the `similar` relationship are close to each other and pairs of instances with the `dissimilar` relationship are far from each other. Our work makes use of this type of relational information as well as weighted variations to project all data sets to a common lower dimensional space, allowing analysis using standard methods. For example, after projecting all instances to a common space we may cluster them for organization reasons (see Figure 9). Similarly we may use a nearest-neighbor retrieval algorithm to related objects (see Figure 8).

Embedding two or more data sets with no overlapping feature or instances is a very challenging problem since it requires embedding data to a lower dimensional space when the data are fundamentally different and no obvious distance function exists between them. Existing methods to achieve this aim fall into a number of different categories

with most being methods that make the multi-view assumption [10,18,27]. We can characterize each by the type of guidance/relationships they require the user to provide, allowing us to see how our method is different. Figure 1 explains the differences between common methods and our method and below we briefly summarize their limitations so as to better explain the contributions of our work.

Though this existing work has made significant progress in the area of heterogenous data embedding it is limited in a number of ways that prevents its use for embedding fundamentally different data types. The limitations with existing work are:

1. They typically restrict the type of data they accept. Many techniques make the restrictive multi-view assumption [7, 10], which assumes datum in each view have a **single** relationship with datum in the other view. Other methods [16] require data sets to have overlapping features.
2. The type of guidance is typically limited to unweighted positive guidance, meaning degree of belief and negatively weighted guidance cannot be encoded.
3. They do not necessarily embed all instances into the same space [20, 26, 30].

Our contribution to the field is we present a framework for embedding heterogeneous sets of data with no limitations on whether the data sets or feature space overlap. This allows a variety of pairwise relationships such that:

1. The embedding is generated using pairwise intra-dataset and inter-dataset relationships.
2. The relational guidance can be both positively (similar) and negatively (dissimilar) weighted which allows associating a degree of belief to the relationship.
3. Every instance is embedded into a common space (see Figure 5 for an embedding of three data sets).
4. Our experimental results show such guidance is useful for supervised learning (i.e. Figure 8) and unsupervised learning (see Figure 9).

The outline of the paper is as follows. In section 2 we describe our method. In section 3 we present experimental results of our method applied to K-Nearest Neighbors classification and K-Means Clustering. Finally, in section 4 we discuss related work and then conclude.

2. OUR METHOD

For clarity we shall initially describe our work for just two data sets X and Y (which may have different feature sets) and in the following subsection show how it can be extended to additional data sets.

2.1 Encoding Guidance

Our guidance comes in two forms: intra-dataset and inter-dataset relationships (as shown in Figure 1) between pairs of instances. For both forms of guidance we allow a weight to be associated in $[-1, +1]$. This weight can be interpreted as a degree-of-belief or confidence in the relationship. A large positive/negative weight indicates strong belief that two instances are very similar/dissimilar. A weight of 0 indicates no knowledge of the relationship between the two instances. We encode both intra-dataset and inter-data set relationships into one matrix W which has a blockwise structure as shown in Figure 2.

Let $W \in [-1, 1]^{(|X|+|Y|)\times(|X|+|Y|)}$ be the matrix which captures all these relations. The upper left and bottom right

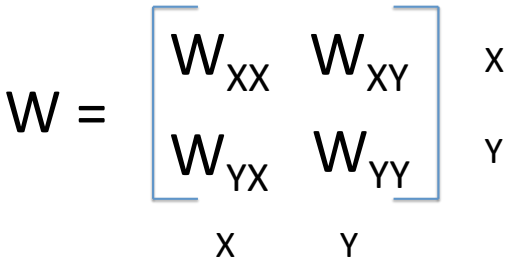

Figure 2: The block structure of W for two different data sets. As we can see W consists of several different matrices. W_{XX} and W_{YY} encodes the intra-dataset relationships and W_{XY} and W_{YX} the intra-dataset relationships.

blocks contain the intra-dataset relationships between the instances in X and Y respectively. The upper right and bottom left entries will contain the inter-dataset similarities/dissimilarities between instances in X and Y.

We can now examine how the guidance used in related work can be encoded in W. Earlier Manifold Learning methods such as Locality Preserving Projections (LPP) and its variations [3, 9] are single view methods that take guidance in the form of an adjacency matrix A_G. They can be viewed as only allowing intra-data set constraints for a single data set. Thus, the weight matrix for LPP would be:

$$W_{LPP} = \begin{bmatrix} A_X & 0 \\ 0 & 0 \end{bmatrix}$$

Multi-view LPP (MVLPP) [27] is a multi-view extension of Locality Preserving Projections. Intra-dataset guidance can be given for each data set and is encoded in matrices A_X and A_Y. Within our framework, W would be:

$$W_{MVLPP} = \begin{bmatrix} A_X & I \\ I & A_Y \end{bmatrix}$$

where I is the identity matrix. The identity matrix in the off diagonal blocks of W_{MVLPP} captures the multi-view assumption. Note that as this is a multi-view method, X and Y must have the same number of instances which is not the case for our method.

Canonical Correlation Analysis (CCA) also makes the multi-view assumption but only specifies **inter-dataset** relations which can be captured as follows:

$$W_{CCA} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

We now discus our formulation which allow both intra-dataset and inter-dataset relationships.

2.2 Our Spectral Formulation

We wish to use the previously mentioned relational information (encoded in W) to project all data into a common lower dimensional space using projection vectors $\mathbf{a}$ and $\mathbf{b}$. Letting $\mathcal{I}$ and $\mathcal{J}$ be index sets over X and Y we can write the optimization problem of learning two sets of projection vectors which respect the given guidance as

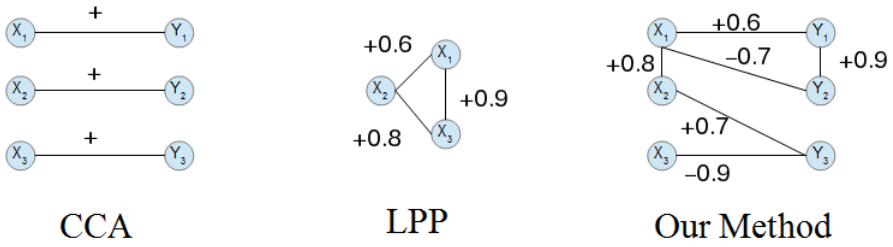

Figure 1: Different types of data embedding methods for data sets X and Y. CCA only allows **similar, unweighted** relationships between instances in different data sets. Additionally, CCA only allows each instance to participate in a single relationship. LPP only allows **similar, weighted** relationships: relationships between instances in the same data set. Our method allows both inter-dataset and intra-dataset relationships that can be weighted, similar or dissimilar with no cardinality restrictions.

$$\arg\min_{\mathbf{a},\mathbf{b}} \frac{1}{2}\Big(\sum_{i_1\in\mathcal{I},i_2\in\mathcal{I}} (x_{i_1}^T\mathbf{a} - x_{i_2}^T\mathbf{a})^2 w_{i_1 i_2} + 2\sum_{i\in\mathcal{I},j\in\mathcal{J}} (x_i^T\mathbf{a} - y_j^T\mathbf{b})^2 w_{ij} + \sum_{j_1\in\mathcal{J},j_2\in\mathcal{J}} (y_{j_1}^T\mathbf{b} - y_{j_2}^T\mathbf{b})^2 w_{j_1 j_2}\Big) \tag{1}$$

The first and third terms of equation 2 optimize the intra-dataset embedded distance between instances in X and Y respectively based on the relational information in W, while the second term optimizes the inter-dataset embedded distance between instances in X and Y. The objective function matches our intuition and can even handle conflicting guidance. Since our aim is to minimize the objective function a positive w_{ij} indicates that the projection vectors should place the two instances close together to minimize the objective. Conversely, a negative value of w_{ij} indicates the two instances should be projected to be far apart to minimize the objective value. Since w_{ij} is weighted the algorithm may choose to satisfy guidance with larger weights at the cost of guidance with smaller weights.

Through some linear algebra (explained below) and introducing a constraint so there are not infinite solutions, equation 2 can be transformed into the following:

$$\begin{aligned} &\arg\min_{\mathbf{p}} \ \mathbf{p}^T Z(D-W)Z^T\mathbf{p} \\ &\textbf{subject to: } \ \mathbf{a}^T(XD_{XX}X^T + \lambda I)\mathbf{a} = 1 \\ &\textit{where we define:} \\ &\mathbf{p} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \quad Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix} \end{aligned} \tag{2}$$

Figure 3: Our formulation to embed two data sets, it can easily be extended to multiple data sets, as discussed later.

We spend the rest of this section deriving equation 2, which can be skipped on the first reading of this paper. The first of the three terms in equation 2 can be expanded to:

$$\frac{1}{2}\Big(\sum_{i_1\in\mathcal{I},i_2\in\mathcal{I}} \mathbf{a}^T x_{i_1} w_{i_1 i_2} x_{i_1}^T\mathbf{a} - \sum_{i_1\in\mathcal{I},i_2\in\mathcal{I}} \mathbf{a}^T x_{i_1} w_{i_1 i_2} x_{i_2}^T\mathbf{a} - \sum_{i_1\in\mathcal{I},i_2\in\mathcal{I}} \mathbf{a}^T x_{i_1} w_{i_1 i_2} x_{i_2}^T\mathbf{a} + \sum_{i_1\in\mathcal{I},i_2\in\mathcal{I}} \mathbf{a}^T x_{i_2} w_{i_1 i_2} x_{i_2}^T\mathbf{a}\Big) \tag{3}$$

Let D_{XX} be a diagonal matrix such that the entry $D_{XX}(i_1,i_1) = \sum_{i_2\in\mathcal{I}} w_{i_1 i_2}$ when $i_1 \in \mathcal{I}$. This is just a diagonal matrix whose diagonal entries are the sums of the weights of the edges between instances in X. Note that in the first expression of equation 3, for each x_i we can sum over the entire row of W and hence rewrite the term as $\mathbf{a}^T X D_{XX} X^T\mathbf{a}$. The fourth expression can be rewritten in an identical form. The second and third expressions can be written together in matrix form as $-2\mathbf{a}^T X W_{XX} X^T\mathbf{a}$ where W_{XX} corresponds to the block of W relating X to X. Hence equation 3 can be more concisely written as

$$\mathbf{a}^T X D_{XX} X^T\mathbf{a} - \mathbf{a}^T X W_{XX} X^T\mathbf{a} \tag{4}$$

Letting $L_{XX} = D_{XX} - W_{XX}$ we get

$$\mathbf{a}^T X L_{XX} X^T\mathbf{a} \tag{5}$$

A similar transformation can be performed on the second and third terms in equation 2 to get

$$\arg\min_{\mathbf{a},\mathbf{b}} \mathbf{a}^T X L_{XX} X^T\mathbf{a} + \mathbf{b}^T Y L_{YY} Y^T\mathbf{b} - 2\mathbf{a}^T X W_{XY} Y^T\mathbf{b} \tag{6}$$

Where W_{XY} and L_{YY} are defined similarly to W_{XX} and L_{XX} respectively. We can then concatenate $\mathbf{a}$ and $\mathbf{b}$ into one vector $\mathbf{p}$, concatenate X and Y into Z as shown in equation 2 and concatenate $W_{XX}, W_{YY}, W_{XY}, W_{YX}$ into W as shown in Figure 2. This allows us to rewrite equation 6 as:

$$\arg\min_{\mathbf{p}} \ \mathbf{p}^T Z(D-W)Z^T\mathbf{p} \tag{7}$$

However, equation 7 can in some circumstances be trivially solved by $\mathbf{p} = \mathbf{0}$ and has infinite number of solutions otherwise so we add the constraint $\mathbf{a}^T(XD_{XX}X^T + \lambda I)\mathbf{a} = 1$ where λ is a small positive constant and I is the identity matrix. This problem can be solved as a generalized eigenvalue problem. We do not add the typical $\mathbf{a}^T XD_{XX}X^T\mathbf{a} = 1$ constraint because in order for this problem to have a real (i.e. non-complex) solution $XD_{XX}X^T$ must be made positive definite which may not be the case if there is more negative guidance for an instance that positive guidance or X has fewer training instances than features. Note that the constraint need not include $\mathbf{b}$ since it is coupled with $\mathbf{a}$ via W.

Extensions to More Than Two Data Sets. This formulation can naturally be extended to m data sets (hence producing projection vectors $\mathbf{a}_1 \dots \mathbf{a}_m$) by further adding to W and Z, letting $\mathbf{p}^T = [\mathbf{a}_1^T \dots \mathbf{a}_m^T]$ and using the formulation in equation 8

$$\begin{aligned} &\arg\min_{\mathbf{p}} \ \mathbf{p}^T Z(D-W)Z^T\mathbf{p} \\ &\textbf{subject to: } \mathbf{a}^T(XD_{XX}X^T + \lambda I)\mathbf{a} = 1 \\ &\textit{where we define:} \\ &\mathbf{p} = \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \\ \dots \\ \mathbf{a_m} \end{bmatrix} \quad Z = \begin{bmatrix} X_1 & 0 & \dots & 0 \\ 0 & X_2 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & X_m \end{bmatrix} \end{aligned} \tag{8}$$

Figure 4: Our formulation for m data sets.

Obtaining multiple projection vectors (the k smallest eigenvectors for instance) allows embedding instances into a k dimensional space.

2.3 Computational Complexity of Method.

As mentioned previously, equation 7 can be solved as a generalized eigenvalue problem. The key operations of this are solving the generalized eigenvalue problem and matrix inversion. For solving the generalized eigenvalue problem we used the `eig` function of MATLAB, which uses the QZ method. The complexity of the QZ method is approximately $\mathcal{O}(66N^3)$ and the space requirement is $\mathcal{O}(N^2)$ where N is the number of dimensions of X [4]. However, this will return all eigenvectors and we only require the k smallest if we are to embed the data into k dimensional space. Instead, libraries such as ARPACK [12] (the library used by Matlab's `eigs` function) that implement more scalable eigenproblem methods can be used. In all our experiments the total runtime was less than a second on a typical quad-core laptop.

Some of the other methods discussed in the paper - CCA, LPP and Manifold Alignment - also reduce to generalized eigendecomposition problems. Depending on the solver, matrix inversion may also be required. Thus, these methods have comparable complexity. Metric learning methods require solving a Semidefinite Program (SDP). While SDP solvers can run in polynomial time with respect to the size of the problem, but because SDPs tend to have many variables they generally do not scale as well to high dimensional problems [21].

2.4 Kernelizing the Method

A limitation to our method as presented is that it can only learn a linear transformation. This can be efficiently addressed by using the "kernel trick" to learn a nonlinear transformation. The kernelized version of our formulation is:

$$\begin{aligned} &\arg\min_{\rho} \rho^T K(D-W)K^T\rho \\ &\textbf{subject to: } \alpha^T K_{XX}D_{XX}K_{XX}^T\alpha = 1 \end{aligned} \tag{9}$$

where we define

$$\rho = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad Z = \begin{bmatrix} K_{XX} & 0 \\ 0 & K_{YY} \end{bmatrix}$$

K_{XX} and K_{YY} are kernels for X and Y. α and β are the dual variables associated with X and Y.

In our experiments we use a combination of Kernels and features to describe our data. See the experimental section for details.

3. EXPERIMENTS

Here we propose experiments to compare our framework to CCA. In particular we shall ask the following questions:

- How well does the embedding respect the relational information used to drive it? (Figure 5 & Table 6).
- How does our method compare against CCA for a simple problem with just two data sets of instances (Figures 7, 8 red and green lines)?
- Can embedding more datasets using more guidance improve results (Figure 7, 8 blue line)?
- Does clustering the heterogenous collection of instances produce meaningful clusters (Figure 9) and accurate clusters (Figure 10)?.

All data sets and code will be made publicly available once the paper is accepted so as to recreate the experimental results presented here.

Our Data Set. We focus on personal information management problems since they are not only important but easy for a non-expert to verify. We believe this setting is an ideal application of our work because personal information management involves finding structure in sets of heterogeneous instances in which arbitrary intra-dataset and inter-dataset relationships are available.

PIM data sets containing video, text and images like the one mentioned in the introduction are not readily available so we use the one made publicly available in [29] that contains images, text-tags and location information. The data set consists of 500 images taken at 99 different locations (described using longitude and latitude) around Asheville, North Carolina. The images are tagged with one or more of 590 possible descriptors. Each image is associated with a single location, but images can have multiple tags and each tag can be associated with multiple images. For our experiments, images are represented using SIFT features [14] which are used to construct a kernel, tags are represented by simple indicator vectors that are all 0 except for the entry corresponding to the tag which is 1 and locations are represented using a Gaussian Kernel applied to the longitude and latitude values.

Now we give the method for constructing the guidance matrix W. For each image-tag tag entry in W, we set it to 1

if the image has the tag, 0 otherwise. Similarly, for experiments that used locations, we set each image-location entry in W to 1 if the image was taken at that location and 0 otherwise. All other entries were set to 0.

For all experiments we selected the top N most common tags (where N varies based on the experiment) excluding the five most common tags in the data set because they are associated with a large majority of the images.

For all experiments except the embedding experiment in section we 3.1:

- Images were split into training and test sets. The embedding was learned using the training data and its associated guidance and results are reported with respect to the test set guidance.
- Set regularization parameters and the number of projection vectors using cross validation.

We hope our method can perform better than or comparable to CCA using the same guidance and perform significantly better when more guidance is used.

For our experiments we compared four methods:

- **Ours:** Our method embedding images and tags.
- **Ours+Locations:** Our method embedding images, tags as well as locations.
- **CCA:** CCA embedding images and tags.
- **Guess:** Always predicting the K most common tags.

3.1 Embedding Experiments

Here we test how well our method embeds the data. We purposely use all available relational information (images to tags and images to locations) to see how well it works with 1000s of pieces of relational guidance. The resultant embedding is shown in Figure 5 and shows instances from all data sets are well interspersed in the embedding. Another test of our embedding is how well the relational information used to drive the process is respected in the embedding. We would hope our learning of non-linear projection vectors will allow for most of the guidance to be respected. Figure 6 shows how well the guidance is respected both in terms of the guidance given to the algorithm (training left plot) and guidance not given to the algorithm (testing right plot). In the left plot we see our method performs significantly better than CCA and adding extra guidance does not diminish its ability to satisfy the guidance. An important result is that in our embedding over 90% of guidance is satisfied by 50% of the closest pairs.

As expected, the amount of guidance satisfied is not as great for the test set (because the test is not used to learn the embedding). Nevertheless, the fraction satisfied is significantly greater than CCA.

3.2 Classification Experiments

The type of guidance our methods can handle is novel so there is not an elegant way to apply CCA using the guidance we want to encode. Hence, we implicitly encode the relationships by duplicating entries. For example, if an image x_i is associated with two tags y_j, y_k, then we augment the data set used by CCA with two pairs: (x_i, y_j) and (x_i, y_k).

For our classification experiments we randomly removed 20% of the images for the test set and the remaining were used for training. This was repeated 10 times for each experiment. The experiments we ran learned a set of projection vectors given a training set of images and tags and then used them to embed the training and test images. To predict tags for the test set we simply use the k nearest tags in the embedded space.

For accuracy we measured the Normalized Cumulative Discounted Gain (NCDG) [11] of the predicted tags. NDCG is a measure of how well an instances nearest tags, **ordered** from closest to furthest, match the order of the tags as given in the data set (by a domain expert). That is if an instance has q tags, then we retrieve its q nearest tags and see how well that ordering matches the ground truth. This is a much more informative measure of performance than measures such as the Rand index or precision since it factors in the orders of the tags.

For all experiments cross validation was used to choose regularization parameters and the number of projection vectors. Figure 7 shows the performance of our method and CCA versus varying numbers of *maximum* projection vectors (the actual number of projection used was still selected using cross validation). We see our methods performs significantly better than CCA. Furthermore, our method is able to obtain stronger results using fewer projection vectors. We conjecture our better performance is because our formulation is based on minimizing embedded distance while CCA is based on maximizing embedded covariance [7].

Figure 8 shows the performance of our method and CCA without constraints on the **maximum** number of projection vectors used (essentially the right-most extreme points in Figure 7 versus varying size training sets. We measure performance using NDCG as before and include in our comparison a baseline method of guessing the most frequent tags (`Guess`). As before if a test set instance has q tags, then we retrieve the q nearest tags to the instance and note their order and compare this ordering to the ground truth. We see that for very small data sets all methods perform similarly but as more training data becomes available our method is able to perform significantly better than CCA which itself only performs marginally better than `Guess`. This shows our method is able to handle lots of guidance as with each additional training instance comes more guidance and a more complex W matrix.

3.3 Clustering Experiments.

A particularly novel use of heterogeneous embedding is that it allows applying unsupervised learning techniques to instances from multiple data sets. This could involve ranking/retrieval, organizing (such as by using hierarchical clustering methods) or, as we show here, clustering heterogeneous instances. As before, we use relationships between images and their geographic location as well as between tags and images to embed images, tags and locations into a common space. We then run standard k-means clustering on the embedded images, tags and locations.

First we show that our method can lead to meaningful clusterings. For k-means we set the number of clusters to 10. Due to space restrictions, Figure 9 shows the images, tags and locations closest to the cluster centroids for just 3 such clusters (all clusters are shown in Figure 5. From these experiments we see that the resulting clusterings contain a mix of similar tags, locations and images. Importantly, we see that the images and locations within each cluster are consistent with each other. The left-hand side cluster contains of a series of parks outside the downtown area, the right-hand side clusters are of two downtown attractions: an annual rock festival and another cultural event.

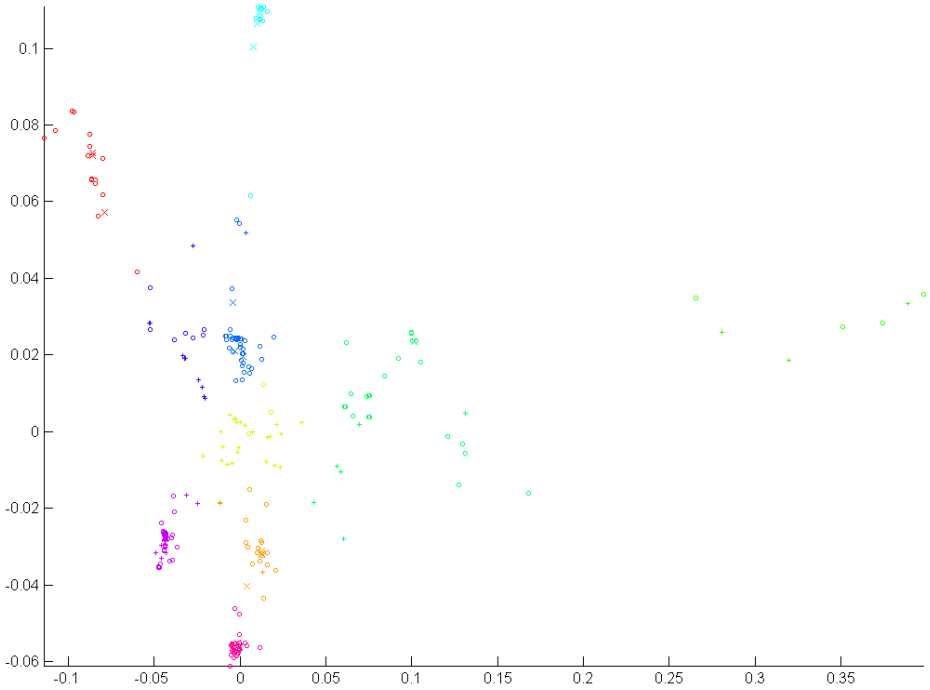

Figure 5: Visualization of embedding produced using our method on images, tags and locations using 30 tags. Circles correspond to images, 'x' to tags and '+' to locations. instances are color coded based on a clustering found by running K-Means Clustering.

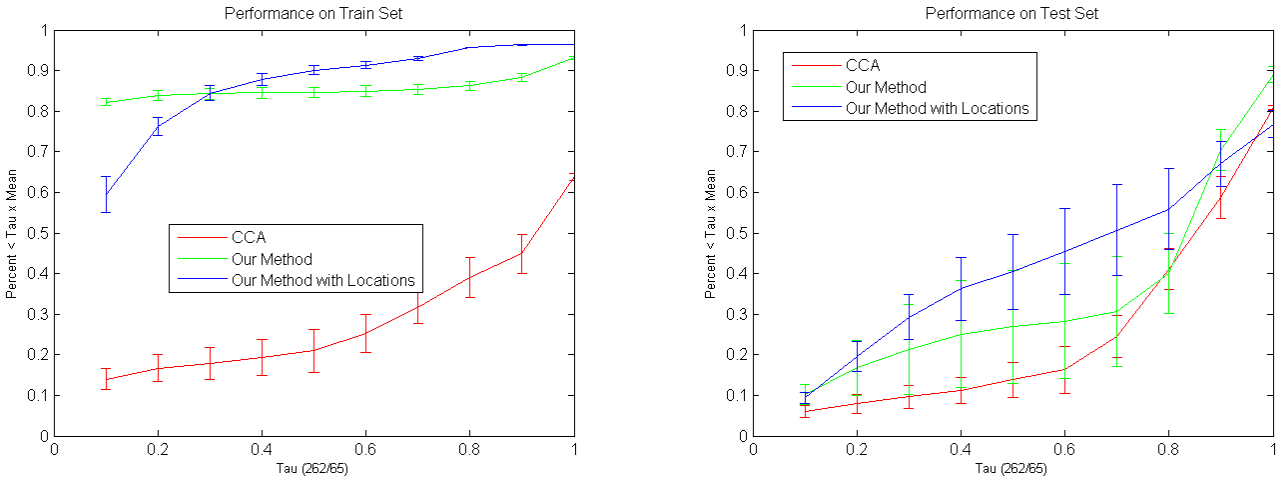

Figure 6: The fraction of positively weighted image-tag pairwise relationships whose distance is less than τ times the mean image-tag distance for *all* image-tag pairs in the embedding. The left shows the performance on the training set while the right shows the performance on the test set.

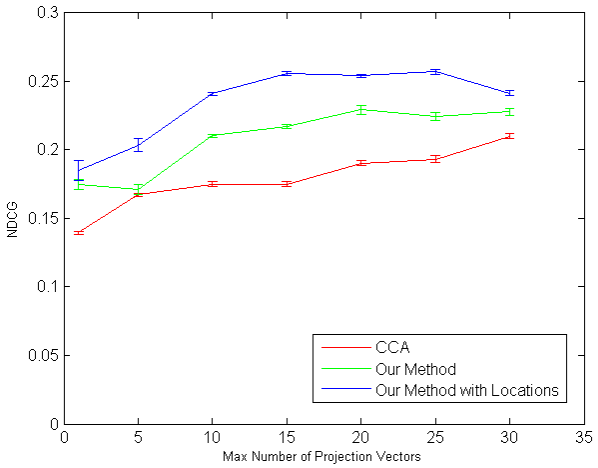

Figure 7: NDCG for image-tag nearest neighbor lookup versus the number of projection vectors. Number of tags was fixed to 30 and training size was all available training images.

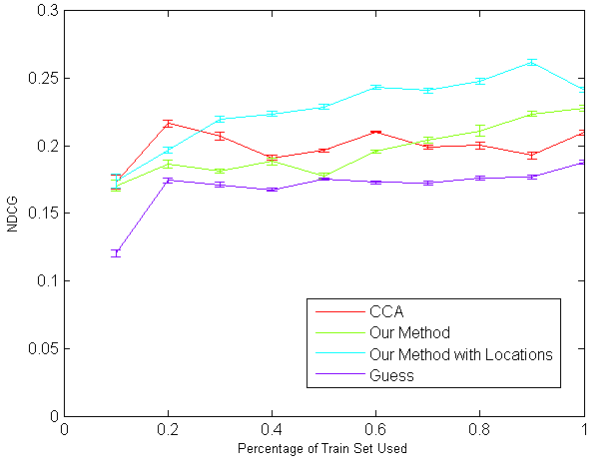

Figure 8: NDCG for image-tag nearest neighbor lookup for increasing training set size. Number of tags was fixed to 30 and maximum number of vectors fixed to 30.

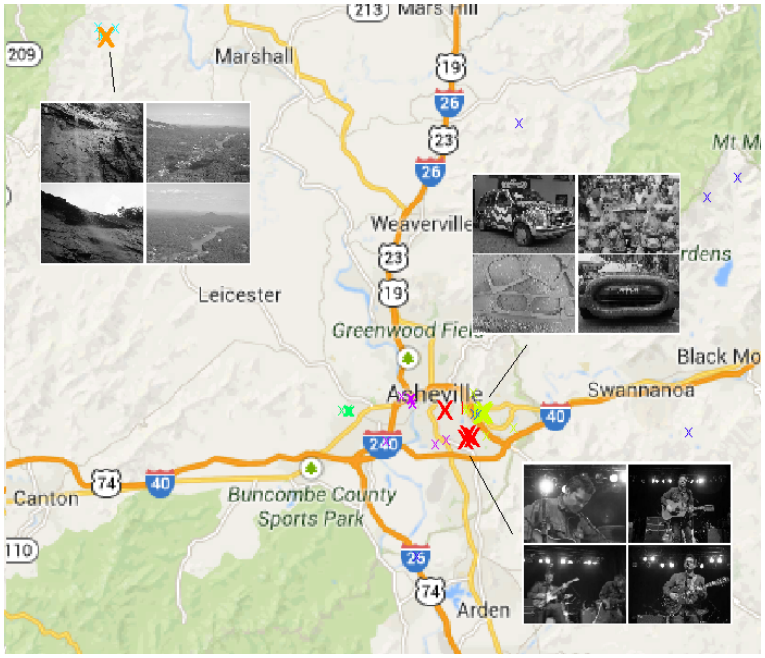

Figure 9: Locations are marked with an X. Colors represent different clusters constructed using K-Means Clustering. Enlarged X's represent locations closest to centroids of three clusters. The four images closest to the centroids of the clusters are pictured as well with a line indicated the location of the images. Tags closest to the centroids of the three clusters are: **Red**: wilco, geotagged, livemusic, orangepeel, **Orange**: chimneyrockpark and **Yellow**: laaff, lexingtonave, lexingtonavenue, lexfest. Map image was generated using Google Maps; Map Data: Google 2014.

Though visual inspection of the clusters is useful, a more quantitative measure of performance is to compute the Rand Index for varying training set sizes and number of clusters. The Rand index here measures how well the image-tag relations are respected by the clusters. A value of 1 means a perfect matching. We hope that each cluster will contain images with similar tags. For these experiments we learn projection vectors on a training set, apply them to a test set, cluster the test set and report the Rand Index. This was repeated for 10 different training and test sets. Our results are shown in Figure 10. We see that regardless of the number of clusters, our method out performs or performs comparable to CCA and again adding in more guidance (and objects) produces even better results.

4. RELATED WORK

The related work falls into a number of distinct areas which we now survey.

Graph Driven Dimension Reduction. Much work has been done on graph based embeddings for machine learning. Locally Linear Embedding [19] and Laplacian Eigenmaps [2] are both popular graph based embedding techniques. Here we can view the graphs as encoding intra-dataset `similar` relationships which are used to embed one data set. [17], [8] and [9] learn projection vectors for out-of-sample embeddings. Many graph embedding methods were unified by the graph embedding framework of [25]. Multi-view extensions to these techniques have been proposed such as [18, 27] but, unlike our work, neither allows arbitrary inter-dataset relationship.

Manifold Alignment. Manifold Alignment methods [6, 22] model multiple data sets as lying on manifolds and attempts to learn projections which align these manifolds. The key different between their work and ours is they model the data as lying on a manifold, while we do not.

Multi-View Dimension Reduction. Multi-view dimensionality reduction has also received a great deal of attention in the machine learning community. Canonical Correlation Analysis (CCA) [10] is a classical technique that has been applied to multi-view dimensionality reduction [7]. Recent work has considered the use of 3 view CCA for embedding images, tags and semantic classes [5]. These works can be viewed as embedding multiple data sets but under very limited relational information, namely an instance from a data set can only have a relationship with a single instance from another data set.

Heterogeneous Learning. Learning with multiple sources has received some attention. The problem of clustering with heterogenous instances and side information was addressed in [29]. [13] and [26] studied *Heterogeneous Transfer Learning* in which relational information is used to transfer knowledge from different domains. In [13] the source domain is documents and the target domain is images labeled with tags while in [26] the source and target domains are text documents in different languages.

Multi-modal Similarity and Metric Learning. Learning with heterogeneous data has been addressed using Similarity Learning [15] and Metric Learning [23,28]. While these works address a similar problem, they differ greatly from our work. First, none of these works propose incorporating weighted guidance and [28] does not allow intra-dataset guidance. [15] requires solving an expensive semidefinite program. [23] suggests a graphic model for metric learning with heterogeneous data with the goal of being more scalable

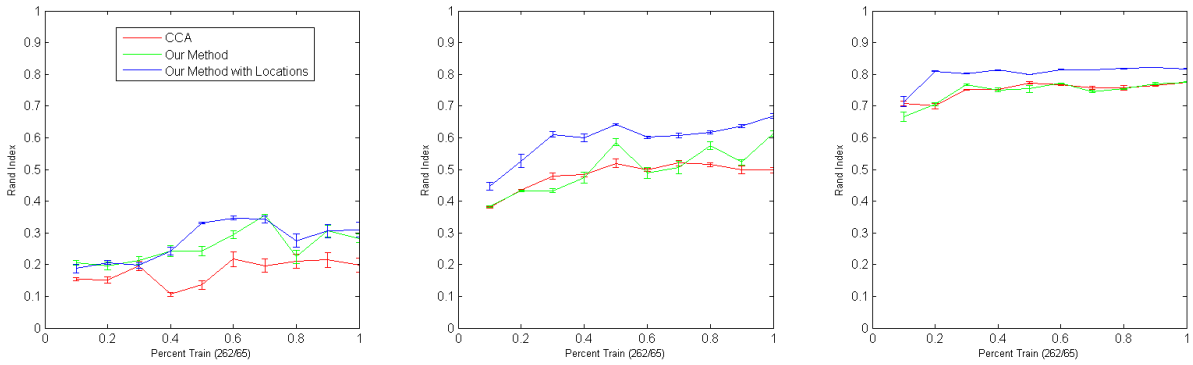

Figure 10: Rand Index for clusterings found via k-Means with preprocessing by CCA, our method and our method also using locations. From left to right, $k = 2, 5, 10$.

than [15]. While their framework is more scalable, the optimization problem they propose cannot be solved for exactly and has many parameters that need to be set which makes it more difficult to use than our framework.

5. CONCLUSION

The complexity of modern data sets has made analysis a great challenge. A growing tend is for data to be heterogeneous consisting of fundamentally different data from different sources. An elegant method is to project such data to a common space so standard algorithms can be used, but existing methods make strong assumptions such as there being overlapping instances or features to perform this projection. We proposed a flexible technique for embedding heterogeneous data into a common space. Our method differs from existing work in that it can provide weighted, positive and negative guidance using both inter-dataset and intra-dataset relationships. Most importantly, it does not assume the instances or features in the different data sets overlap. Existing multi-view work such as CCA typically can only provide positive unweighted guidance of limited cardinality (that is each instance can be paired with only a single other instance). Though manipulating the data can help alleviate some of these situations, such as duplicating instances to remove the cardinality restriction, we seek a more principled method.

Our spectral formulation learns a projection vectors for each data set such that intra-dataset and inter-dataset guidance is respected. Guidance takes the form of weighted values between -1 (dissimilar) and 1 (similar) with no limits on the amount of guidance given for a particular instance. In our experimental results testing the embedding (see Figure 6) we see that our method does a good job of of embedding the data to respect the guidance given to it. In our experiments our method performs better than or at least comparable to CCA for both classification and clustering applications. Our classification experiments use the embedding our method learns to perform K-Nearest Neighbors classification (see Figures 7 and 8). Our results on clustering show that it produces meaningful clusterings (see Figure 9) whose Rand index (see Figure 10) is better than those achieved by a comparable method.

Our plans for future work are to explore further uses of embedding data into a common space and in particular explore new forms of guidance. Additionally, we plan to investigate connections between our work and Spectral Graph Theory.

6. ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this research via ONR grants N00014-09-1-0712, N00014-11-1-0108 and NSF Grant NSF IIS-0801528.

7. REFERENCES

[1] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[3] Ian Davidson. Knowledge driven dimension reduction for clustering. In *IJCAI*, 2009.

[4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, 1996.

[5] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014.

[6] Jihun Ham, Daniel Lee, and Lawrence Saul. Semisupervised alignment of manifolds. In *10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[7] David R. Hardoon, Sandor Szedmak, Or Szedmak, and John Shawe-taylor. Canonical correlation analysis; an overview with application to learning methods. Technical report, 2007.

[8] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *ICCV*, 2005.

[9] Xiaofei He and Partha Niyogi. Locality preserving projections. In *NIPS*, 2003.

[10] Harold Hotelling. Relations between two sets of variables. In *Biometrika*, 1936.

[11] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR*, 2000.

[12] R. B. Lehoucq, D. C. Sorensen, and C. Yang. Arpack users guide: Solution of large scale eigenvalue problems by implicitly restarted arnoldi methods., 1997.

[13] Yuan Lin, Yuqiang Chen, Gui-Rong Xue, and Yong Yu. Text-aided image classification: Using labeled text from web to help image classification. In *APWeb*, 2010.

[14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[15] Brian McFee and Gert R. G. Lanckriet. Learning multi-modal similarity. *CoRR*, 2010.

[16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10), 2010.

[17] Yanwei Pang, Lei Zhang, Zhengkai Liu, Nenghai Yu, and Houqiang Li. Neighborhood preserving projections (npp): A novel linear dimension reduction method. In *Advances in Intelligent Computing*, Lecture Notes in Computer Science, pages 117–125. 2005.

[18] Novi Quadrianto and Christoph Lampert. Learning multi-view neighborhood preserving projections. In *ICML*, 2011.

[19] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000.

[20] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *KDD*, 2008.

[21] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 1996.

[22] Chang Wang and Sridhar Mahadevan. A general framework for manifold alignment, 2009.

[23] Pengtao Xie and Eric P. Xing. Multi-modal distance metric learning. In *IJCAI*, 2013.

[24] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2003.

[25] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong jiang Zhang. Graph embedding: A general framework for dimensionality reduction. In *CVPR*, 2005.

[26] Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the social web. In *ACL*, 2009.

[27] Xuesong Yin, Qi Huang, and Xiaodong Chen. Multiple view locality preserving projections with pairwise constraints. In *Communication Systems and Information Technology*. 2011.

[28] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. Heterogeneous metric learning with joint graph regularization for cross-media retrieval. In *AAAI*, 2013.

[29] Weifeng Zhi, Xiang Wang, Buyue Qian, Patrick Butler, Naren Ramakrishnan, and Ian Davidson. Clustering with complex constraints - algorithms and applications. In *AAAI*, 2013.

[30] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.