Statement of Research Interests

Geoffrey Iyer

I. Introduction

The main focus of my research is in multimodal data and data fusion. With the increasing availability of data we often come upon multiple datasets, derived from different sensors, that describe the same object or phenomenon. We call the sensors *modalities*, and because each modality represents some new degrees of freedom, it is generally desirable to use more modalities rather than fewer. For example, in the area of speech recognition researchers have found that integrating the audio data with a video of the speaker results in a much more accurate classification [1]. However, correctly processing a multimodal dataset is not a simple task [2]. The main difficulty in multimodality lies in finding a way to coordinate information that is represented in different formats. In the current state-of-the-art, most multimodal algorithms apply only to very specific types of data, and are therefore only useful for a small class of problems. Therefore it is desirable to produce an algorithm that is robust to many different types of input data. This has been the motivation of my PhD research.

The direction of my research group is heavily based around the representation of the data as a weighted graph and the information we can glean from the associated graph Laplcian (section II). The graph representation allows us to discard the specifics of each data format and create objects that can be directly compared. From this, we created an algorithm segments a multimodal dataset under the assumption that the data is co-registered (the modalities share a common indexing). In section III we give a brief overview of the work done, and the full version can be found in [3]. Our second algorithm is a work in progess, and is the focus of the remainder of my PhD. Here we focus on eliminating the coregistration assumption by introducing a matching process based on the geometry of each dataset. An overview of the work done here can be found in section IV. Finally, in section V we outline the relevance to the Anticipatory Analysis project.

II. Graph Representation

For notation, we label our different modalities as $X^1, X^2, \ldots, X^k$. We represent each X^ℓ using an undirected graph $G^\ell = (V^\ell, E^\ell)$. The nodes $v_i^\ell \in V^\ell$ of the graph correspond to elements of X, and we give each edge e_{ij}^ℓ a *weight* $w_{ij}^\ell \geq 0$ representing the similarity between nodes v_i^ℓ, v_j^ℓ, where large weights correspond to similar nodes, and small weights to dissimilar nodes. This gives rise to a *similarity matrix* (also called the *weight matrix*)

$$W^\ell = \left(w_{ij}^\ell\right)_{i,j=1}^n . \tag{1}$$

There are many different notions of "similarity" in the literature, and each has its own merits. In particular, it is common to use an RBF kernel

$$w_{ij} = \exp\left(-\left\|v_i - v_j\right\| / \sigma\right), \tag{2}$$

where the details of the choice of norm and scaling parameter σ depend on the specific application.

Given a similarity matrix W, we then define the *normalized graph Laplacian*. For each node $v_i \in V$, define the *degree* of the node $d_i = \sum_j w_{ij}$. Intuitively, the degree represents the strength of a node. Let D be the diagonal matrix with d_i as the i-th diagonal entry. We then define the normalized graph Laplacian.

$$L_{sym} = I - D^{-1/2} W D^{-1/2}. \tag{3}$$

For a thorough explanation of the properties of the graph Laplacian, see [4]. Here we use the fact that the eigenvectors of the graph Laplacian solve the *relaxed Graph min-cut problem*

$$\text{argmin}_{Q^TQ=I} \text{Tr}\left(Q^T L_{sym} Q\right), \tag{4}$$

and can be considered as features extracted from the dataset.

A. Nystrom Extension

Calculating the full graph Laplacian is computationally intensive, as the matrix contains n^2 entries. Instead we use Nyström's extension to find approximate eigenvalues and eigenvectors with a heavily reduced computation time [5].

Let X denote the set of nodes of the complete weighted graph. We choose a subset $A \subset X$ of "landmark nodes", and have B its complement. Up to a permutation of nodes, we can write the weight matrix as

$$W = \begin{pmatrix} W_{AA} & W_{AB} \\ W_{BA} & W_{BB} \end{pmatrix}, \tag{5}$$

where the matrix $W_{AB} = W_{BA}^T$ consists of weights between nodes in A and nodes in B, W_{AA} consists of weights between pairs of nodes in A, and W_{BB} consists of weights between pairs of nodes in B. Nyström's extension approximates W as

$$W \approx \begin{pmatrix} W_{AA} \\ W_{BA} \end{pmatrix} W_{AA}^{-1} \begin{pmatrix} W_{AA} & W_{AB} \end{pmatrix}. \tag{6}$$

In fact, it is possible to find $|A|$ approximate eigenvectors of W using only the matrices W_{AA}, W_{AB}. This results in a significant reduction in computation time, as we compute and store matrices of size at most $|A| \times |X|$, rather than $|X| \times |X|$.

III. Feature Extraction and Segmentation on Multimodal Datasets

In this section we assume that our datasets are *co-registered*. That is, the sets share a common indexing. Let $n = |X^1| = \cdots = |X^k|$, and form the concatenated set

$$X = (X^1, \ldots, X^k) \subseteq \mathbb{R}^{n \times (dim_1 + \cdots + dim_k)}. \tag{7}$$

to define the similarity matrix W, we must somehow compare the individual $X^1, \ldots, X^k$. We do this by comparing distances between points in each dataset. For $\ell = 1, \ldots, k$ define the scaling factor

$$\lambda_\ell = \text{stdev}\left(\|x_i^\ell - x_j^\ell\|_{L^2} \ : \ 1 \le i, j \le n \right) \tag{8}$$

We then define the weight matrix W by

$$w_{ij} = \exp\left(-\max\left(\frac{\|x_i^\ell - x_j^\ell\|_{L^2}}{\lambda_\ell} \ : \ 1 \le \ell \le k \right) \right). \tag{9}$$

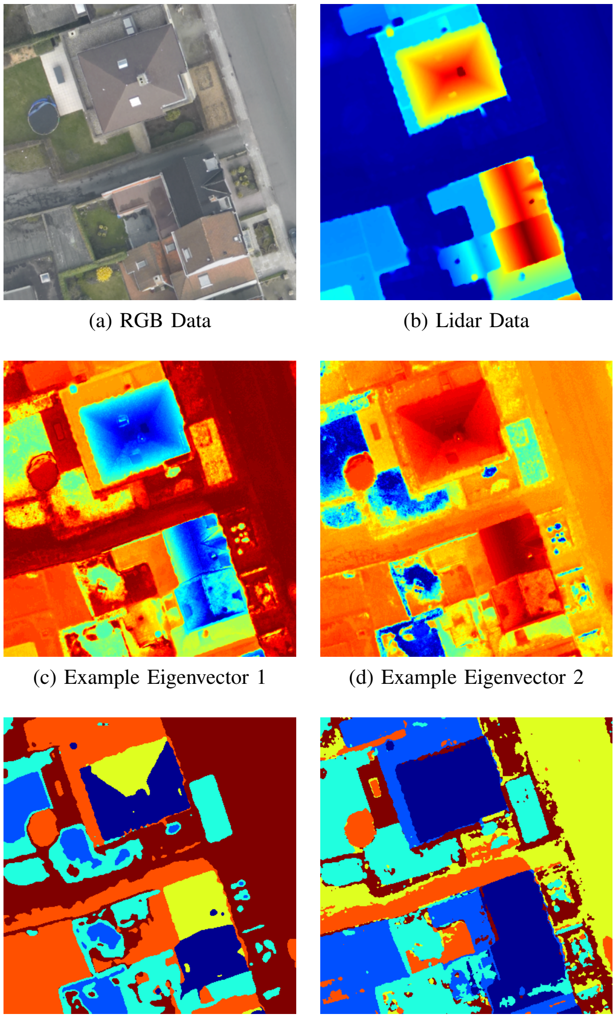
We specifically choose to use the maximum of the individual measurements to emphasize the unique information that each dataset brings. With this norm, two data points x_i, x_j are considered similar only when they are similar in every modality.

Based on section II, we can extract features from X via graph Laplacian theory applied to the matrix W. From these features we then apply segmentation algorithms to the feature set to obtain the final output. One simple and straightfoward segmentation method is *Spectral Clustering*, in which we apply k-means to the set of features. We also implemented a semisupervised graph MBO method [6], in which we minimize a Ginzburg-Landau energy

$$E(u) = \epsilon \cdot \text{Tr}\left(u^T L_{sym} u\right) + \frac{1}{\epsilon} \sum_i W(u_i) \tag{10}$$

$$+ \sum_i \frac{\mu}{2} \lambda(x_i) \|u_i - \hat{u}_i\|_{L_2}^2 . \tag{11}$$

by iterative diffusion and thresholding. In figure 1, we show the results of our algorithm applied to the Data Fusion Contest 2015 dataset, which consists of both lidar and RGB satellite images. Note that in the example feature vectors 1c 1d, objects are separated based both on color and on elevation. These separations are then reflected in the final segmentation result.


(a) RGB Data

(b) Lidar Data

(c) Example Eigenvector 1

(d) Example Eigenvector 2

(e) Spec. Clust. (Unsupervised)

(f) MBO (Semisupervised)

Fig. 1: DFC2015 Data and Results

IV. Graph Matching

The above algorithm gives an effective way of comparing multiple modalities, but for some problems we cannot assume our sets are coregistered. To address this problem, we look to create our own registration by using the geometry of the data. One common formulation of this problem is the Weighted Graph Matching Problem (WGMP), described below.

Let $G = (X, W_X)$, $H = (Y, W_Y)$ be undirected, weighted graphs. Here X, Y represent the nodes of G, H (respectively), and W_X, W_Y are the corresponding matrix of edge weights. For convenience of notation we will assume $|X| = |Y| = N$. The extension to the general case is quite straightforward. The goal of the WGMP is to find a bijection $\rho : X \to Y$ that minimizes the squared difference of edge weights. Phrased in terms of matrices, our minimization problem becomes

$$\text{argmin}_{P \text{ a permutation matrix}} \left\| P W_X P^T - W_Y \right\|_F^2 . \tag{12}$$

Finding an exact solution to this problem is NP-Hard [7]. Instead, we look for an approximate solution via the methods presented in [8]. Specifically, we relax the problem and search for an orthogonal matrix.

$$Q^* = \text{argmin}_{QQ^T = I} \left\| Q W_X Q^T - W_Y \right\|_F^2 . \tag{13}$$

This problem was solved theoretically in [8] using eigenvectors of the graph Laplacian. Let $L_X = U_X \Lambda_X U_x^t$, be the eigendecomposition of the graph Laplacian of X, and similarly have $L_Y = U_Y \Lambda_Y U_Y^T$. Then the spectral graph matching theorem from [8] states that if each L_x, L_y has distinct eigenvalues, the optimal Q from (13) is given by

$$Q^* = U_Y^T S U_X, \tag{14}$$

where S is a diagonal matrix with values ± 1 to account for the sign ambiguity in eigenvectors. In the current state of our project, we are using a semi-supervised method to determine S. If we know of even one preexisting match between the nodes in X, Y, we can use this to compare eigenvectors U_X, U_Y and add the appropriate signs.

Note that if we ignore the sign ambiguity (assume $S = I$), then the solution reduces to $Q^* = U_Y^T U_X$. In other words, we measure the similarity between graph nodes by taking inner products in the eigenvector feature space. To complete our approximate solution to the WGMP, we use the matrix Q^* to find a matching

$$\rho : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}. \tag{15}$$

The task of converting Q^* to a permutation ρ is easily solved using the Hungarian algorithm, which in $O(N^3)$ time finds the matching ρ that maximizes the sum of similarity scores

$$\sum_{i=1}^{N} Q^*(i, \rho(i)). \tag{16}$$

A. Hierarchical graph matching

The advantage of the Hungarian algorithm is that it results in the optimal one-to-one matching based on the input data, but the $O(N^3)$ runtime is an undesirable feature. To solve this we introduce a hierarchical structure to the above algorithm. Specifically, we create smaller "coarsified" graphs $\tilde{G}, \tilde{H}$ of size $M \ll N$. These $\tilde{G}, \tilde{H}$ should represent the same geometric structure as G, H, but with many fewer nodes. We then graph matching algorithm on $\tilde{G}, \tilde{H}$, giving us a match on the coarse level. To lift this to a match between G, H, for each match $i \to j$ in the coarse graphs we run our graph matching algorithm between the corresponding clusters in the original graph. So in total, we create 1 match of size M, and M matches of size $\frac{N}{M}$, which significantly improves the runtime. In practice, our prototype code can handle sets of size up to $N = 100,000$. In theory, we could achieve an even better runtime via better optimizations and parallel processing. We could also speed up the matching process by performing multiple coarsification steps, and iteratively matching the graphs on increasingly fine levels. However, this would give dimishing returns with each added layer, as well as create some issues with error propagation.

B. Example graph matching

Here we show the results of our graph matching algorithm on a synthetic dataset, pictured in figure 2. The nodes of the graphs G, H are represented by 2-dimensional datasets X, Y, and the weight matrices W_X, W_Y are determined via an RBF kernel applied to the 2-norm.

$$E_X(i, j) = \|X(i) - X(j)\| \tag{17}$$

$$W_X(i, j) = -\text{exp}\left(\frac{E_X(i, j)}{\text{stdev}(E_X)}\right), \tag{18}$$

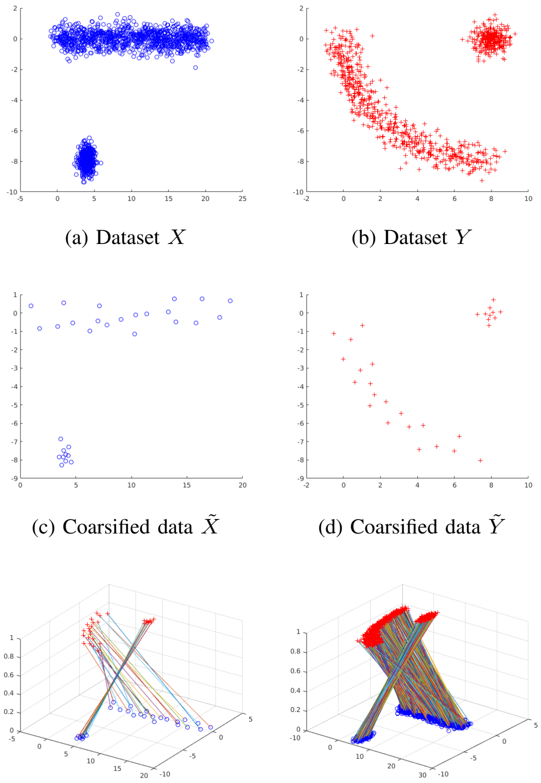
and similar for W_Y. The matching is then calculated using the hierarchical algorithm in IV-A, with the match on the coarse level shown in figures 2c, 2d, 2e. This example has datasets of size $N = 1500$, with $M = 50$, so that the coarsified data has size $\left|\tilde{X}\right| = \left|\tilde{Y}\right| = 30$.

The purpose of this example is to show that the matching algorithm can recognize the geometry of the data. Both sets X, Y contain a tight cluster of points, as well as longer line segment. In figures 2e, 2f, we see the final result of the algorithm, where each match is represented by a line connecting the two points in question. As we can see in the figure, the algorithm successfully matches the objects based on their shape, as we desired.

V. Relevance to Anticipatory Analysis

Speaking in general terms, the Anticipatory Analysis project is deeply rooted in multimodality. The scope of the data involved is very large both in size, as well as in variety of formats, leading to many problems similar to the ones addressed in my work. For example, timeseries data from a multiple different sources can be considered to be coregistered based on observation time. We can then apply the ideas from III to extract feature vectors which can be used for many different applications (including, but not limited to, segmentation).

We could also apply our graph matching algorithm to such a dataset, then compare the indexing given by graph matching against the natural coregistration indexing.

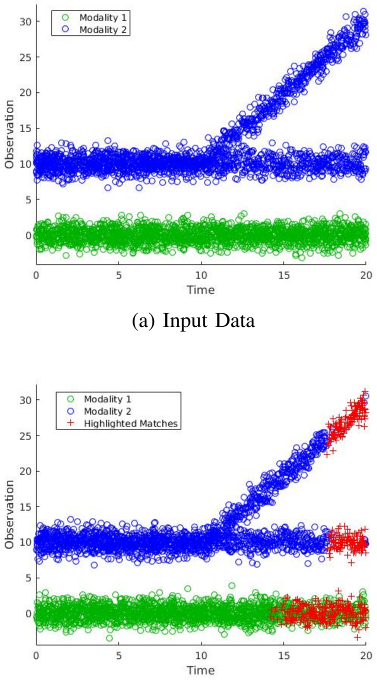


(a) Dataset X (b) Dataset Y

(c) Coarsified data $\tilde{X}$ (d) Coarsified data $\tilde{Y}$

(e) Graph match on coarse data (f) Hierarchical match result

Fig. 2: Example hierarchical matching on synthetic data



(a) Input Data

(b) Poor Matches Highlighted

Fig. 3: Manifold comparison via graph match

This would give us an idea of the level of redundancy between the datasets. An example of this idea is shown in figure 3. Here we have two synthetic datasets, where the x-axis represents time of observation (and is used to register the modalities), while the y-axis represents the observed value. Looking at the input data (figure 3a, we see that the two modalities contain redundant information for smaller x values. Because of this, the graph match result is very similar to the coregistration indexing for this region. Conversely, the two modalities contain very different information for larger x values. In other words, the manifolds have different local geometries in this area. Therefore the graph match does not compare well against the coregistration. By applying a simple threshold to the observed match quality, we are able to highlight the unique information contained in each modality, and leave the redundant information untouched (figure 3b).

Overall, graphs offer a powerful tool in comparing different modalities. By reducing the input data to a set of similarities, we discard most of the formatting specific to each dataset while still retaining the relevant information. The resulting graph objects can then be compared directly, either through some prior knowledge of registration, or via the geometry of each set. Although this sounds intimidating from a computational standpoint, we have access to powerful approximation tools that make our graph algorithms competitive with the state of the art.

References

[1] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, Sept 2003. I

[2] Dana Lahat, Tülay Adali, and Christian Jutten. Challenges in Multimodal Data Fusion. In *22nd European Signal Processing Conference (EUSIPCO-2014)*, pages 101–105, Lisbonne, Portugal, September 2014. I

[3] Geoffrey Iyer, Jocelyn Chanussot, and Andrea Bertozzi. A graph-based approach for feature extraction and segmentation of multimodal images. 2017. I

[4] Bojan Mohar. The laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications*, 2:871–898, 1991. II

[5] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2), February 2004. II-A

[6] Ekaterina Merkurjev, Cristina Garcia-Cardona, Andrea L. Bertozzi, Arjuna Flenner, and Allon G. Percus. Diffuse interface methods for multiclass segmentation of high-dimensional data. *Applied Mathematics Letters*, 33:29 – 34, 2014. III

[7] Vikraman Arvind, Johannes Köbler, Sebastian Kuhnert, and Yadu Vasudev. *Approximate Graph Isomorphism*, pages 100–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. IV

[8] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, September 1988. IV, IV