A GRAPH-BASED APPROACH FOR SEGMENTATION OF MULTIMODAL DATASETS

Name of author

TODO: Author Affiliations

ABSTRACT

TODO: The whole abstract

***Index Terms*—** TODO: keywords

1. INTRODUCTION

With the increasing availibility of data we often come upon multiple datasets, derived from different sources, that describe the same object or phenomenon. These are called *multimodal datasets*, and a proper treatment of such sets requires more than analyzing each set individually. One common practice is to define maps from each set into a common latent space, then perform any analysis on this common data. (TODO: talk about canonical correlation analysis, and maybe look up parallel factor analysis. I found this in [1].) These methods aim to find and correlate the redundant information between the different sets, however they struggle to identify any information that is exclusive to one dataset. Here we present a graph-based approach that aims to use the unique information found in each dataset to give a more detailed explanation of the source object.

In this paper we consider the case where each dataset contains the same number of elements, and these elements are co-registered (so the i-th point in one set corresponds to the i-th point in another). This situation is often found in image processing, where the sets may be images obtained from different sources (as is the case in our experimental data), or taken at a different time. Call the sets, $X^1, X^2, \ldots, X^m$, with dimensions $d_1, d_2, \ldots, d_m$. Let $X = (X^1, X^2, \ldots, X^m) \subset \mathbb{R}^{n\times(d_1+\cdots+d_2)}$ be the concatenated dataset. Our method creates a latent space Y, along with a map $X \to Y$. We then use k-means to segment the data in Y apply this segmentation to X. In section 2 we give the general theory behind our method, and in 3 we show the results of the method applied to an Optical/LIDAR dataset.

2. THE METHOD

2.1. Graph Laplacian

We approach this problem via graph-based methods. A more detailed survey of the theory can be found in [2]. Here we state only the results necessary to implement our algorithm.

2.1.1. The Graph Min-Cut Problem

We represent our dataset X using an undirected graph $G = (V, E)$. The nodes $v_i \in V$ of the graph correspond to elements of X, and we give each edge e_{ij} a *weight* $w_{ij} \geq 0$ representing the similarity between nodes v_i, v_j, 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*)

$$W = (w_{ij})_{i,j=1}^{n} .$$

Since G is undirected, we require that $w_{ij} = w_{ji}$, which implies that W is a symmetric matrix. There are many different notions of “similarity” in the literature, and each has its own merits. In many applications, one defines

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

where σ is a scaling parameter. In this work we adapt this definition to apply to our multimodal dataset, as is explained in 2.3.

Once the weight matrix has been defined, the data clustering problem can be rephrased as a graph-cut-minimization problem of the similarity matrix W. Given a partition of V into subsets $A_1, A_2, \ldots, A_m$, we define the *ratio graph-cut*

$$\text{RatioCut}(A_1, \ldots, A_m) = \frac{1}{2}\sum_{i=1}^{m} \frac{W(A_i, A_i^c)}{|A_i|}.$$

Where

$$W(A, B) = \sum_{i\in A, j\in B} w_{ij},$$

and the $\frac{1}{2}$ is added to account for double-counting each edge. Heuristically, minimizing the ratio cut serves to minimize the connection between distinct A_i, A_j, while still ensuring that each set is of a reasonable size. Without the term $|A_i|$ term, the optimal solution often contains one large set and $m - 1$ small sets.

Solving the graph min-cut problem is equivalent to finding m indicator vectors $f_1, \dots, f_m \in \mathbb{R}^n$ such that

$$f_{m,j} = \begin{cases} 1 & \text{if } x_j \in A_m \\ 0 & \text{else} \end{cases}.$$

It has been shown in [3] that explicitly solving this problem is an $O(|V|^{m^2})$ process. As this is infeasible in most cases, we instead introduce the Graph Laplacian along with an approximation of the minimization problem.

2.1.2. *Graph Laplacian*

After forming the weight matrix W, we define the 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 *Graph Laplacian*

$$L = D - W.$$

For a thorough explanation of the properties of the Graph Laplacian, see [4]. In our work we will use that L is symmetric and positive definite, as well as the following fact (proven in [2]).

Fact 2.1. For a given graph-cut $A_1, \dots, A_m$, define the $f_1, \dots, f_m$ as above, and have $h_j = f_j / \|f_j\|$. Let H be the $n \times m$ matrix whose columns are h_j. Then $H^T H = I$, and

$$\text{RatioCut}(A_1, \dots, A_m) = \text{Tr}\left(H^T L H\right).$$

As explained in 2.1.1, we cannot solve this problem explicitely, so instead we relax the problem to allow entries of H to take on arbitrary real values. That is, we find

$$\text{argmin}_{H \in \mathbb{R}^{n\times m}} \text{Tr}\left(H^T L H\right) \quad \text{where } H^T H = I.$$

As L is symmetric and H is orthogonal, this problem is solved by choosing H to be the matrix containing the m eigenvectors of L corresponding to the m smallest eigenvalues.

Using the eigenvectors H we define a map $X \to \mathbb{R}^m$. For each graph node $x_i \in X$ we get a vector $y_i \in \mathbb{R}^m$ given by the ith row of H. These y_i give the solution to the relaxed min-cut problem. We then get an approximate solution to the original min-cut problem by using any data clustering algorithm on the y_i. In section 3 we use k-means to segment the y_i, resulting in a well-known algorithm called *spectral clustering*.

2.2. Nyström 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. See [5], [6] for a more complete discussion of this method.

Let X denote the set of nodes of the complete weighted graph. We randomly 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

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

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

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

It is in fact 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 $|X| \times |X|$.

In practice, $|A|$ can be chosen to be quite small without noticeably affecting performence. In Section 3 we use $|A| = n^{\frac{1}{4}}$, and choosing a larger set A gives only a small change in error of approximation.
NOTE: Could expand this section to further explain the approximation (it is an orthogonal projection of W_{BB} onto W_{AB}), and how the eigenvectors are actually derived. This can all be found in [7]. If there is space I'll add it.

2.3. Edge Weights

To calculate the weight matrix W, we first scale our sets X^s, X^t to make distances in each set comparable. Let $X = (X^s, X^t) \subset \mathbb{R}^{n\times(d_1+d_2)}$ be the concatenated dataset, and let $A \subset X$ be the collection of landmark nodes as in 2.2. For simplicicty of notation, rearrange the entries of X so that $A = \{x_1, \dots, x_m\}$. So $|A| = m$, and $m \ll n$. Then for $k = s, t$ define the scaling factor

$$\lambda_k = \text{stdev}\left(\left\|x_i^k - x_j^k\right\| \;;\; 1 \le i \le n,\; 1 \le j \le m\right)$$

For a graph node $x \in X$, we define

$$\|x\| = \max\left(\frac{\|x^s\|}{\lambda_1}, \frac{\|x^t\|}{\lambda_2}\right).$$

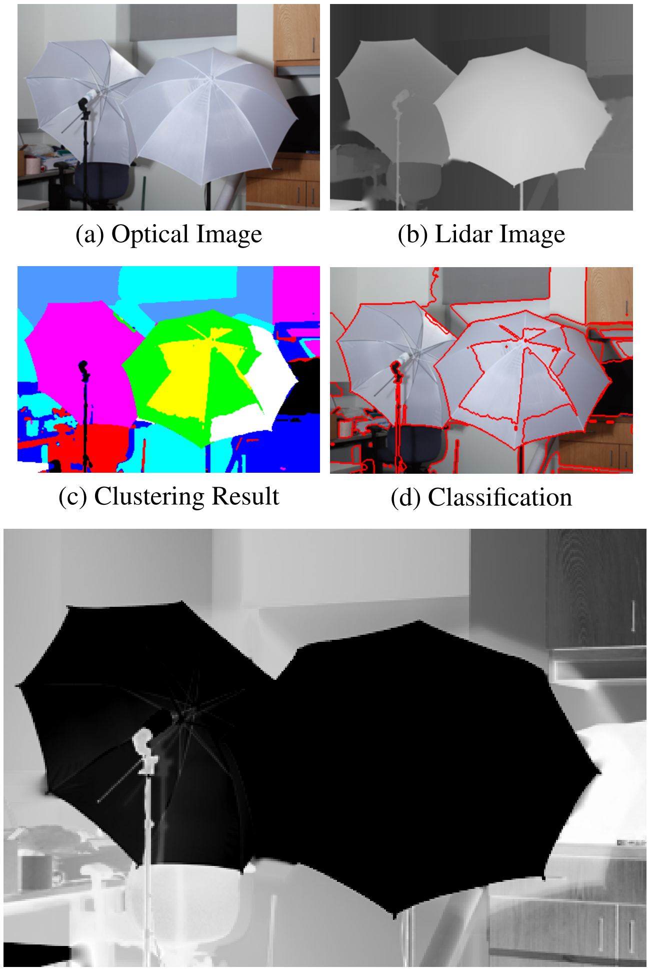
Then define the weight matrix W (using the Nyström Extension), by

$$W = \begin{pmatrix} W_{AA} \\ W_{AB} \end{pmatrix} = (w_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$$

with $w_{ij} = \exp\left(- \|x_i - x_j\|\right)$.

Note that the $\|\cdot\|$ defined above is a norm on the concatenated dataset X. The choice of this particular norm serves to emphasize the information that is unique to the individual datasets. If nodes x_i, x_j are considered similar in X^s, but different in X^t, then they will be treated as different using this norm.

3. EXPERIMENT


(a) Optical Image

(b) Lidar Image

(c) Clustering Result

(d) Classification

(e) First Eigenvector of GL

Fig. 1. Experimental results.

TODO: Talk about the dataset a little. RGB and LIDAR. Number of pixels in the dataset (to give an idea of runtime of the algorithm). In particular give a reference to where we found it.

Method	**Error**	**Time**
Our norm on concatenated set	2.0803e-3	9.2s
2-norm on concatenated set	2.2073e-3	9.4s
Segment images separately	2.4676e-3	17.6s

Table 1. Experimental Results

The results of our algorithm are pictured in 1. The scene captured is or two umbrellas in an indoor setting. We choose this particular scene because of the large amount of non-redundancy between the two datasets. The two umbrellas and the background wall are nearly the same shade of white, and so are difficute to tell apart using only the optical image. Adding the lidar image allows the algorithm to differentiate between these objects.

In figure 1e we show the first eigenvector of the Graph Laplacian, as explained in 2.1.2. This vector corresponds to an approximate segmentation of the image into 2 sets. One can see that the two umbrellas are considered very different from the background (as we see from the lidar data). In addition, the black umbrella-stand on the left-side umbrella is also considered to be different from the two umbrellas (as we see in the optical data). This shows that the eigenvectors of the Graph Laplacian successfully fuse our two datasets.

In 1c and d we show a segmentation of the original image into 8 classes, using a total of 30 eigenvectors (not pictured here).

For a given segmentation of an image, computing the graph-cut error as described in 2.1.1 is an $O(n^2)$ calculation, and requires the full weight matrix W. To avoid this, we instead measure the error of segmentation by calculating the variance of the concatenated data set $X = (X^s, X^t) \subset \mathbb{R}^{n \times (d_1 + d_2)}$ within each class. More explicitly, we use the metric

$$\text{Error} = \frac{\sum_{\text{classes } \mathcal{C}} \text{stdev}(x \in \mathcal{C})}{n \cdot \sqrt{d_1 + d_2}}.$$

To test our method, we compare against a few other common methods. The results are given in Table 3. The concatenation method uses the 2-norm on the concatenated set X, with the scaling as defined in 2.3. I.e,

$$\|x\| = \sqrt{\lambda_1 \|x^s\|^2 + \lambda_2 \|x^t\|^2}.$$

The intersection method computes the full classification via spectral clustering on X^s, X^t separately, then combines the results.

TODO: Some sort of qualitative discussion of the results

4. CONCLUSIONS

TODO: This whole section

5. REFERENCES

[1] Dana Lahat, Tülay Adalı, and Christian Jutten, “Multimodal Data Fusion: An Overview of Methods, Challenges and Prospects,” *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, Aug. 2015. 1

[2] Ulrike von Luxburg, “A tutorial on spectral clustering,” *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. 2.1, 2.1.2

[3] Olivier Goldschmidt and Dorit S. Hochbaum, “A polynomial algorithm for the k-cut problem for fixed k,” *Mathematics of Operations Research*, vol. 19, no. 1, pp. 24–37, 1994. 2.1.1

[4] Bojan Mohar, “The laplacian spectrum of graphs,” *Graph Theory, Combinatorics, and Applications*, vol. 2, pp. 871–898, 1991. 2.1.2

[5] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik, “Spectral grouping using the nystrom method,” *IEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, February 2004. 2.2

[6] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi, ,” *SIAM Journal on Imaging Sciences*, vol. 6, pp. 1903–1930, October 2013. 2.2

[7] J. T. Woodworth, G. O. Mohler, A. L. Bertozzi, and P. J. Brantingham, “Non-local crime density estimation incorporating housing information,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2028, 2014. 2.2