A GRAPH-BASED APPROACH FOR FEATURE EXTRACTION AND SEGMENTATION OF MULTIMODAL IMAGES

Geoffrey Iyer1,2, Jocelynn Chanussot1,2, and Andrea Bertozzi1
1. University of California, Los Angeles
2. Univ. Grenoble Alpes, GIPSA-lab

Introduction
With the increasing availability of data we often come upon problems that collect data from more than once source, or modality. To properly handle these problems, we need to compare data across the different modalities. The step presents a lot of difficulty, as it requires some understanding of the format of the data [5]. Our goal is to create a general algorithm for processing multiple modalities simultaneously.

In this paper, we assume our datasets are co-registered (each modality contains the same number of points, and they share a common indexing), as is often the case in image processing problems. Our method compares graph representations of each modality, extracts features in the form of eigenvectors of the graph Laplacian, then applies standard data-segmentation algorithms on these features to obtain a final classification.

1 Feature Extraction

With the max norm, two nodes are considered similar only if they are similar in every individual modality. Heuristically, this emphasizes the unique information that each dataset brings.

The Graph Laplacian (GL) and Eigendecomposition
Once we have created the weights, we define the normalized graph Laplacian (GL) [9]

\[ L_{sym} = D^{-1/2} LD^{-1/2}, \]

where \( D \) is the diagonal matrix consisting of degrees of nodes. We use the graph Laplacian in our energy minimizations below.

We use the GL to solve the graph NCut problem. Given a partition of our dataset \( A \), we define

\[ NCut(A_1, \ldots, A_m) = \frac{1}{2} \sum_{i\neq j} W(A_i, A_j) \square \rho(A_i, A_j). \]

Minimizing the NCUT separates disimilar nodes (the \( W(A_i, A_j) \) term) and groups similar nodes (the \( \rho(A_i, A_j) \) term). Solving the min-cut problem is equivalent to finding an \( n \times m \) indicator matrix \( u \), where \( u_{ij} = 1 \) if \( x_i \in A_j \), and \( u_{ij} = 0 \) otherwise. Note that

Minimizing this energy is computationally infeasible [4]. We relax the problem, allowing \( u \) to be an orthogonal matrix. We find

\[ \min_{u \in \mathbb{R}^{n \times m}} - Tr(u^T L_{sym} u) \]

where \( u^T u = I \).

This problem is solved by choosing the columns of \( u \) to be the eigenvectors of \( L_{sym} \), corresponding to the \( m \) smallest eigenvalues. Eigenvectors are features extracted from the original dataset \( X \).

Example: Data Fusion Contest 2015 [1]

Graph Representation

- Datasets \( X_1, X_2, \ldots, X_k \), with \( |X_i| = \cdots = |X_k| = m \).
- \( E_i = \text{distance matrix}, e_{ij} = \| x_i - x_j \| \).
- \( A_i = \text{indexing} (E_i) \), scaling factor.
- \( X = \{ X_1, X_2, \ldots, X_k \} \subseteq \mathbb{R}^{n \times (d_1 + \cdots + d_k)} \)
- \( W = \text{similarity matrix} \), \( w_{ij} = \text{similarity between } x_i \text{ and } x_j \), \( w_{ij} = \exp(-\max(e_{ij}/\lambda_0, 1 \leq i \leq k)) \).

Segmentation

We apply two different segmentation algorithms to these features. The first is Spectral Clustering, in which we directly apply k-means to the feature vectors. The second is Graph MBO, explained below.

Graph MBO
Here we minimize a Ginzburg-Landau energy with a semisupervised term [3, 7, 8]. Have \( u \) an \( n \times m \) assignment matrix with

\[ u_{ij} \geq 0 \forall i, j, \sum_{i} u_{ij} = 1. \]

The final output of the algorithm will be a matrix \( u \) where each value is either \( 0 \) or \( 1 \). The energy we minimize is

\[ E(u) = e \cdot \text{Tr}(u^T L_{sym} u) + \frac{1}{2} \sum_{i} u_{ij} W(u_{ij}) \]

where \( e \) is a standard basis vector. The last term includes the fidelity, where \( e \) represents the semisupervised input.

\[ \lambda(x_i) = \begin{cases} 1 & \text{if } x_i \text{ is part of fidelity input} \\ 0 & \text{else} \end{cases} \]

We minimize this via iterative diffusion and thresholding. If \( o^t \) represents the \( n \)-th iterate, then to calculate \( u^{t+1} \) we first diffuse

\[ o^{t+1} = \frac{o^t}{d} - L_{sym} o^t = \mu(x) o^t - u, \]

Then threshold each row

\[ o^{t+1} = \mu \text{ where } \mu = \text{argmax} w_{ij} o_{ij}^t. \]

The diffusion calculation can be done very efficiently by using the eigendecomposition of \( L_{sym} \) (the feature vectors described in (5)). If we change coordinates to the eigenbasis, then the diffusion step reduces to solving for coefficients

\[ a_{ij}^{t+1} = (1 - d - \lambda_k) a_{ij}^t - d_k \cdot d_k^t, \]

where \( \lambda_k \) is the \( k \)-th eigenvalue of \( L_{sym} \), in ascending order.

Results: Umbrella Data [11]

Results: Jade Plant Data [11]

Acknowledgments
This work was supported by NSF grant DMS-1118971, ONR grant N00014-16-1-2119, NSF grant DMS-1417674, European Research Council (Grant no. 320684 - CHESS project), and CNRS (Grant no. PICS-USA 263484).
References


