Multi-Objective Multi-View Spectral Clustering via Pareto Optimization

Xiang Wang* Buyue Qian* Jieping Ye† Ian Davidson*

Abstract

Traditionally, spectral clustering is limited to a single objective: finding the normalized min-cut of a single graph. However, many real-world datasets, such as scientific data (fMRI scans of different individuals), social data (different types of connections between people), web data (multi-type data), are generated from multiple heterogeneous sources. How to optimally combine knowledge from multiple sources to improve spectral clustering remains a developing area. Previous work on multi-view clustering formulated the problem as a single objective function to optimize, typically by combining the views under a compatibility assumption and requiring the users to decide the importance of each view *a priori*. In this work, we propose a multi-objective formulation and show how to solve it using Pareto optimization. The Pareto frontier captures all possible good cuts without requiring the users to set the "correct" parameter. The effectiveness of our approach is justified by both theoretical analysis and empirical results. We also demonstrate a novel application of our approach: resting-state fMRI analysis.

1 Introduction

Traditional spectral clustering only applies to a single graph/view [19, 22]. However, in a wide range of applications, the same dataset can be simultaneously characterized by multiple graphs, which are often constructed from heterogeneous sources. The most common setting, multi-view spectral clustering, is an extension of spectral clustering to multi-view datasets and it is still a developing area.

Previous work on multi-view spectral clustering typically combines different views so as a *single objective* function is optimized. This inherently makes the assumption that the different views are *compatible* to each other [6, 8, 18]. Previous work also required the users to set a parameter that regularizes the combination and thus implicitly decides the outcome of the algorithm.

In this paper, we explore an alternative and more natural formulation that treats the problem as a *multi-objective* problem. Given two views, we create a bi-criteria objective function (see Equation (2.1)) that simultaneously considers the quality of **a single cut** on both graphs. This cut can be viewed as a tradeoff between the two views/objectives. To solve the problem, we use the classic Pareto optimization framework, which allows multiple objectives to compete with each other in deciding the optimal tradeoffs.

Our multi-objective spectral clustering formulation has several benefits and makes the following contributions to the field:

- We solve the multi-objective problem using Pareto optimization. The Pareto frontier captures all possible good cuts that are preferred by one or more objectives. (Section 3)
- We present a novel algorithm that reduces the search space from an infinite number of possible cuts (since a cut in the relaxed sense is just a real vector) to a small set of mutually orthogonal cuts so that the Pareto frontier can be computed efficiently. (Section 3.1)
- We provide an approximation bound on how good the solution in the reduced space is. The bound states how much better an optimal solution in the full search space can be than the one in the reduced search space. (Section 3.3)
- The Pareto optimal cuts can be interpreted either individually as alternative clusterings or collectively as a Pareto embedding of the dataset. (Section 3.2)
- The effectiveness of our approach is evaluated on benchmark datasets with comparison to the state-of-the-art multi-view spectral clustering techniques (Section 4). We also demonstrate a novel application of our algorithm for resting-state fMRI analysis, where one graph represents the ground truth and the other the observed data. (Section 5)

Related work To our knowledge no work exists on multi-objective spectral clustering with the closest work being multi-view clustering. Previous work on multi-view (spectral) clustering relies on a fundamental assumption that all the views are *compatible* to each other,

*Department of Computer Science, University of California, Davis. Email: xiang@ucdavis.edu; byqian@ucdavis.edu; davidson@cs.ucdavis.edu.

†Computer Science and Engineering, Arizona State University. Email: jieping.ye@asu.edu.

Table 1: Table of notations

Symbol	Meaning
N	The number of instances/nodes
D	The degree matrix
$\bar{L}$	The normalized graph Laplacian
$\mathbf{v}$	The normalized relaxed indicator vector
Ω	The set of all nontrivial cuts
P	The set of Pareto optimal cuts
$\mathcal{J}(\cdot,\cdot)$	The joint numerical range of two graphs
$\mathcal{F}(\cdot,\cdot)$	The Pareto frontier of $\mathcal{J}(\cdot,\cdot)$

i.e. different views are generated from the same underlying distribution [5], or different views agree on a consensus partition that reflects a hidden ground truth [17]. This assumption is then exploited to convert multi-view spectral clustering into a single objective problem, which either tries to maximize the agreement between the partitions generated by different views [15,16,21], or combines multiple views into one view with the anticipation that the combined view is a better representation of the underlying distribution [10, 20, 23]. In contrast, our multi-objective formulation allows the two graphs to be incompatible and compete with each other based on their own preferences. The most preferred cuts will be captured by the Pareto frontier, which represents a range of alternative yet optimal ways to partition the dataset. Pareto optimization is popular in many computer science areas (see [14] for a review), since it provides a principled way of optimizing tradeoffs between competing objectives.

2 A Pareto Optimization Framework for Multi-View Spectral Clustering

In this section, we propose our multi-objective formulation for spectral clustering and show how to solve it in the context of Pareto optimization. We follow the standard formulation and notations of spectral clustering [19, 22] (see Table 1).We start with the two-view case, then later discuss its extension to more than two views (Section 3.4).

2.1 A Multi-Objective Formulation A two-view dataset can be represented by two graphs that share the same set of nodes but have two different sets of edges, namely $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2)$. Our goal is to find a shared cut that simultaneously cuts both graphs with minimal cost. This leads us to a natural extension of spectral clustering, where instead of finding the normalized min-cut on one graph, we find the normalized min-cut over the two graphs simultaneously:

$$\operatorname*{argmin}_{\mathbf{v}\in\Omega} \{\mathbf{v}^T\bar{L}_1\mathbf{v}, \mathbf{v}^T\bar{L}_2\mathbf{v}\}, \tag{2.1}$$

where

$$\Omega \triangleq \{\mathbf{v}\in\mathbb{R}^N \mid \mathbf{v}^T\mathbf{v}=1, \mathbf{v}\perp_{\bar{L}_1} D_1^{1/2}\mathbf{1}, \mathbf{v}\perp_{\bar{L}_2} D_2^{1/2}\mathbf{1}\} \tag{2.2}$$

is the set of all nontrivial cuts. The notation $\mathbf{v}\perp_X \mathbf{v}'$ means $\mathbf{v}^T X\mathbf{v}' = 0$. $\mathbf{v}\in\Omega$ means that $\mathbf{v}$ is normalized and it is orthogonal to (w.r.t. $\bar{L}_1$ and $\bar{L}_2$) the trivial cut $\mathbf{1}$ (the $N\times 1$ vector of all 1's).

Note that Equation (2.1) can be reduced to spectral clustering if we replace $\bar{L}_2$ with $\bar{L}$ and $\bar{L}_1$ with the identity matrix I. In other words, spectral clustering on a single graph is covered as a special case of our model where one graph is combined with a zero-knowledge graph (whose normalized graph Laplacian is I).

2.2 Joint Numerical Range and Pareto Optimality Rather than converting the two objectives in Equation (2.1) to a single objective, we solve them simultaneously using Pareto optimization. Since we aim to find a single cut for both graphs, we can consider finding this cut as a competition between the two graphs: each graph gives the cut a "score" (the cut quality); we enumerate all possible cuts by their costs on the respective graphs, which constitute the *joint numerical range* [12] of the two graphs. Each point in the joint numerical range represents a tradeoff between the two graphs in terms of cut cost. Next we compute the *Pareto frontier* of the joint numerical range, which corresponds to the cuts that are optimal in terms of *Pareto improvement*: its cost on one graph cannot be improved (decrease) without making the cost on the other graph worse (increase).

The joint numerical range of $\mathcal{G}_1$ and $\mathcal{G}_2$ is defined as follows:

$$\mathcal{J}(\mathcal{G}_1,\mathcal{G}_2) \triangleq \{(\mathbf{v}^T\bar{L}_1\mathbf{v},\ \mathbf{v}^T\bar{L}_2\mathbf{v}) \mid \mathbf{v}\in\Omega\}, \tag{2.3}$$

where Ω is defined as in Equation (2.2). Essentially, $\mathcal{J}(\mathcal{G}_1,\mathcal{G}_2)$ is the set of the costs of all nontrivial cuts over $\mathcal{G}_1$ and $\mathcal{G}_2$.

Recall that in spectral clustering we evaluate the quality of any two cuts by comparing their costs on a single graph. We say $\mathbf{v}$ is better than $\mathbf{v}'$ if $\mathbf{v}$ has a lower cost than $\mathbf{v}'$ does. Now consider the joint numerical range of two graphs. When we evaluate the quality of a cut, we must consider its cost on both graphs. Specifically, we need to introduce the notion of Pareto improvement:

Definition 1. ***(Pareto Improvement)*** *Given two different cuts $\mathbf{v}\in\Omega$ and $\mathbf{v}'\in\Omega$ over two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we say $\mathbf{v}$ is a Pareto improvement over $\mathbf{v}'$ if and only if one of the following two conditions holds:*

$$\mathbf{v}^T\bar{L}_1\mathbf{v} < \mathbf{v}'^T\bar{L}_1\mathbf{v}' \wedge \mathbf{v}^T\bar{L}_2\mathbf{v} \le \mathbf{v}'^T\bar{L}_2\mathbf{v}',$$

or

$$\mathbf{v}^T \bar{L}_1 \mathbf{v} \leq \mathbf{v}'^T \bar{L}_1 \mathbf{v}' \wedge \mathbf{v}^T \bar{L}_2 \mathbf{v} < \mathbf{v}'^T \bar{L}_2 \mathbf{v}'.$$

When $\mathbf{v}$ *is a Pareto improvement over* $\mathbf{v}'$*, we say* $\mathbf{v}$ ***dominates*** $\mathbf{v}'$*, or* $\mathbf{v}'$ ***is dominated by*** $\mathbf{v}$*, and we use the following notation:*

$$\mathbf{v} \prec_{(\mathcal{G}_1,\mathcal{G}_2)} \mathbf{v}'.$$

In terms of Pareto improvement, the optimal solution to Equation (2.1) is the Pareto frontier of $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$.

DEFINITION 2. *(**Pareto Frontier**) Define:*

$$\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2) \triangleq \{(\mathbf{v}^T \bar{L}_1 \mathbf{v},\ \mathbf{v}^T \bar{L}_2 \mathbf{v}) \mid \mathbf{v} \in P\}.$$

$\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$ is the Pareto frontier of $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ if P satisfies:

1. *$P \subset \Omega$;*
2. *(Optimality) $\forall \mathbf{v} \in P$, $\neg\exists \mathbf{v}' \in \Omega$, such that $\mathbf{v}' \prec_{(\mathcal{G}_1,\mathcal{G}_2)} \mathbf{v}$;*
3. *(Completeness) $\forall \mathbf{v} \in \Omega \backslash P$, $\exists \mathbf{v}' \in P$, such that $\mathbf{v}' \prec_{(\mathcal{G}_1,\mathcal{G}_2)} \mathbf{v}$.*

We say $\mathbf{v}$ lies on the Pareto frontier of $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ if $\mathbf{v} \in P$.

We call P the set of Pareto optimal cuts. Intuitively speaking, P satisfies the following properties: 1) any cut in P is better than cuts that are not in P (completeness); 2) any two cuts in P are equally good; 3) for any cut in P, it is impossible to reduce its cost on one graph without increasing its cost on the other graph (optimality). Therefore, our Pareto optimization framework captures the complete set of equally good cuts (in terms of Pareto optimality) that are superior to any other possible cuts.

We summarize our approach as follows:

1. Given the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, construct their joint numerical range $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$.
2. Compute the Pareto frontier of $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$, which is $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$.
3. Output P, the set of Pareto optimal cuts.

3 Algorithm Derivation

In this section, we present an efficient approximation algorithm to compute the Pareto frontier. We also discuss how to interpret the Pareto optimal cuts and convert them into actual clusterings in practice. Our algorithm is summarized in Algorithm 1.

Algorithm 1 Multi-Objective Multi-View Spectral Clustering via Pareto Optimization

Input: Two graph Laplacians $\bar{L}_1, \bar{L}_2$
Output: The set of Pareto optimal cuts $\tilde{P}$

```
Solve the generalized eigenvalue problem: L̄1 v = λ L̄2 v;
Normalize all v's such that v^T v = 1;
Let P̃ be the set of all eigenvectors, excluding the two associated with eigenvalue 0 and ∞;
for all v ∈ P̃ do
    for all v' ∈ P̃, v' ≠ v do
        if v ≺_(G1,G2) v' then
            Remove v' from P̃;
            continue;
        end if
        if v' ≺_(G1,G2) v then
            Remove v from P̃;
            break;
        end if
    end for
end for
(Optional) Consolidate the cuts in P̃ into a single clustering u (see Section 3.2);
```

3.1 Computing the Pareto Frontier via Generalized Eigendecomposition

Recall that $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2) \subset \mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ is the Pareto frontier and $P \subset \Omega$ is the set of Pareto optimal cuts. Our goal is to compute $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$, or equivalently P. However, Ω consists of an infinite number of different cuts (in the relaxed form)[1], which map to an infinite number of points in $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$. To the best of our knowledge, there is no efficient way to compute $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$ in closed form.

Nevertheless, although Ω consists of an infinite number of cuts, many of those cuts are effectively identical to each other. For instance, one cut may only differ from another cut by a small perturbation. From a practical point of view, those two cuts will lead to exactly the same clustering. Therefore, we introduce an additional constraint to narrow down our search space: we only focus on a subset of cuts that are distinct to each other, namely they must be mutually orthogonal. Consequently, instead of dealing with a continuous vector space Ω, we only consider the set of vectors, $\tilde{\Omega}$, which comprise an orthogonal basis of Ω.

Formally, we define

$$\tilde{\Omega} \triangleq \{\mathbf{v} \in \Omega \mid \forall \mathbf{v} \neq \mathbf{v}', \mathbf{v} \perp_{\bar{L}_1} \mathbf{v}', \mathbf{v} \perp_{\bar{L}_2} \mathbf{v}'\},$$

$$\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2) \triangleq \{(\mathbf{v}^T \bar{L}_1 \mathbf{v},\ \mathbf{v}^T \bar{L}_2 \mathbf{v}) \mid \mathbf{v} \in \tilde{\Omega}\}.$$

[1] The infinite amount of real vectors map to 2^{N-1} distinct clusterings, which are still too many to enumerate through.

Under a mild assumption that the null space of $\bar{L}_1$ and $\bar{L}_2$ do not overlap, $(\bar{L}_1, \bar{L}_2)$ is a Hermitian definite matrix pencil [3]. Then $\tilde{\Omega}$ is the set of N (N is the number of nodes) eigenvectors of the generalized eigenvalue problem [11]

$$\bar{L}_1\mathbf{v} = \lambda\bar{L}_2\mathbf{v} \tag{3.4}$$

less the principal eigenvector of $\bar{L}_1$, which is $D_1^{1/2}\mathbf{1}$, and the principal eigenvector of $\bar{L}_2$, which is $D_2^{1/2}\mathbf{1}$. The generalized eigenvalue problem in Equation (3.4) can be solved efficiently in closed form.

Now since $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$ only consists of $N-2$ points, corresponding to the $N-2$ mutually orthogonal cuts in $\tilde{\Omega}$, we can efficiently find its Pareto frontier (see Algorithm 1), which is:

$$\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2) \triangleq \{(\mathbf{v}^T\bar{L}_1\mathbf{v},\ \mathbf{v}^T\bar{L}_2\mathbf{v}) \mid \mathbf{v} \in \tilde{P}\}.$$

$\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2)$ is an approximation to $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$. We call $\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2)$ the orthogonal Pareto frontier and $\tilde{P}$ the orthogonal Pareto optimal cuts. We will provide a bound for this approximation in Section 3.3.

The runtime of our algorithm is dominated by that of generalized eigendecomposition, which is on par with that of spectral clustering in big-O notation.

Example We use the Wine dataset from the UCI archive to demonstrate how our algorithm works. It consists of 119 instances. Each instance has 13 features (attributes). We construct one view using the first 6 features and the other view using the remaining 7 features. After applying our approximation algorithm, we have 117 points in $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$ that correspond to 117 nontrivial orthogonal cuts of the graph, as shown in Figure 1 (+'s). Among the 117 cuts, three lie on the Pareto frontier $\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2)$ (the circled points). We visualize the clusterings derived from the three Pareto optimal cuts in Figure 2. Note that Cut 3 (Figure 2(c)) coincides with the ground truth labeling of the Wine dataset (Figure 2(d)).

3.2 Interpreting and Using the Pareto Optimal Cuts

The Pareto optimal cuts in $\tilde{P}$ can be interpreted either individually as alternative clusterings or collectively as a Pareto embedding of the dataset.

Specifically, if the two views are compatible with each other, then by definition, they would agree on a single cut that is Pareto optimal. In this case, our algorithm will produce a unique clustering that is optimal. If the two views are incompatible (which is the case for the Wine dataset in Figure 1), the cardinality of $\tilde{P}$ will be greater than 1. In this case, the Pareto optimal cuts can be interpreted as a set of alternative clusterings. On the one hand, these cuts are alternative to each other in terms of orthogonality. On the other hand, as shown in Figure 2, different Pareto optimal cuts correspond to different ways to partition the dataset: Figure 2(a) separates three outliers from the rest of the data points, Figure 2(b) partitions the points vertically, and Figure 2(c) partitions the points horizontally. These three alternative clusterings are all informative and could all be valid, depending on the users' needs.

In practice, $|\tilde{P}|$ is usually small. Hence it is feasible to submit $\tilde{P}$ directly to domain experts for further review. We argue that it is more intuitive and much easier for domain experts to choose among a few plausible clusterings than assigning a parameter *a priori* which only implicitly decides the outcome of the algorithm.

Sometimes the application demands one single partition as output. In this case, we can interpret the Pareto optimal cuts in $\tilde{P}$ collectively using the classic spectral embedding technique [2, 4]. Specifically, let V be a $N \times |\tilde{P}|$ matrix, whose columns are the Pareto optimal cuts in $\tilde{P}$. If we look at the i-th rows of V, it can be considered as an embedding of the i-th node of the graph in a $|\tilde{P}|$-dimensional subspace, spanned by the mutually orthogonal generalized eigenvectors (Figure 2 is the Pareto embedding of the Wine dataset). To derive a single clustering, we perform K-means on the Pareto embedding of all nodes, which is also common practice.

In addition, we used in our experiments a simple but effective unsupervised weighting scheme that can further improve the result. We assigned each Pareto optimal cut a weight that is inversely proportional to the squared sum of its costs on respective graphs. In other words, all cuts being Pareto optimal, we assign higher weights to those with lower overall costs.

3.3 Approximation Bound for Our Algorithm

In our algorithm, we compute the *orthogonal* Pareto frontier $\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2)$ as an approximation to the Pareto frontier $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$. Here we create an upper bound on how far a point in the Pareto frontier can be to the *orthogonal* Pareto frontier. This effectively bounds the difference between the costs of the cuts on the Pareto frontier and those on the orthogonal Pareto frontier.

Let $\mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$ be the convex hull of $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$. It is a convex polygon that lies in $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ (see Figure 1). Let $\mathbf{ext}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$ be its extreme points ("corners" of the convex polygon). Let

$$B \triangleq \mathbf{ext}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)) \cap \tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2).$$

B is nonempty (e.g. the leftmost and lowest points in $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$ are both in B). First, it is obvious that

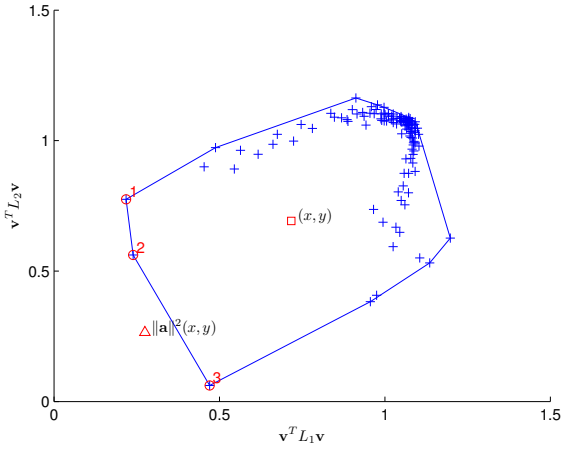
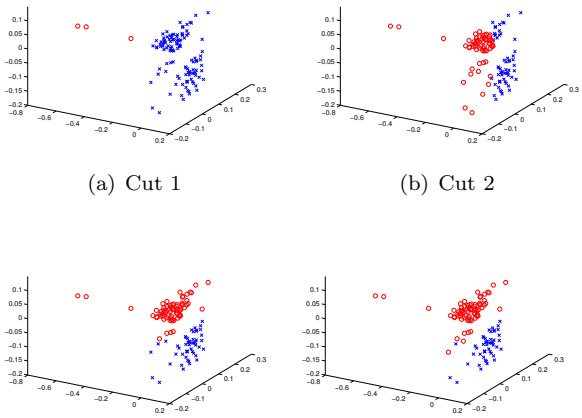


Figure 1: The joint numerical range of the Wine dataset. The +'s correspond to points in $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$. The ∘'s are the Pareto optimal cuts found by our algorithm, which is $\tilde{\mathcal{F}}(\mathcal{G}_1, \mathcal{G}_2)$.



(a) Cut 1 (b) Cut 2

(c) Cut 3 (d) Ground truth label

Figure 2: The Pareto embedding of the Wine dataset. (a)(b)(c) show the clusterings derived from the Pareto optimal cuts in Figure 1; (d) shows the original labels of the dataset.

any points in $\mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$ cannot dominate points in B. Then, we examine the chance that any points in $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ dominate points in B.

Let $\tilde{\Omega} = \{\tilde{\mathbf{v}}_i\}_{i=1}^{N-2}$ and $\tilde{V} = (\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_{N-2})$. Any $\mathbf{v} \in \Omega$ can be represented by a linear combination of $\tilde{\mathbf{v}}_i$'s: $\mathbf{v} = \tilde{V}\mathbf{a}$, $\mathbf{a} = (a_1, \ldots, a_{N-2})^T$. We define $f(\mathbf{v}) : \Omega \mapsto \mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$:

(3.5)
$$f(\mathbf{v}) = \big(\mathbf{v}^T \bar{L}_1 \mathbf{v},\ \mathbf{v}^T \bar{L}_2 \mathbf{v}\big)$$

(3.6)
$$= \big((\sum_{i=1}^{N-2} a_i \tilde{\mathbf{v}}_i)^T \bar{L}_1 (\sum_{j=1}^{N-2} a_j \tilde{\mathbf{v}}_j), (\sum_{i=1}^{N-2} a_i \tilde{\mathbf{v}}_i)^T \bar{L}_2 (\sum_{j=1}^{N-2} a_j \tilde{\mathbf{v}}_j)\big)$$

(3.7)
$$= \big(\sum_{i=1}^{N-2}\sum_{j=1}^{N-2} a_i a_j \tilde{\mathbf{v}}_i^T \bar{L}_1 \tilde{\mathbf{v}}_j, \sum_{i=1}^{N-2}\sum_{j=1}^{N-2} a_i a_j \tilde{\mathbf{v}}_i^T \bar{L}_2 \tilde{\mathbf{v}}_j\big)$$

(3.8)
$$= \big(\sum_{i=1}^{N-2} a_i^2 \tilde{\mathbf{v}}_i^T \bar{L}_1 \tilde{\mathbf{v}}_i, \sum_{i=1}^{N-2} a_i^2 \tilde{\mathbf{v}}_i^T \bar{L}_2 \tilde{\mathbf{v}}_i\big)$$

(3.9)
$$= \|\mathbf{a}\|^2 \big(\sum_{i=1}^{N-2} \frac{a_i^2}{\|\mathbf{a}\|^2} \tilde{\mathbf{v}}_i^T \bar{L}_1 \tilde{\mathbf{v}}_i, \sum_{i=1}^{N-2} \frac{a_i^2}{\|\mathbf{a}\|^2} \tilde{\mathbf{v}}_i^T \bar{L}_2 \tilde{\mathbf{v}}_i\big)$$

(3.10)
$$= \|\mathbf{a}\|^2 (x, y)$$

$\|\cdot\|$ is 2-norm. The transition from Equation (3.7) to (3.8) is due to the fact that, for $i \neq j$, $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ are mutually orthogonal with respect to $\bar{L}_1$ and $\bar{L}_2$, according to the definition of $\tilde{\Omega}$. Equation (3.10) simply replaces the two items in Equation (3.9) with shorter notation.

Since $\frac{a_i^2}{\|\mathbf{a}\|^2} \geq 0$ and $\sum_{i=1}^{N-2} \frac{a_i^2}{\|\mathbf{a}\|^2} = 1$, (x, y) is a convex combination of points in $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$, therefore $(x, y) \in \mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$. In other words, for any $\mathbf{v} \in \Omega$, $f(\mathbf{v})$ can be represented by a point in $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$ multiplied by a scaling factor $\|\mathbf{a}\|^2$. If $\|\mathbf{a}\|^2 = 1$, then $f(\mathbf{v}) = (x, y) \in \mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$ and it cannot dominate any point in B. If $\|\mathbf{a}\|^2 > 1$, then $f(\mathbf{v})$ is dominated by (x, y), therefore, it cannot dominate any point in B. On the other hand, we can derive a lower-bound for $\|\mathbf{a}\|^2$. We have:

$$1 = \|\mathbf{v}\| = \|\tilde{V}\mathbf{a}\| \leq \|\tilde{V}\|\|\mathbf{a}\| = \sigma_{\max}(\tilde{V})\|\mathbf{a}\|,$$

where $\sigma_{\max}(\tilde{V})$ is the largest singular value of $\tilde{V}$. Consequently we have:

$$\|\mathbf{a}\|^2 \geq 1/\sigma_{\max}^2(\tilde{V}). \tag{3.11}$$

$1/\sigma_{\max}^2(\tilde{V})$ effectively bounds how far $f(\mathbf{v})$ can be from the point (x, y), which is in $\mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$. The larger $1/\sigma_{\max}^2(\tilde{V})$ is, the closer $f(\mathbf{v})$ is to (x, y), thus the better $\mathbf{co}(\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2))$ approximates $\mathcal{J}(\mathcal{G}_1, \mathcal{G}_2)$ and the more likely B coincides with $\mathcal{F}(\mathcal{G}_1, \mathcal{G}_2)$. The equality in Equation (3.11) holds when $\mathbf{v}$ is the largest right singular vector of $\tilde{V}$. As shown in Figure 1, $\triangle$ is

$f(\mathbf{v}) = \|\mathbf{a}\|^2(x, y)$ when $\|\mathbf{a}\|^2$ reaches the lower-bound on the Wine dataset; $\square$ is the corresponding (x, y). Note that $\|\mathbf{a}\|^2 < 1$ is a necessary but not sufficient condition for $f(\mathbf{v})$ to dominate any point in B. For example, in Figure 1, although $f(\mathbf{v})$ lies outside the convex hull of $\tilde{\mathcal{J}}(\mathcal{G}_1, \mathcal{G}_2)$, it does not dominate any point in B, which are the three circled points.

3.4 Extension to Multiple Views It is possible to extend our framework to M views. Given a finite number of cuts across M views, it is not difficult to compute the M-dimensional Pareto frontier. The challenge is to discretize the joint numerical range of M graphs, since the generalized eigenvalue system can only accommodate two graphs at a time. To cope with the limitation, we combine each view with the average of the other $M-1$ views, respectively. Then we use generalized eigendecomposition to compute the orthogonal joint numerical range of those two graphs. We repeat this process M times and will get $M(N-2)$ cuts. Then we compute the Pareto frontier of the $M(N-2)$ cuts. This approach ensures that a good cut will be preserved as long as it is preferred by at least one view.

4 Empirical Study

In this section, we use six UCI benchmark datasets [1] and the 20 Newsgroups dataset[2] to empirically evaluate the effectiveness of our approach. We aim to answer the following questions:

- How does our algorithm perform on datasets with incompatible views? (Table 2, Figure 3).
- How does it perform on datasets with compatible views? (Table 3, Figure 3).
- How does it compare to the single-view spectral clustering baseline and the state-of-the-art multi-view spectral clustering techniques? (Tables 2 and 3, Figure 3).

The short answer to these questions is that (see Figure 3) our technique performs comparably to other multi-view clustering techniques on datasets with compatible views, and it outperforms the other techniques by a large margin on datasets with incompatible views. This is a significant result since testing if two views are compatible or not is an open research problem.

We chose six UCI benchmark datasets, namely `Hepatitis`, `Iris`, `Wine`, `Glass`, `Ionosphere`, and `Breast Cancer`. To construct the two views, we divided the features into two disjoint subsets. We divided them in such a way that the two views tend to be incompatible, e.g. we put different types of features into opposite views. The graph Laplacians were computed using RBF kernel. We also used the 20 Newsgroups dataset that contains documents from four high-level categories: `comp`, `rec`, `sci`, and `talk`. These categories were used as ground truth labels. The features of the dataset are 100 representative words. To construct the two views, we randomly divided the features into two subsets, each with 50 words. Therefore, for this dataset, the two views tend to be compatible. The graph Laplacians were computed using inner-product kernel, based on the word-frequency vectors.

For our algorithm, we first computed the Pareto optimal cuts, then used their Pareto embedding to find a clustering. We evaluated this clustering against ground truth labels using adjusted Rand index [13]: 0 means the partition is as good as a random assignment and 1 means the partition perfectly matches the ground truth.

To make comparisons, we implemented several state-of-the-art multi-view spectral clustering algorithms (which all use a single objective). `MM` is the Markov mixture algorithm proposed in [23], where the two views are combined using a mixing random walk on both graphs. `KerAdd` is kernel addition algorithm that combines the two views by averaging their graph Laplacians. Though simplistic, this method has been shown to be very effective when two views are compatible [9, 15, 23] and it outperforms many more sophisticated alternatives. `CoReg` is the co-regularization multi-view spectral clustering algorithm proposed in [16]. We implemented the centroid based version and used the centroids to compute the final clustering. As a baseline, we also report the results of performing spectral clustering on each single view (`View 1`, `View 2`), as well as the concatenation of two views (`Concat.`).

The results are summarized in Table 2 and 3. Our approach (`Pareto`) outperformed all three spectral clustering baselines (`View 1`, `View 2`, `Concat.`) in most cases. This suggests that our approach is effective in combining the two views in a constructive way. When comparing to existing multi-view clustering techniques, our approach outperformed any single one of them. Across all 12 datasets, our approach achieved highest ARI on 6 and second highest on 3.

More importantly, our approach is more reliable in terms of performance than its competitors when the two views were constructed to be incompatible. Across 6 UCI datasets (Table 2), our approach achieved highest performance on 4 and second highest on the other 2. This justifies the advantage of our multi-objective framework over the single-objective framework used by previous methods. On the other hand, for the 20

[2] `http://cs.nyu.edu/~roweis/data.html`

Table 2: The adjusted Rand index of various algorithms on six UCI datasets with **incompatible** views. Bold numbers are best results. The number in the parenthesis is the performance gain of our approach (Pareto) over the best competitor. Our method performs the best on the majority of datasets.

	View 1	View 2	Concat.	MM	KerAdd	CoReg	Pareto
Hepatitis	-0.109	0.247	0.193	-0.091	-0.111	0.247	**0.360**(+0.113)
Iris	0.136	**0.808**	0.485	0.430	0.430	0.404	**0.808**(+0.000)
Wine	-0.015	0.869	-0.015	0.869	**0.933**	**0.933**	**0.933**(+0.000)
Glass	**0.510**	0.041	0.413	0.474	0.448	**0.510**	0.490(−0.020)
Ionosphere	0.209	-0.043	-0.043	0.209	**0.257**	0.209	0.209(−0.048)
Breast Cancer	0.005	0.005	0.112	0.005	0.002	0.297	**0.368**(+0.071)

Table 3: The adjusted Rand index of various algorithms on the 20 Newsgroups dataset with **compatible** views. Bold numbers are best results. The number in the parenthesis is the performance gain of our approach (Pareto) over the best competitor. Note our method is comparable to other methods. The best performing method here, MM, performs poorly in Table 2.

	View 1	View 2	Concat.	MM	KerAdd	CoReg	Pareto
comp-rec	0.697	0.719	0.747	**0.758**	0.747	0.741	0.747(−0.011)
comp-sci	0.520	0.506	0.700	0.702	**0.717**	0.688	0.684(−0.033)
comp-talk	0.837	0.702	0.939	0.939	0.939	0.939	**0.957**(+0.018)
rec-sci	0.533	0.605	**0.640**	0.633	**0.640**	0.626	**0.640**(+0.000)
rec-talk	0.684	0.681	0.754	**0.764**	0.748	0.748	0.725(−0.039)
sci-talk	-0.011	0.520	0.558	**0.566**	0.559	0.393	0.542(−0.024)

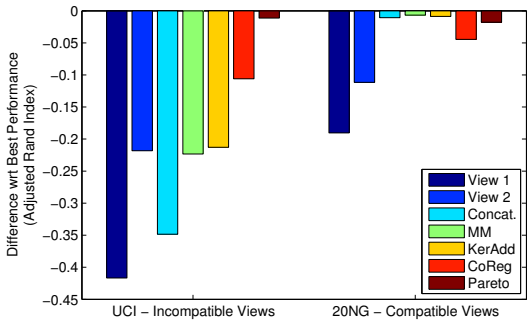


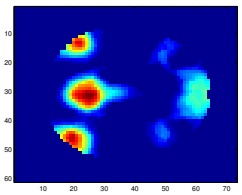
Figure 3: The mean difference (in terms of adjusted Rand index) of various techniques wrt the best-performing technique on each dataset, grouped by two cases (datasets with compatible views vs. datasets with incompatible views).

Newsgroups datasets (Table 3) where the two views are constructed to be compatible, the advantage of our approach was less significant. Nevertheless, it was not outperformed by its competitors by a large margin.

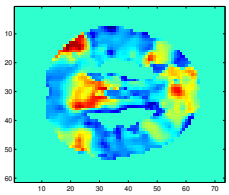
To better demonstrate our approach's consistent performance with both compatible and incompatible views, we compute the relative difference (in terms of adjusted Rand index) of each technique's performance with respect to the best-performance approach per dataset. Then we compute the mean relative difference for each technique on the UCI datasets and the 20 Newsgroups dataset, respectively. Since no technique was always the best, the mean relative difference of all techniques is always less than zero. However, in Figure 3, we can clearly see that our algorithm is the only technique that performed consistently well in both cases (compatible and incompatible). In contrast, although the Concat., MM, and KerAdd performed very well on compatible views, they performed poorly on incompatible views.

5 Applying Our Algorithm to Automated fMRI Analysis

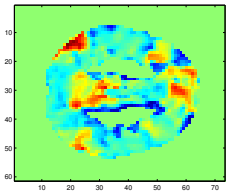
In this section, we explore an application of our work where incompatible views naturally occur: resting-state fMRI analysis. A resting-state fMRI scan is a series of 3D brain images over time of a person at resting state. We can construct a graph for each scan, where each node corresponds to a voxel in the brain image, and the edge weight corresponds to the correlation between the activity of two voxels over time. If we partition this graph into two parts, one will comprise regions in the brain that share the same functionality (called a cognitive network), the other background. For our application, we are interested in a particular network, called the Default Mode Network (DMN) (see Figure 4(a)), which is periodically activated when the



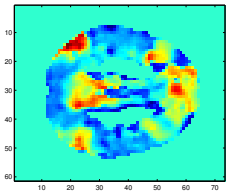
(a) An illustration of idealized DMN



(b) The DMN exhibited in an exemplary scan



(c) The DMN in a target scan, induced by (b)



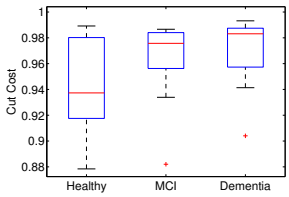
(d) The DMN in another target scan, induced by (b)

Figure 4: The results of applying our algorithm to resting-state fMRI scans. Illustrated is a horizontal slice of the scan (eyes are on the right-hand side). We use an exemplary scan (View 1) to induce the Default Mode Network (the red/yellow pixels in the figures) in a set of target scans (View 2). Our algorithm produced consistent partitions across different target scans.

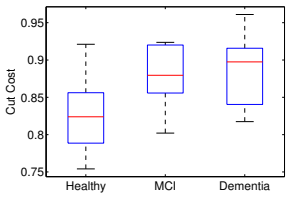
person is in resting state. The absence of the DMN has been related to the Alzheimer's disease [7]. Our goal is to elicit the DMN from a given scan and determine its strength.

The challenge of this task is that fMRI scans are notoriously noisy. Many factors, such as equipment calibration, head positioning, and the mental state of the subject, can introduce a significant amount of noise into the scan. As a result, the same person scanned twice over the period of a month (as our data is) will produce two incompatible scans which suggest two very different clusterings. Combining two incompatible scans is not desirable because the noise in one scan can dominate the other scan. In effort to overcome this, we use our algorithm to simultaneously cut two scans: an exemplary scan and a target scan. The exemplary scan is a scan verified by domain experts that exhibits a strong DMN pattern. We pair this exemplary scan with a target scan, which may or may not be compatible, to detect the DMN therein.

Figure 4(a) shows what a DMN should look like. Note that it only illustrates the general shape of the DMN based on the average of a large number of scans. The actual DMN differs from individual to individual. Figure 4(b) shows the DMN exhibited by an exemplary scan from a young healthy person.



(a) Induced by scan A



(b) Induced by scan B

Figure 5: The costs of induced DMN cuts on the target scans, grouped by 3 sub-populations. The costs increase as the cognitive symptom gets worse.

Given the exemplary scan and a target scan, our algorithm finds the set of Pareto optimal cuts. We compare each Pareto optimal cut to the DMN cut exhibited by the exemplary scan and choose the most similar one (as shown in Figure 4(c) and (d)) as the induced DMN cut for the target scan. **The induced DMN cut can be considered as the target scan's best effort (in terms of Pareto optimality) to accommodate the exemplary DMN cut.** We then record the cost of the induced DMN cut for the target scan, which can be naturally viewed as an indicator for the strength of the DMN in the target scan. The lower the cost is, the more the target scan prefers the DMN cut, thus the stronger the DMN is in the target scan.

The dataset we used was collected and processed within the research program of UC Davis Alzheimer's Disease Center. The exemplary scans were chosen by domain experts from a group of young healthy individuals. The target scans were from 31 elderly individuals: 11 diagnosed as Healthy, 10 Mild Cognitive Impairment (MCI), and 10 Dementia.

We observed that, despite the ubiquitous noise in fMRI scans, our algorithm managed to induce the DMN cut across all target scans, i.e. the candidate set always included a cut that is highly similar to the exemplary DMN in Figure 4(b). In Figure 4(c) and (d), we illustrate two induced DMN cuts from two different target scans. This demonstrated that our formulation can accommodate incompatible views and avoid destructive knowledge combination. Then we studied the costs of the induced cuts on three different sub-populations, namely Healthy, MCI, and Dementia. As shown in Figure 5(a), as the cognitive symptom develops, the costs of the induced cuts tend to increase, which means the strength of the DMN tends to decrease. To verify this, we tried a different exemplary scan and had similar results (Figure 5(b)). This observation provided direct support to the claim made in previous study [7] that the DMN diminishes as the Alzheimer's disease progresses.

Existing multi-view techniques do not work well for this task since they assume compatible views. However, the two views, the exemplary and the target scan, are often incompatible due to not only the noise but also the fact that they are from different individuals. Consequently, existing methods suffer from destructive combination as indicated by earlier results (see Table 2). Moreover, the pattern we are interested in, the DMN, is often not the dominant pattern in the exemplary scan. This makes it much more difficult, if possible, for single-objective based techniques to find the DMN pattern in all the target scans.

6 Conclusion

In this paper we explored multi-view spectral clustering using a multi-objective formulation. The search space of our objective is the joint numerical range of two graphs. We use Pareto optimization to find the optimal solutions, which is the Pareto frontier of the joint numerical range. To the best of our knowledge, we are the first to use Pareto optimization for multi-objective multi-view spectral clustering. We also proposed an efficient approximation algorithm to compute the Pareto frontier, which reduces the search space from an infinite number of cuts to a finite set of mutually orthogonal cuts. We compared our work against a variety of algorithms in the multi-view setting. The pragmatic benefits of our approach over existing single-objective techniques are: 1) the users do not need to specify the weights for different views *a priori*; 2) the views need not to be compatible (a difficult-to-test property); 3) it efficiently enumerates plausible and alternative clusterings. We also explored using our multi-objective formulation in the setting where one objective captures the adherence to the ground truth and the other the adherence to the observed data.

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. The authors thank Professor Owen Carmichael from Department of Neurology at UC Davis and UC Davis Alzheimer's Disease Center for providing the fMRI dataset.

References

[1] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[2] F. R. Bach and M. I. Jordan. Learning spectral clustering. In *NIPS*, 2003.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, 2000.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2001.

[5] S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, pages 19–26, 2004.

[6] A. Blum and T. M. Mitchell. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100, 1998.

[7] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter. The brain's default network. *Annals of the New York Academy of Sciences*, 1124(1):1–38, 2008.

[8] C. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *UAI*, 2008.

[9] C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combinations of kernels. In *NIPS*, pages 396–404, 2009.

[10] V. de Sa. Spectral clustering with two views. In *ICML workshop on learning with multiple views*, pages 20–27, 2005.

[11] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins Univ. Press, 1996.

[12] R. Horn and C. Johnson. *Matrix analysis*. Cambridge Univ. Press, 1990.

[13] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[14] Y. Jin and B. Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(3):397–415, 2008.

[15] A. Kumar and H. Daumé III. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

[16] A. Kumar, P. Rai, and H. Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[17] B. Long, P. S. Yu, and Z. M. Zhang. A general model for multiple view unsupervised learning. In *SDM*, pages 822–833, 2008.

[18] I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, pages 435–442, 2002.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[20] L. Tang, X. Wang, and H. Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.*, 25(1):1–33, 2012.

[21] W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *ICDM*, pages 1016–1021, 2009.

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

[23] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, pages 1159–1166, 2007.