# Multi-View Graph Embedding Using Randomized Shortest Paths

Anuththari Gamage*, Brian Rappaport*, Shuchin Aeron*, and Xiaozhe Hu†
*Department of Electrical and Computer Engineering, Tufts University
anuththari.gamarallage@tufts.edu brian.rappaport@tufts.edu shuchin@ece.tufts.edu
†Department of Mathematics, Tufts University
xiaozhe.hu@tufts.edu

***Abstract*—Real-world data sets often provide multiple types of information about the same set of entities. This data is well represented by multi-view graphs, which consist of several distinct sets of edges over the same nodes. These can be used to analyze how entities interact from different viewpoints. Combining multiple views improves the quality of inferences drawn from the underlying data, which has increased interest in developing efficient multi-view graph embedding methods. We propose an algorithm, C-RSP, that generates a common (C) embedding of a multi-view graph using Randomized Shortest Paths (RSP). This algorithm generates a dissimilarity measure between nodes by minimizing the expected cost of a random walk between any two nodes across all views of a multi-view graph, in doing so encoding both the local and global structure of the graph. We test C-RSP on both real and synthetic data and show that it outperforms benchmark algorithms at embedding and clustering tasks while remaining computationally efficient.**

***Index Terms*—multi-view graphs, multi-view graph embedding, graph distances, randomized shortest paths, graph clustering**

## I. Introduction

To model and understand complex systems, we must consider how different entities within a system relate to one another. Many such relational data sets provide multiple views of the same underlying set of entities. For example, a group of people can be characterized by their interactions with one another on a social networking platform. As a first approximation, we can consider only whether a relationship exists between two people on this platform. However, we can also study their interactions across other platforms, or across the different modes of communications provided by the platform [1], which provide us multiple views of the relationships between the same group of people. Other examples of multi-view data sets include multi-omics measurements in single cell RNA sequencing data [2] and 2D projections of a single 3D object captured from multiple angles for 3D reconstruction [3].

Graphs are used extensively to model this type of relational data for machine learning tasks, where the nodes or vertices of the graph represent the entities studied in the data set and edges represent their relationships. By learning a vector for each node in the graph, we can create a graph embedding, which gives the entities in the data set a representation in Euclidean space. These embeddings can be used for various applications such as data visualization, clustering, and link prediction. There are several well-known algorithms for creating graph embeddings from a single-view graph. The extension to multi-view or multi-layer graphs, however, has not been well studied up to this point. Providing more views leads to improved accuracy in clustering and embedding. This has increased recent interest in developing efficient multi-view graph embedding methods.

In a multi-view graph embedding, each node is assigned a vector that incorporates data from all views of the graph. Simple methods to create multi-view embeddings include combining multiple views of the graph into one graph using an AND/OR aggregation of the edge sets and embedding the resulting single graph, or embedding each view independently and concatenating the different embeddings obtained for each node [4]. More sophisticated algorithms have been developed based on matrix factorization [5], [6], tensor factorization [7], [8], and spectral embedding [9]–[11]. Many of these algorithms focus on clustering multi-view graphs, a specific application thereof. High clustering accuracy indicates a good embedding since relative similarity between nodes should be correctly reflected in the embedding.

The similarity between nodes of a graph can be quantified by a distance measure, such as the shortest path or geodesic distance (number of edges in the shortest path connecting two nodes) or the commute time distance (expected number of edges in a random walk from one node to the other and back). If two nodes are similar, then they are likely to have a shorter distance between them. Ideally, graph embeddings should preserve the distances between the nodes in their respective node embeddings. The commute time distance encodes a graph's clusters better than the shortest path distance [12], [13]. However, for large graphs with greater than 1000 nodes, or for graphs where the dimensionality of the underlying data is high, the commute time distance fails to capture the global structure of the graph accurately [14]. This is because it degenerates to a function of the node degrees, which captures only the local connectivity of the nodes [14].

This work was supported by the US National Science Foundation (NSF) grants CCF-1319653 and CCF-1553075.

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

In light of these deficiencies of the commute time and shortest path distance measures, there has been increased interest in alternative distance measures that generalize these two distances [15]–[17]. The Randomized Shortest Path (RSP) dissimilarity measure [16] generalizes the two distances by computing an intermediate measure parameterized by a tunable variable $\beta$, such that both limiting cases reduce to one of these measures: as $\beta \to \infty$, the RSP dissimilarity reduces to the shortest path distance and as $\beta \to 0$, it reduces to the commute time distance. This type of distance measure is particularly suitable for graph embedding as it preserves in the embedding space both the local and global features of the manifold from which the data set is sampled.

In this paper, we propose a generalized distance on multi-view graphs called the **Common Randomized Shortest Path Dissimilarity (C-RSP)** based on the RSP dissimilarity on single-view graphs. We highlight some of the advantages of the proposed approach below:

1) Like RSP, C-RSP generalizes the shortest path and commute time distances on multi-view graphs using a single parameter $\beta$. As $\beta \to \infty$, it reduces to the shortest path distance and as $\beta \to 0$, it reduces to the commute time distance. This type of generalized distance generates more accurate graph embeddings, and as a result, produces higher clustering and visualization accuracy for a given data set.
2) The RSP dissimilarity has an intuitive interpretation as the minimum expected cost of a random walk between any two nodes of a graph over all possible transition probability matrices [16]. C-RSP has a similar interpretation: the minimum expected cost of a random walk between nodes across all views.
3) C-RSP retains the computational efficiency of RSP [16], [17]. The proposed algorithm first combines the multiple views and then generates an embedding using the combined matrix, eliminating the need for factorization and simultaneous optimization approaches found in other multi-view graph embedding algorithms. This makes it less computationally intensive and more scalable.

The rest of the paper is organized as follows: Section II provides an overview of the Randomized Shortest Path dissimilarity measure. Section III describes the proposed C-RSP algorithms and its derivation. Section IV presents experimental results comparing C-RSP to benchmark multi-view algorithms on a variety of synthetic and real-world data sets, comparing their clustering and embedding accuracy. Section V discusses our results and future work.

## II. Randomized Shortest Path Dissimilarity

### A. Mathematical Preliminaries

Let $G = \{V, E\}$ be a simply connected graph, where $V = \{1, \ldots, n\}$ denotes the set of nodes of the graph and $E = \{(i,j) \,|\, i,j \in V\}$ denotes the set of edges between nodes. This graph can be represented by its affinity matrix $A \in \mathbb{R}^{n \times n}$, where the elements $a_{ij}$ are termed the *affinities* or *weights*. $a_{ij} = 1$ for $(i,j) \in E$ in unweighted graphs, $a_{ij} \neq 0$ for $(i,j) \in E$ in weighted graphs, and for all graphs $a_{ij} = 0$ if $(i,j) \notin E$. The degree matrix $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix containing the weighted degree (the sum of all edges leaving the node) of node $i$ in element $D_{ii}$ and zeros elsewhere.

We can compute the transition probability matrix $P^{\text{ref}} = D^{-1}A$ of the graph $G$, which is row-stochastic and defines a probability distribution on the edges of the graph. A random walk on the graph follows a sequence of nodes with the order determined by these transition probabilities. Consider a particular path on this graph starting at a source node $s$ and a destination node $t$, denoted by $p_{s \to t} = \{s, v_1, v_2, \ldots, v_m, t\}$. Then the probability of the path is given by the product $P^{\text{ref}}_{s,v_1} P^{\text{ref}}_{v_1,v_2} \ldots P^{\text{ref}}_{v_m,t}$, denoted $P^{\text{ref}}(p_{s \to t})$.

Since affinities refer to a positive correlation between nodes, we define the *cost* of each edge $(i,j)$ by a cost matrix $C$ with elements $c_{ij} = a_{ij}^{-1}$ where $0 < c_{ij} < \infty$. We can compute the total cost for a given path $p_{s \to t}$, denoted by $C(p_{s \to t}) = c_{s,v_1} + c_{v_1,v_2} + \ldots + c_{v_m,t}$.

An *absorbing* path is a path where the destination node $t$ has no outgoing edges except to itself ($c_{t,t} = 1, c_{t,k} = \infty\,, k \neq t \in V$). The cost of an absorbing path (even permitting infinite length ones) is finite since a random walk on the path will terminate in a finite number of steps with probability 1.

For C-RSP, we consider only absorbing paths from $s$ to $t$, and our path is denoted $p_{s \to t}$, with the path probability under a given distribution $P$ denoted by $P(p_{s \to t})$ and the cost of traversing the path denoted by $C(p_{s \to t})$. Suppose the set of all such absorbing paths is $\mathcal{P}_{s \to t}$. Then, the *expected cost* of a random walk from a source node $s$ to a destination node $t$ over a given distribution $P$ is given by $\sum_{p \in \mathcal{P}_{s \to t}} P(p)C(p)$.

### B. Randomized Shortest Paths Dissimilarity

The Randomized Shortest Path (RSP) is defined to be the path between two nodes with the minimum expected cost over all transition probability matrices [16]. In order to constrain a random walk between two nodes to a RSP, we compute a new probability distribution $P(p)$ that achieves the minimum expected cost among all possible probability distributions having fixed relative entropy (Küllback-Leibler divergence) with respect to the reference probability distribution $P^{\text{ref}}(p)$:

$$\begin{aligned} P^{RSP} &= \operatorname*{argmin}_{P} \sum_{p \in \mathcal{P}_{s \to t}} P(p)C(p) \\ \text{subject to } & \sum_{p \in \mathcal{P}_{s \to t}} P(p) \ln \frac{P(p)}{P^{\text{ref}}(p)} = J_0, \\ & \sum_{p \in \mathcal{P}_{s \to t}} P(p) = 1 \end{aligned} \tag{1}$$

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

The solution to this constrained optimization is given by the following expression for any $p_{s\to t} \in \mathcal{P}_{s\to t}$ [16]:

$$P^{RSP}(p_{s\to t}) = \frac{P^{\text{ref}}(p_{s\to t})e^{-\beta C(p_{s\to t})}}{\sum_{p\in\mathcal{P}_{s\to t}} P^{\text{ref}}(p)e^{-\beta C(p)}} \tag{2}$$

Using the probability distribution for Randomized Shorted Paths derived above, we can define the symmetric RSP dissimilarity between two nodes as follows. Suppose the expected cost of traversing the randomized shortest path between source node $s$ and destination node $t$ is given by

$$\overline{C}_{st} = \sum_{p\in\mathcal{P}_{s\to t}} P^{RSP}(p)C(p).$$

This expression is not guaranteed to be symmetric, so we calculate the symmetric RSP dissimilarity measure between the two nodes by

$$\Delta^{RSP}_{st} = \frac{\overline{C}_{st} + \overline{C}_{ts}}{2}. \tag{3}$$

Note that the computed RSP distance measure is termed a "measure" instead of a "metric" since it does not follow the triangle inequality for certain ranges of $\beta$ used [16], [17].

### C. Efficient Computation of the RSP Dissimilarity

Following the derivation of the RSP dissimilarity by Yen et al. [16], an efficient closed-form expression for its computation was derived by Kivimäki et al. [17], which we describe in Algorithm 1. This computation is done entirely through matrix operations, which lends itself nicely to input graphs in the form of affinity matrices. The output of the algorithm is the symmetric matrix $\Delta^{RSP} \in \mathbb{R}^{n\times n}$, in which each entry $\Delta^{RSP}_{ij}$ gives the RSP dissimilarity between the nodes $i$ and $j$.

**Algorithm 1** RSP Dissimilarity

**Input:** $A \in \mathbb{R}^{n\times n}$ (affinity matrix for a simply connected graph $G$), $\beta$ (optimization parameter)
**Output:** $\Delta^{RSP} \in \mathbb{R}^{n\times n}$ (RSP dissimilarity matrix)

$P^{\text{ref}} = D^{-1}A$ ($D$ is the degree matrix of $A$)
$C = 1 \div A$ (element-wise inverse)
$W = P^{\text{ref}} \circ e^{-\beta C}$ (element-wise multiplication)
**if** $\rho(W) \geq 1$ **then**
  Stop: will not converge
**end if**
$Z = (I - W)^{-1}$  ($I \in \mathbb{R}^{n\times n}$ is the identity matrix)
$S = (Z[C \circ W]Z) \div Z$
$\overline{C} = S - \mathbf{1}d_S^T$  ($\mathbf{1}, d_S \in \mathbb{R}^n$, $d_i = S_{ii}$)
$\Delta^{RSP} = \frac{1}{2}(\overline{C} + \overline{C}^T)$

## III. Combining Multiple Graph Views Using Common Randomized Shortest Paths

### A. Deriving a Common RSP Probability Distribution

In this work, we extend the core RSP framework to generate a multi-view graph distance measure. If we represent a single-view graph by $G = \{V, E\}$, then a multi-view graph is denoted $\mathcal{G} = \{V, (E_1, \ldots, E_m)\}$ where each view is given by $G_i = \{V, E_i\}$. We represent this graph with an $n \times n \times m$ affinity tensor, where each $n \times n$ slice of the tensor $A_i$ represents the affinity matrix for that edge set. Note that each $G_i$ is assumed to be a simply connected graph.

We first derive a common probability distribution, $P^{CRSP}$, over all views of the graph. This is accomplished by minimizing the expected cost for all possible paths on all views, with the condition that the common distribution $P^{CRSP}$ and the reference probability distribution of each view, $P_i^{\text{ref}}$, have the same fixed relative entropy. This constrained optimization is represented as follows, with reference probability distributions $P_1^{\text{ref}}, \ldots, P_m^{\text{ref}}$ and cost matrices $C_1, \ldots, C_m$:

$$\begin{aligned} P^{CRSP} &= \underset{P}{\text{argmin}} \sum_{i=1}^{m} \sum_{p\in\mathcal{P}_{s\to t}} P(p)C_i(p) \\ \text{subject to } &\sum_{i=1}^{m} \sum_{p\in\mathcal{P}_{s\to t}} P(p) \ln \frac{P(p)}{P_i^{\text{ref}}(p)} = J_0, \\ &\sum_{p\in\mathcal{P}_{s\to t}} P(p) = 1 \end{aligned} \tag{4}$$

Solving this constrained optimization results in the following probability distribution, which we term the Common Randomized Shortest Paths (C-RSP) distribution. Multiplication indicated by "$\circ$" is done element-wise. The full derivation of this expression is detailed in the Appendix. This equation holds for any $p_{s\to t} \in \mathcal{P}_{s\to t}$.

$$P^{CRSP}(p_{s\to t}) = \frac{\sqrt[m]{\bigcirc_{i=1}^{m} P_i^{\text{ref}}(p_{s\to t})} \circ e^{-\beta \sum_{i=1}^{m} C_i(p_{s\to t})}}{\sum_{p\in\mathcal{P}_{s\to t}} \sqrt[m]{\bigcirc_{i=1}^{m} P_i^{\text{ref}}(p)} \circ e^{-\beta \sum_{i=1}^{m} C_i(p)}} \tag{5}$$

### B. Common Randomized Shortest Paths Dissimilarity

Using the derived common probability distribution, $P^{CRSP}$, we can compute a dissimilarity measure $\Delta^{CRSP}$ for multi-view graphs following an approach similar to that detailed in section II-B above. Note that we can use the same algorithm used for computing RSP if we were to have a single reference probability matrix and a single cost matrix instead of the tensors associated with a multi-view graph. Using the expression for $P^{CRSP}(p_{s\to t})$ derived in equation (5), we can combine the individual views of these tensors to obtain these matrices as detailed below.

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

Let $\overline{\mathbf{P}}$ denote the combined reference transition probability matrix and $\overline{\mathbf{C}}$ denote the combined cost matrix. By comparing equations (2) and (5), we obtain

$$\overline{\mathbf{P}} = \frac{\sqrt[m]{\overset{m}{\underset{i=1}{\circ}} P_i^{\text{ref}}}}{\mathbf{1}^T \cdot \sqrt[m]{\overset{m}{\underset{i=1}{\circ}} P_i^{\text{ref}}}} \tag{6}$$

$$\overline{\mathbf{C}} = \sum_{i=1}^{m} C_i \tag{7}$$

Note that in equation (5), we derive the probability of an individual path and not the entire set of possible paths. Thus, we need to take care to omit instances when the path does not exist given a particular $P_i^{\text{ref}}$, which occurs when an entry in any $P_i^{\text{ref}}$ is zero. Also note that the multiplication in this expression is taken element-wise, as is the $m^{th}$ root. Finally, note that this manner of combining the different $P_i^{\text{ref}}$ matrices does not guarantee a row-stochastic matrix, which is necessary for it to be a probability distribution. Thus, the resulting matrix $\overline{\mathbf{P}}$ must be normalized to obtain a row-stochastic matrix. This can be achieved easily by dividing the entries in each row by the row sum. In this way, we obtain a combined reference probability matrix $\overline{\mathbf{P}}$ and a combined cost matrix $\overline{\mathbf{C}}$ that can then be used in the original RSP algorithm to obtain the C-RSP dissimilarity measure $\Delta^{CRSP}$, as detailed in Algorithm 2.

**Algorithm 2** C-RSP Dissimilarity

**Input:** $A_1, \ldots, A_m$ ($A_i \in \mathbb{R}^{n \times n}$ is the affinity matrix for view $G_i$ of a multi-view graph $G$ where $G_i$ is connected), $\beta$ (optimization parameter)
**Output:** $\Delta^{CRSP} \in \mathbb{R}^{n \times n}$ (C-RSP dissimilarity matrix)
  **for** $i = 1 \ldots m$ **do**
    $P_i^{\text{ref}} = D_i^{-1} A_i$ ($D_i$ is the degree matrix of $A_i$)
    $C_i = 1 \div A_i$ (element-wise division)
  **end for**
  $\overline{\mathbf{P}} = \{P_1^{\text{ref}}, \ldots, P_m^{\text{ref}}\}$ combined as given in equation (6)
  $\overline{\mathbf{C}} = \sum_{i=1}^{m} C_i$
  $W = \overline{\mathbf{P}} \circ e^{-\beta \overline{\mathbf{C}}}$ (element-wise multiplication)
  **if** $\rho(W) \geq 1$ **then**
    Stop: will not converge
  **end if**
  $Z = (I - W)^{-1}$ ($I \in \mathbb{R}^{n \times n}$ is the identity matrix)
  $S = (Z[C \circ W]Z) \div Z$
  $\overline{C} = S - \mathbf{1} d_S^T$ ($\mathbf{1}, d_S \in \mathbb{R}^n$, $d_i = S_{ii}$)
  $\Delta^{CRSP} = \frac{1}{2}(\overline{C} + \overline{C}^T)$

Using this C-RSP dissimilarity matrix, we obtain a multi-view graph embedding by applying Multidimensional Scaling. To cluster a multi-view graph, we can use Spectral Clustering [13] on $(\Delta^{CRSP})^{-1}$, which serves as a measure of affinity.

## IV. Experimental Results

For a general graph, evaluating the effectiveness of its embedding is not straightforward, as we do not know *a priori* how distant the nodes should be once they are embedded or how they should be oriented relative to each other. However, if we know that the graph contains latent clusters of nodes, we can assume that the nodes belonging to the same cluster are likely to appear closer together in their embedding, while those that are in different groups are likely to be distant. Thus, for data sets with latent clusters, we can evaluate the embedding of its representative graph by the clustering accuracy achieved using the embedding vectors.

In order to evaluate C-RSP, we first test the quality of its embedding on a standard Swiss roll data set and visually compare it to the embeddings generated by other algorithms. We then test its clustering performance against a number of benchmark multi-view graph clustering algorithms on a variety of data sets with latent clusters. Clustering performance is compared using two metrics: the Correct Classification Rate (CCR) as a percent and the Normalized Mutual Information (NMI) between the ground truth and the derived clusters. Our experimental results are available online at https://github.com/Anu-Gamage/C-RSP.

### A. Benchmark Algorithms

- **SC-ML**: Spectral Clustering on Multi-Layer Graphs using the Grassmannian Manifold [9]
  This algorithm embeds a multi-view graph by projecting each of the different views of the graph into the Grassmannian manifold. These projections are then combined into a consensus matrix to represent the full multi-view graph, which is clustered using Spectral Clustering [13]. We use $\lambda = 0.5$ in our tests.
- **CSC**: Co-regularized Spectral Clustering [10]
  This method combines the views of a multi-view graph using co-regularization, a process that chooses the optimal embedding based upon its similarity with all different views of the data. Two co-regularization algorithms are commonly used: pairwise and centroid-based co-regularization. In this work, we use centroid-based co-regularization, which pushes the eigenvector matrices of all views towards a common consensus matrix. The resulting matrix is then clustered using Spectral Clustering. We use $\lambda = 0.05$ across all tests.
- **MultiNMF**: Joint Non-negative Matrix Factorization [6]
  This algorithm factorizes each view of the graph into a basis matrix and a coefficient matrix and computes a consensus matrix such that the coefficient matrices are relatively similar to the consensus matrix. The $i^{th}$ row of the consensus matrix is then taken as the vector embedding of the $i^{th}$ node and clustered using $k$-means clustering. We use the parameters listed in the original code in all of the tests.

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

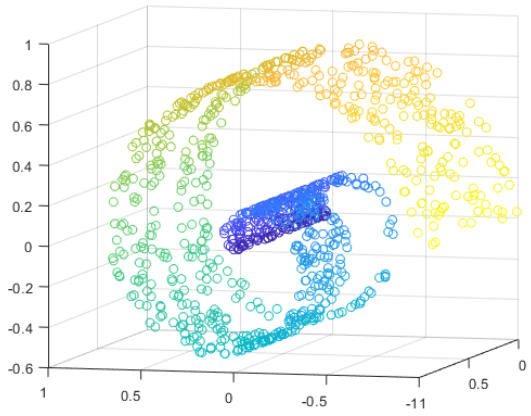
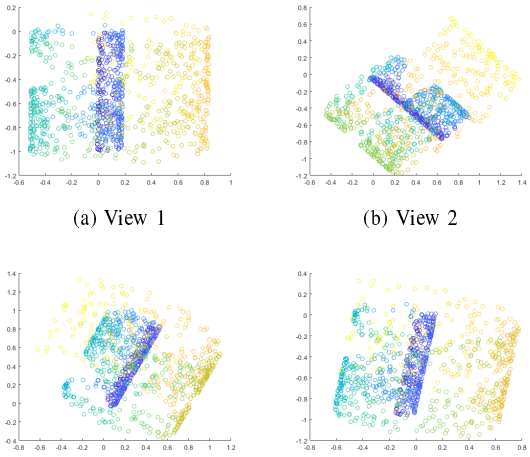


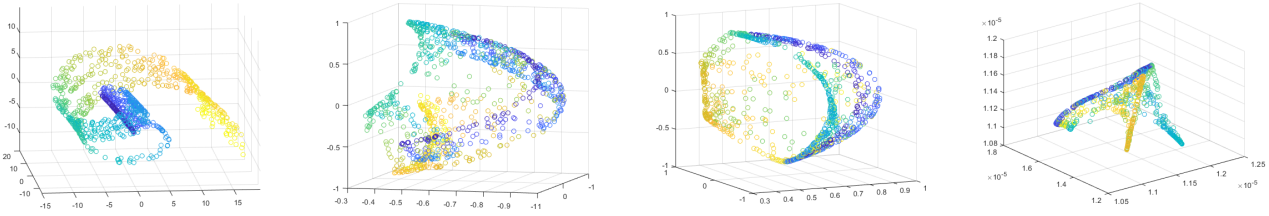
Fig. 1. Swiss roll ground truth (in 3D), with two holes in the blue and green areas.



(a) View 1 (b) View 2

(c) View 3 (d) View 4

Fig. 2. Multiple 2D projections of the Swiss roll, used as different views in the multi-view graph.



(a) C-RSP embedding (b) SC-ML embedding (c) CSC embedding (d) MultiNMF embedding

Fig. 3. Embeddings of the Swiss roll generated by C-RSP and other benchmark algorithms. C-RSP retains the Swiss roll shape as well as the relative distances between nodes. SC-ML and CSC both use spectral embedding methods which preserve the relative distances of the nodes but lose the overall structure, while MultiNMF uses a nonnegative matrix factorization to obtain the embedding vectors.

### *B. Synthetic Data Sets*

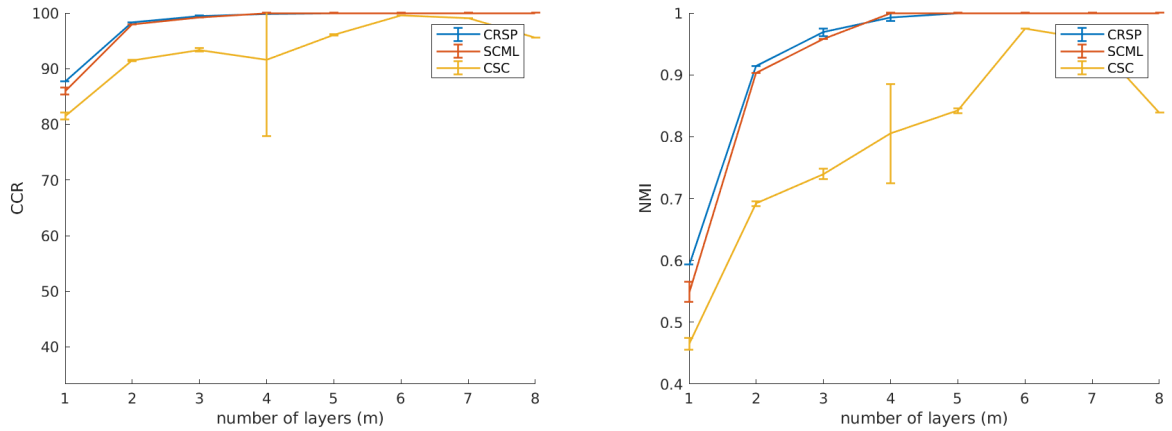
- **Swiss Roll**: To compare the embedding performance across algorithms, we constructed a 3-dimensional Swiss roll with holes, a standard test case for this task. Points were distributed in a plane with holes removed from the plane and the plane was wrapped in a spiral to create a relatively complex 3-dimensional structure. We obtained multiple views of the Swiss roll by projecting it onto planes at different angles, resulting in a number of affinity matrices forming a multi-view graph. The goal is to reconstruct the original Swiss roll geometry using the embeddings generated by each algorithm.
- **Stochastic Block Model**: The stochastic block model (SBM) is used to simulate graphs with a latent cluster structure. To generate a graph under this model, the nodes are partitioned into $k$ equally sized clusters. Each possible edge is determined with a specific probability, with intra-cluster edges assigned with a probability of $\frac{c}{n}$, where $c$ represents the average degree of the graph; and inter-cluster edges assigned with probability $\frac{c(1-\lambda)}{n}$, where $\lambda$ is a parameter used to determine how distinct the clusters should be: $\lambda = 0.9$ implies that the edges are $\frac{9}{10}$ less likely to occur between clusters as within a cluster. To simulate multi-view graphs, we generate $m$ independent SBM graphs with the same set of parameters $n, k, c$, and $\lambda$, and the same partition of nodes for each view. Since this process does not necessarily produce connected graphs, we cull nodes that are not connected in each of the views. We measure clustering performance on this data set, varying the number of clusters, number of layers, and sparsity of the generated graphs.

### *C. Real-world Data Sets*

- **3Sources**[1]: This data set contains information about a set of news articles reported by three different news sources: the BBC, the Guardian, and Reuters. It covers 416 distinct news stories, of which 169 are reported on by all three agencies. These stories are classified under 6 disjoint clusters: business, entertainment, health, politics, sports, and technology. The three sources provide different views of the same news story, which are represented as different views of a multi-view graph. In our experiments, we extracted the 169 stories common to all three sources and constructed a $169 \times 169$ affinity matrix for each source using a Gaussian kernel on the

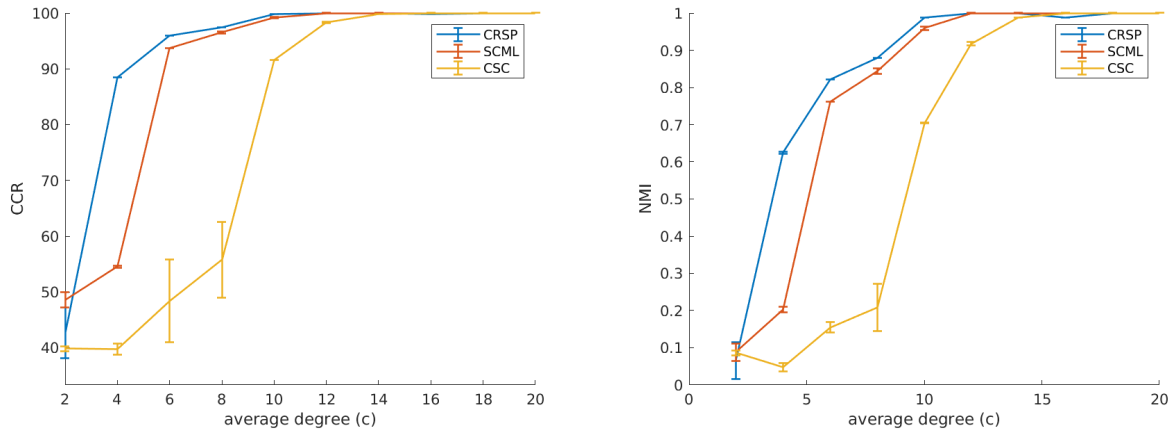
[1] http://mlg.ucd.ie/datasets/3sources.htm

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.




(a) CCR for varying number of views

(b) NMI for varying number of views




(c) CCR for varying graph sparsity

(d) NMI for varying graph sparsity

Fig. 4. Simulation results on Stochastic Block Model graphs with $n = 500, k = 3, \lambda = 0.9$ averaged over 10 runs. As the number of views increases, all algorithms perform better, with C-RSP outperforming the others. Similarly as the average degree increases, resulting in denser graphs, all algorithms perform more and more accurately, with C-RSP once again outperforming the two benchmark algorithms.

feature vectors provided. If $a_{ij}$ were the affinity between nodes $i$ and $j$ and their corresponding feature vectors were $x_i$ and $x_j$, then $a_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$, where $\sigma$ was taken to be the median of the pairwise Euclidean distances between the feature vectors.

- **UCI Handwritten Digits**[2]: In this data set, 2000 images of handwritten digits from 0 to 9 are analyzed, resulting in features matrix of Fourier coefficients, pixel averages, and several other feature matrices. In our experiments, we used 5 of the 6 feature types (excluding the Karhunen-Love coefficients) as our multiple views and constructed affinity matrices as before using a Gaussian kernel.
- **Multi-view Twitter**[3]: This data set consists of five Twitter user networks and various methods of interaction on Twitter. We chose the *politics-uk* data set, consisting of 419 user accounts belonging to political figures and organizations from the UK and 3 views of their pairwise interactions: $x$ follows $y$, $x$ mentions $y$, and $x$ retweets $y$. The user accounts form the nodes of the three $419 \times 419$ graphs, and they are partitioned into 5 disjoint clusters based on political party affiliation: Labour, Conservative, Scottish National Party, Liberal Democrats, and other.

[2]https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3]http://mlg.ucd.ie/aggregation/index.html

### D. Results on Synthetic Data

We first tested the quality of the embeddings generated by C-RSP using the Swiss roll data set in figure 1. To obtain a multi-view graph from this data, the Swiss roll was rotated and projected into 2 dimensions, resulting in the four views pictured in figures 2a - 2d. An affinity matrix for each view was constructed using the pairwise Euclidean distances between points. In the case of C-RSP, the embeddings were generated from the output C-RSP distance matrix using Multidimensional Scaling. For the benchmarks, the embedding vectors generated via each algorithm prior

TABLE I
CLUSTERING RESULTS ON VARIOUS REAL-WORLD DATA SETS, IN THE FORM "MEAN (STD)"

| Metric | Data Set | C-RSP | SC-ML | CSC | MultiNMF |
|---|---|---|---|---|---|
| CCR | **UCI** | 86.96% (4.10%) | 80.09% (6.76%) | 82.11% (7.99%) | 92.33% (0.03%) |
| | **3Sources** | 58.22% (4.14%) | 51.48% (3.55%) | 43.55% (2.59%) | 34.50% (0.02%) |
| | **MultiviewTwitter** | 82.84% (3.96%) | 69.86% (1.50%) | 49.54% (3.19%) | 56.67% (0.01%) |
| NMI | **UCI** | 0.80 (0.01) | 0.76 (0.03) | 0.78 (0.04) | 0.88 (0.02) |
| | **3Sources** | 0.56 (0.04) | 0.42 (0.02) | 0.31 (0.02) | 0.07 (0.01) |
| | **MultiviewTwitter** | 0.60 (0.04) | 0.42 (0.01) | 0.28 (0.01) | 0.45 (0.01) |

to clustering the vectors were used to obtain labels for the nodes.

As seen in figure 3a, the C-RSP embedding accurately captures the curvature of the Swiss roll and produces a slightly flattened version of the original spiral structure The embedding also retains the two holes present in the original Swiss roll data. The benchmark algorithms SC-ML, CSC, and MultiNMF all fail to recover the spiral structure and the holes, but retain the relative distances between nodes with some accuracy. This is shown by the grouping of similarly colored nodes in figures 3b-3d.

To evaluate C-RSP at clustering tasks, we tested it on synthetic multi-view graphs generated using the Stochastic Block Model. Unless otherwise noted, the graphs have $n = 500$ nodes, $k = 3$ clusters, an average node degree of $c = 10$, and $\lambda = 0.9$. In figure 4, we report the variation in CCR and NMI as the number of views of the multi-view graph increases, as well as the variation across multi-view graphs of different sparsity. In both cases, C-RSP shows significantly better clustering accuracy compared to the benchmark algorithms.

### *E. Results on Real-world Data*

Having observed that C-RSP shows high embedding and clustering accuracy on synthetic data sets, we now evaluate whether this performance holds on real-world data sets. For this, we run C-RSP and all the benchmark algorithms on three widely used multi-view data sets. In table I, we report the CCR and NMI values obtained for each algorithm averaged over 10 runs with the standard deviation listed in parentheses.

C-RSP significantly outperforms the benchmark algorithms on the MultiviewTwitter data set with respect to both CCR and NMI. On 3Sources, the CCR of C-RSP is almost matched by SC-ML, but we observe a significant increase in the NMI of C-RSP compared to the other algorithms. MultiNMF shows much higher clustering performance on the UCI Handwritten Digits data set, while the other three methods have comparable values in both metrics. The dip in performance of C-RSP is likely due to the choice of parameter. For all experiments conducted above, we chose $\beta = 0.02$ for C-RSP since it is shown to be the optimal $\beta$ value for the RSP measure [17]. At this $\beta$, the C-RSP dissimilarities calculated tend more towards the commute time distances, which may not capture the graph structure effectively for graphs larger than 1000 nodes [14], to which category the UCI data set falls.

Overall, C-RSP provides superior clustering results, confirming that good multi-view graph embeddings results in higher clustering accuracy. Furthermore, the results show that C-RSP has robust performance across different types of data, providing reasonably high clustering across the board. A more extensive parameter study on C-RSP, which remains to be completed, would optimize the performance of C-RSP further.

### *F. Comparison of Computational Speed*

Since all algorithms were coded in MATLAB, we were able to obtain a fair comparison of their respective running times. On smaller graphs like 3Sources, C-RSP and SC-ML had comparable run times, both taking only 25% of the time taken by CSC. All three algorithms were significantly faster than MultiNMF. On the larger UCI data set, SC-ML finished in roughly 60% of the run time of C-RSP, but both were still faster than CSC and MultiNMF. Overall, C-RSP has an efficient run time, especially compared to CSC and MultiNMF, and we believe that it could be made more efficient by using faster method of computing the matrix inversion $(I - W)^{-1}$, which is the most computationally intensive step of the algorithm. A more rigorous experimental analysis of the comparative speeds of the multi-view algorithms remains to be completed.

## V. CONCLUSION

This paper introduced a novel distance measure for multi-view graphs named C-RSP (Common Randomized Shortest Paths), an extension of the RSP dissimilarity for single-view graphs. The C-RSP measure is a generalization of the commute time distance and the shortest path distance, which allows it to encode both the local and global structure of a multi-view graph. This leads to more accurate graph embeddings, resulting in better visualization and high clustering accuracy. We tested C-RSP at both embedding and clustering tasks and showed that it produces superior results compared to other benchmark embedding algorithms while being computationally efficient.

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

## REFERENCES


[1] D. Greene and P. Cunningham, "Producing a unified graph representation from multiple social network views," in *Proceedings of the 5th Annual ACM Web Science Conference*, ser. WebSci '13. New York, NY, USA: ACM, 2013, pp. 118–121. [Online]. Available: http://doi.acm.org/10.1145/2464464.2464471

[2] M. Colome-Tatche and F. Theis, "Statistical single cell multi-omics integration," *Current Opinion in Systems Biology*, vol. 7, pp. 54 – 59, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2452310018300039

[3] K. Kolev, M. Klodt, T. Brox, and D. Cremers, "Continuous global optimization in multiview 3d reconstruction," *Int. J. Comput. Vision*, vol. 84, no. 1, pp. 80–96, Aug. 2009. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0233-1

[4] Y. Shi, F. Han, X. He, C. Yang, J. Luo, and J. Han, "mvn2vec: Preservation and Collaboration in Multi-View Network Embedding," *ArXiv e-prints*, Jan. 2018.

[5] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ser. ICDM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1016–1021. [Online]. Available: https://doi.org/10.1109/ICDM.2009.125

[6] J. Liu, C. Wang, J. Gao, and J. Han, "Multi-view clustering via joint nonnegative matrix factorization," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 252–260.

[7] X. Liu, W. Glänzel, and B. D. Moor, "Hybrid clustering of multi-view data via tucker-2 model and its application," *Scientometrics*, vol. 88, no. 3, pp. 819–839, Sep. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11192-011-0348-3

[8] E. E. Papalexakis, L. Akoglu, and D. Ience, "Do more views of a graph help? community detection and clustering in multi-graphs," in *Proceedings of the 16th International Conference on Information Fusion*, July 2013, pp. 899–905.

[9] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 905–918, Feb 2014.

[10] A. Kumar, P. Rai, and H. Daumé, III, "Co-regularized multi-view spectral clustering," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. USA: Curran Associates Inc., 2011, pp. 1413–1421. [Online]. Available: http://dl.acm.org/citation.cfm?id=2986459.2986617

[11] A. Kumar and H. D. III, "A co-training approach for multi-view spectral clustering," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. USA: Omnipress, 2011, pp. 393–400. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104482.3104532

[12] U. Von Luxburg, A. Radl, and M. Hein, "Hitting and commute times in large random neighborhood graphs," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1751–1798, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2638591

[13] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11222-007-9033-z

[14] U. Von Luxburg, A. Radl, and M. Hein, "Hitting and commute times in large graphs are often misleading," *ArXiv e-prints*, Mar. 2010. [Online]. Available: https://arxiv.org/pdf/1003.1266.pdf

[15] T. B. Hashimoto, Y. Sun, and T. S. Jaakkola, "From random walks to distances on unweighted graphs," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 3429–3437. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969442.2969622

[16] L. Yen, M. Saerens, A. Mantrach, and M. Shimbo, "A family of dissimilarity measures between nodes generalizing both the shortest-path and the commute-time distances," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 785–793. [Online]. Available: http://doi.acm.org/10.1145/1401890.1401984

[17] I. Kivimäki, M. Shimbo, and M. Saerens, "Developments in the theory of randomized shortest paths with a comparison of graph node distances," *Physica A: Statistical Mechanics and its Applications*, vol. 393, pp. 600–616, 2014.


Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.

## APPENDIX
## DERIVATION OF THE C-RSP DISTRIBUTION

Suppose that $P_1^{\text{ref}}, \ldots, P_m^{\text{ref}}$ are the reference probability distributions of each view of a multi-view graph with cost matrices $C_1, \ldots, C_m$. To obtain Common Randomized Shortest Paths (C-RSP), we solve the following optimization problem:

$$P^{CRSP} = \underset{P}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{p \in \mathcal{P}_{s\to t}} P(p) C_i(p)$$
$$\text{subject to } \sum_{i=1}^{m} \sum_{p \in \mathcal{P}_{s\to t}} P(p) \ln \frac{P(p)}{P_i^{\text{ref}}(p)} = J_0$$
$$\sum_{p \in \mathcal{P}_{s\to t}} P(p) = 1$$

Consider a multi-view graph $\mathcal{G}$ with $m = 2$ layers, $G_1 = \{V, E_1\}$ and $G_2 = \{V, E_2\}$, with the reference transition probability distributions $P_1^{\text{ref}}$ and $P_2^{\text{ref}}$ and cost matrices $C_1$ and $C_2$ respectively. To derive the common distribution (which we will call $Q$ for ease of notation) under the above stated constrained optimization, we use the following Lagrange function:

$$\mathcal{L} = \sum_{i=1}^{m} \sum_{p \in \mathcal{P}_{s\to t}} Q(p) C_i(p) + \mu \left[ \sum_{p \in \mathcal{P}_{s\to t}} Q(p) - 1 \right] + \lambda \left[ \sum_{p \in \mathcal{P}_{s\to t}} Q(p) \ln \frac{Q(p)}{P_i^{\text{ref}}(p)} - J_0 \right]$$

Considering only one path, we obtain the following:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial Q} &= \sum_{i=i}^{m} C_i(p) + \lambda \sum_{i=1}^{m} \left( \ln \frac{Q(p)}{P_i^{\text{ref}}(p)} + 1 \right) + \mu \\
&= \sum_{i=i}^{m} C_i(p) + \lambda \ln \left( \prod_{i=1}^{m} \frac{Q(p)}{P_i^{\text{ref}}(p)} \right) + \lambda m + \mu \\
&= \sum_{i=i}^{m} C_i(p) + \lambda \ln \frac{Q^m(p)}{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} + \lambda m + \mu \\
&= 0
\end{aligned}$$

or

$$\ln \left[ \frac{Q^m(p)}{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} \right] = -\frac{1}{\lambda} \sum_{i=1}^{m} C_i(p) - \frac{\mu}{\lambda} - m$$

which gives

$$\begin{aligned}
Q^m(p) &= \prod_{i=1}^{m} P_i^{\text{ref}}(p) \cdot e^{-\frac{1}{\lambda} \sum_{i=1}^{m} C_i(p) - \frac{\mu}{\lambda} - m} \\
Q(p) &= \sqrt[m]{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} \cdot e^{-\frac{1}{m\lambda} \sum_{i=1}^{m} C_i(p) - \frac{\mu}{m\lambda} - 1} \\
&= \sqrt[m]{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} \cdot \left[ e^{-\frac{1}{m\lambda} \sum_{i=1}^{m} C_i(p)} \right] \left[ e^{-\frac{\mu}{m\lambda} - 1} \right] \\
&= c \sqrt[m]{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} \cdot e^{-\beta \sum_{i=1}^{m} C_i(p)} \\
&= c\, \overline{\mathbf{P}} \cdot e^{-\beta \overline{\mathbf{C}}}
\end{aligned}$$

Normalizing this to make it a probability distribution (which is the same as RSP, detailed in [17]), we obtain the following expression for the C-RSP probability distribution for a single path:

$$P^{CRSP}(p_{s\to t}) = \frac{\sqrt[m]{\prod_{i=1}^{m} P_i^{\text{ref}}(p_{s\to t})} \cdot e^{-\beta \sum_{i=1}^{m} C_i(p_{s\to t})}}{\sum_{p \in \mathcal{P}_{s\to t}} \sqrt[m]{\prod_{i=1}^{m} P_i^{\text{ref}}(p)} \cdot e^{-\beta \sum_{i=1}^{m} C_i(p)}}$$

When deriving the combined matrix $P^{CRSP}$, where all paths are considered, using the $P_i^{\text{ref}}$ matrices, the multiplication must be done element-wise.

Note that the constraint $J_0$ on the Küllback-Leibler Divergence disappears during the optimization and that it is not present in the expression derived for the C-RSP distribution above. Thus, $\beta := \frac{1}{m\lambda}$ is the only parameter that needs to be tuned for this distribution.

Preprint submitted to 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Received July 9, 2018.