

Randomly Walking Can Get You Lost: Graph Segmentation with Unknown Edge Weights

Hanno Ackermann¹, Björn Scheuermann¹, Tat-Jun Chin², and Bodo Rosenhahn¹

¹ Institute for Information Processing, Leibniz University Hannover, Germany
² The University of Adelaide, Australia

Abstract. Spectral graph clustering is among the most popular algorithms for unsupervised segmentation. Applications include problems such as speech separation, segmenting motions or objects in video sequences and community detection in social media. It is based on the computation of a few eigenvectors of a matrix defining the connections between the graph nodes.

In many real world applications, not all edge weights can be defined. In video sequences, for instance, not all 3d-points of the observed objects are visible in all the images. Relations between graph nodes representing the 3d-points cannot be defined if these never co-occur in the same images. It is common practice to simply assign an affinity of zero to such edges. In this article, we present a formal proof that this procedure decreases the separation between two clusters. An upper bound is derived on the second smallest eigenvalue of the Laplacian matrix. Furthermore, an algorithm to infer missing edges is proposed and results on synthetic and real image data are presented.

1 Introduction

Grouping similar data without any knowledge about the possible labeling is an important problem. This so-called unsupervised segmentation task is necessary in bioinformatics, machine learning, pattern recognition and computer vision.

One known technique for unsupervised segmentation is spectral clustering. It rests upon the segmentation of a graph capturing the relations between the data. Minimum cuts are used to decide the segmentation of the the graph into two [1, 2] or more sub-graphs [3].

Constructing the graph requires that affinities are computed between the vertices representing the data. If more than two data items are necessary to estimate the affinity, the corresponding *hyper*-edge connects all involved vertices [4]. For such hyper-graphs, the number of edges is exponential in the number of data items necessary to compute the affinity. Since many real-world problems induce prohibitively large hyper-edge sets, a commonly used approach is to estimate a subset of edges only [5–8], ie. many edge weights remain undefined.

In applications such as motion segmentation from video sequences [9] 2d-trajectories corresponding to different 3d-points have to be compared. Due to

occlusion or tracking failure, 2d-projections of different 3d-points may never co-occur in the same images so affinities between the corresponding graph nodes cannot be defined.

This situation is handled by state-of-the-art algorithms [6–9] by setting the corresponding edge weight to 0. This procedure is equivalent to assuming *maximum dissimilarity* between the two trajectories even though they may belong to the same group.

The contribution made here is twofold: (1) We model the impact of unknown edge weights in the context of spectral clustering. A *lower* bound on the separation between the two clusters and the *upper* bound on the eigenvalue gap is derived and proved. (2) We propose an algorithm to infer the weights of unknown edges using the known edges.

Whereas the effect of noise on the affinities has been investigated before [10, 11], this is the first work that considers the impact caused by undefined edges on spectral graph clustering.

The structure of this article is as follows: Some definitions are made in Section 2. Some facts of spectral clustering and the NCut-algorithm are shortly explained in the same section. Our first contribution, the derivation of upper and lower bounds on the cluster separability and the eigenvalue gap is presented in Sec. 3. Sections 4 and 5 present the proposed algorithm for inferring unknown edges of the graph and an application on motion segmentation. Experimental results using synthetic and real image data are shown in Sec. 6. The article concludes with a discussion in Sec. 7.

2 Definitions

For matrices, we use capital letters, such as matrix W . Vectors are indicated by lower-case letters, e.g. w . By $w(i)$ we denote the i th entry of a vector w whereas w_i denotes the i th vector, for instance given $W = [w_1 \cdots w_n]^\top$ the vector w_3 implies the third row of matrix W . The (i, j) th entry of matrix W is indicated by w_{ij} . By $\|v\|$, we mean the L_2 -norm of v .

Let $G = (V, E)$ be an undirected graph consisting of $|V| = n$ nodes V and a set of edges E connecting the nodes. Let there be subsets $V_1 \subset V$ and $V_2 \subset V$ such that $V_1 \cap V_2 = \emptyset$ and $|V_1| = n_1$, $|V_2| = n_2$, $n_1 + n_2 = n$.

The real-valued weight $w_{ij}(e_{ij})$ of an edge e_{ij} between two vertices v_i and v_j equals c_1 if and only if both v_i and v_j are vertices of either V_1 or V_2 . Otherwise, its edge weight equals $c_2 < c_1$. We may further assume that $c_1, c_2 \in [0, 1]$. The idea motivating c_1 and c_2 is that the clusters do not need be perfectly separated. Noise on the edge weights can be considered by decreasing c_1 and increasing c_2 , respectively.

Denote by W the matrix consisting of the edges weights between all nodes. Assuming without loss of generality that the vertices are sorted, the $n_1 \times n_1$ upper left block and lower right $n_2 \times n_2$ block of W are all but c_1 whereas the remaining entries of W equal c_2 .

Let $d_i = \sum_{j=1}^n w_{ij}$ and D the matrix with its (i, i) th diagonal entry equal to d_i . The *Laplacian* L of W can be defined as

$$L = I - D^{-1}W, \quad (1)$$

where I denotes the identity matrix. The segmentation is given by the eigenvector x_2 to the second smallest eigenvalue of L . Assuming that the nodes are sorted and the clusters do not overlap, it is a piece-wise constant vector [1, 3, 10]. The labelling can be obtain by *kmeans*, for instance.

3 An Upper Bound on the Eigenvalue Gap

This section introduces the first contribution of this work, namely bounds on the second smallest eigenvalue of the Laplacian if not all edge weights are defined.

Let w be a vector consisting of the entries of a particular row of W . Denote by w' the same vector with some yet not all of its entries set to zero. It is possible to establish upper and lower bounds on the angle between w and w' :

Lemma 1. *For the angle between w and w' we have*

$$0^\circ < \angle(w, w') < 90^\circ \quad (2)$$

Proof. Let $\mathcal{P}(w')$ denote the set of non-zero entries of w' . If a particular i is chosen such that $w'(i) \neq 0$, we have $w(i) \cdot w'(i) = w(i)^2$, hence we have for the scalar product between w and w'

$$\frac{w^T \cdot w'}{\|w\| \cdot \|w'\|} > 0. \quad (3)$$

Regarding the upper bound on the scalar product, we have

$$w^T \cdot w' = \sum_{i \in \mathcal{P}(w')} w'(i)^2 = \|w'\|^2. \quad (4)$$

we can see that Eq. (3) cannot attain a value of 1 since the first factor of the denominator is $\|w\|$ and we have $\|w\| > \|w'\|$. \square

These vectors w_i induced by the vertices $v \in V_k$, with either $k = 1$ or $k = 2$, can be regarded to span an n_k -dimensional subspace

$$\mathcal{S}_k = \text{span} \left(w_{i_1}, \dots, w_{i_{n_k}} \right). \quad (5)$$

Denote by the matrix S_k an orthonormal basis of \mathcal{S}_k . For the angle between w' and the corresponding subspace \mathcal{S}_k we obtain that

Lemma 2. $\angle(w', \mathcal{S}_k) > \angle(w, \mathcal{S}_k)$.

Proof. Noticing that $\angle(w, \mathcal{S}_k) = 0 \Leftrightarrow \left\| \frac{w^\top}{\|w\|} \cdot \mathcal{S}_k \right\| = 1$ we can show that $\left\| \frac{w^\top}{\|w\|} \cdot \mathcal{S}_k \right\| > \left\| \frac{w'^\top}{\|w'\|} \cdot \mathcal{S}_k \right\|$ as in the proof of Lemma 1. \square

Let w'_p be the vector with p of its entries being set to zero. From the above two lemmata, we immediately obtain that

Corollary 1. *The angle between w'_p and \mathcal{S}_k increases with the weight of the “zeroed” entries of w'_p , $\sum_{i \notin \mathcal{P}(w'_p)} w(i)^2$*

$$\angle(w'_1, \mathcal{S}_k) < \angle(w'_2, \mathcal{S}_k) < \dots \quad (6)$$

Notice that the *gap* between the two clusters – measured by the difference between the second and the third smallest eigenvalue of the Laplacian L – is maximum if and only if $\mathcal{S}_1 \perp \mathcal{S}_2$. Increasingly perturbed vectors $w'_i, i = 1, \dots, n_k$, thus cause increasingly perturbed subspaces \mathcal{S}'_k spanned by the vectors w'_i .

The question we are interested in is what happens to the angle between \mathcal{S}'_1 and \mathcal{S}'_2 compared with $\angle(\mathcal{S}_1, \mathcal{S}_2)$. If we are interested in exactly determining the gap, the convex hulls of the two sets of vectors w'_i need be compared.

However, analyzing the worst-case turns out to be much easier. This worst-case is defined by the *ex-radius*, ie. the largest distance between the centroid b_k of the points $w_i \in \mathcal{S}_k$ and the points w_i .

To derive an upper bound we need to determine the distance between w and w'_p . Obviously, the Euclidean distance between w and w'_p is largest if and only if c_1 entries are zeroed. Here, we further assume that $p < n_k$, ie. not all edges to other vertices of the same cluster are removed.

Theorem 1. *The squared Euclidean distance between w and w'_p is bounded by*

$$\|w - w'_p\|^2 \leq p \cdot c_1^2. \quad (7)$$

With probability $\rho_1 > \left(\frac{1}{1+\frac{n_2}{n_1}}\right)^p$ if $w \in \mathcal{S}_1$, and $\rho_2 > \left(\frac{1}{1+\frac{n_1}{n_2}}\right)^p$ if $w \in \mathcal{S}_2$, respectively, the distance $\|w - w'_p\|^2$ is smaller.

Proof. Since $c_1 > c_2$, the distance $\|w - w'_p\|^2$ is largest if and only if all zeroed edges have weight c_1 . The probability to sample c_1 entries of vectors $w \in \mathcal{S}_1$ is given by

$$\rho'_1 = \frac{n_1!}{p!(n_1-p)!} \cdot \frac{p!(n_1+n_2-p)!}{(n_1+n_2)!} = \frac{n_1!(n_1+n_2-p)!}{(n_1-p)!(n_1+n_2)!} \quad (8)$$

$$= \frac{n_1!}{(n_1-p)! \cdot \prod_{i=1}^p (n_1+n_2-p+i)} = \prod_{i=1}^p \frac{n_1-p+i}{n_1+n_2-p+i}. \quad (9)$$

Defining $d_i = n_1 - p + i$ we obtain

$$\rho'_1 = \prod_{i=1}^p \frac{d_i}{d_i + n_2} = \prod_{i=1}^p \frac{d_i}{d_i(1 + \frac{n_2}{d_i})} = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{d_i}} = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1-p+i}}. \quad (10)$$

Using an upper bound of each factor we finally arrive at the claim:

$$\rho'_1 = \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1 - p + i}} \quad (11)$$

$$< \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1 - p + p}} \quad (12)$$

$$= \prod_{i=1}^p \frac{1}{1 + \frac{n_2}{n_1}} = \left(\frac{1}{1 + \frac{n_2}{n_1}} \right)^p. \quad (13)$$

The derivation of the probability to only sample c_1 entries of vectors $w \in \mathcal{S}_2$ is equivalent. \square

In other words, theorem 1 implies that the probability that the distance $\|w - w'_p\|^2$ is smaller than $p \cdot c_1^2$ reduces exponentially as the number of zeroed entries p in w'_p grows. In the following, we call this distance the *separation* between \mathcal{S}_1 and \mathcal{S}_2 .

Let the centroids $b_1 = [c_1 \cdots c_1 \ c_2 \cdots c_2]$, $b_2 = [c_2 \cdots c_2 \ c_1 \cdots c_1]$, and a vector v parallel to $b_1 - b_2$ with $\|v\| = 1$. Assuming that p entries of each vector $w'_p \in \{\mathcal{S}_1, \mathcal{S}_2\}$ are zeroed where $p < \{n_1, n_2\}$, we arrive at

Theorem 2. *The separation between the two perturbed clusters equals $s = s_1 + s_2$ where*

$$s_k \geq \|(b_k^\top v) v\|^2 - p \cdot c_1^2, \quad (14)$$

and the angle equals $\alpha = \alpha_1 + \alpha_2$ where

$$\alpha_k \geq \tan^{-1} \frac{\sqrt{s_k}}{\|l\|} \quad (15)$$

with $l = b_1 - (b_1^\top v) v = b_2 - (b_2^\top v) v$.

Proof. The vector v indicates the line between the two cluster centroids b_k . If the orthogonal projection of the vector b_k onto v is subtracted from b_k , we obtain the vector l from the origin to the perpendicular point on the line between b_1 and b_2 .

The length of the line segment between one of the two centroids and the perpendicular point is given by the length of the orthogonal projection of the vector b_k onto v , ie. $\|(b_k^\top v) v\|^2$. Subtracting the radius of the sphere around b_k by lemma 1 yields the expression in Eq. (14).

Since the perpendicular line and the line between the perpendicular point and the sphere around each b_k form a right triangle, we can compute the angle between the perpendicular l and the vector between origin and the closest point on the sphere around b_k by Eq. (15). \square

The probability that the separation s is in fact larger than the minimum stated in theorem 2 is $1 - \rho_1 \rho_2$, ie. usually very large.

Let v'_1 and v'_2 be the vectors from the origin to the intersection of the line between b_1 and b_2 with the sphere around each cluster center. Let the first two rows of the $n \times n$ matrix T' be the vectors v'_1 and v'_2 , and the other rows being zero.

Let further v_1 and v_2 be the two 2d-vectors resulting from rotating $[\sqrt{2} \ \sqrt{2}]$ by $\frac{\alpha}{2}$ and $-\frac{\alpha}{2}$, respectively. Assume further that v_1 and v_2 are normalized such that $\sum_i v_1(i) = \sum_i v_2(i) = 1$. Let the 2×2 matrix T consist of v_1 and v_2 as first and second row.

Using theorem 2 we are now able to derive a bound on the second largest eigenvalue of the Laplacian. The following theorem constitutes the first of our two main contributions.

Theorem 3. *The second smallest eigenvalue λ_2 of the Laplacian matrix L is bounded by*

$$\lambda_2 \leq 1 - \lambda_{min} \quad (16)$$

where λ_{min} is the smaller of the two solutions $0 \leq \lambda_{min} \leq 1$ of the quadratic equation

$$(t_{11} - \lambda_{min})(t_{22} - \lambda_{min}) - t_{12}t_{21} = 0 \quad (17)$$

where the scalars t_{ij} are the entries of the matrix T .

Proof. We can see that the two eigenvalues of T and T' are identical: Obviously both share the eigenvalue 1. The second eigenvalue is also identical since $\alpha = \angle(v'_1, v'_2) = \angle(v_1, v_2)$, and the principal angle θ and the singular value and eigenvalue are related by

$$0 \leq \cos \theta = \sigma(v_1^\top v_2) = \lambda(|v_1^\top v_2|) \leq 1 \quad (18)$$

where $\sigma(\cdot)$ denotes the singular value of the argument and $\lambda(\cdot)$ the eigenvalue.

Lastly, as the eigenvalues of the Laplacian L are related to those of $D^{-1}W$ by $\Lambda(L) = 1 - \Lambda(D^{-1}W)$, we obtain the claim in Eq. (16). \square

4 Graph Completion

Let the edge $\tilde{e}_{i,j}$ between vertices v_i and v_j be unknown. Denote by $\mathcal{T}_u = (e_{i,\mathcal{L}_1}, e_{\mathcal{L}_1,\mathcal{L}_2}, \dots, e_{\mathcal{L}_{n_l},j})$ the u th path between v_i and v_j of length $n_l > 1$.

Proposition 1. *The unknown weight $\tilde{w}_{i,j}$ of an edge $e_{i,j}$ between vertices v_i and v_j can be inferred by*

$$\tilde{w}_{i,j} = \max_u \{ \min \{ w(e), e \in \mathcal{T}_u \} \}. \quad (19)$$

This is motivated by the following idea: Let v_i and v_j both be vertices of the same cluster \mathcal{S}_k . Then, there exists at least a single path $\mathcal{T}(i, j)$ such that all edges along this path have large weight. Assume, conversely, that $v_i \in \mathcal{S}_{k_1}$ and

$v_j \in \mathcal{S}_{k_2}$, $k_1 \neq k_2$. Then, all paths $\mathcal{T}(i, j)$ contain at least a single edge with low weight.

As the probability that a path contains an edge with low weight increases with the path length, it suffices to search the q shortest paths \mathcal{T}_u , $u = 1, \dots, q$, between v_i and v_j .

5 Application and Algorithm

In this section we present an algorithm for motion segmentation. Suppose that several sets of 3d-points move independently and are projected into images by a camera possibly also rotating and translating. Due to occlusion within the scene or failure of the feature point tracker, not every 2d-projection of a 3d-point is visible in all the images.

The problem is then to assign each trajectory – temporally consecutive 2d-projections of a particular 3d-point – a label indicating which group of 3d-points it belongs to. Associating one graph node with each trajectory, it requires to define affinities between each two vertices. If two trajectories do not overlap sufficiently such an affinity cannot be defined.

In the following we explain the procedure how to estimate affinities between nodes. It strongly rests upon the guided-sampling algorithm proposed in [12] yet only uses random sampling in a strict sense.

Low-dimensional subspaces are fitted to a subset of vertices representing the trajectories. Since subspace fitting is susceptible to missing data, a random vertex is selected first. We then discard vertices that are not visible at exactly those images the first vertex is visible at. From this subset, $F - 1$ vertices are randomly chosen and the model parameters are computed by SVD. Here, we define $F > D$.

Given this model, the error is computed for *all* vertices which are visible in at least 8 of the images the subspace model is valid for. As error measure we use the Euclidean distance between each visible trajectory and the subspace. The resulting error is appended to a residual matrix. If a 3d-point is not visible in at least 8 of the images used for estimating the subspace, the corresponding entry in the residual matrix is set to undefined.

These steps are repeated R times. The error matrix is then sorted similarly as in [12]. The difference to [12] is that undefined entries are discarded at the sorting. If error vectors corresponding to two vertices v_i and v_j share the same b models among those H with lowest error, the weight of the edge between vertices v_i and v_j is set to $e_{ij} = b/H$. The scalar parameter H controls the connectivity of graph.

Finally, weights of edges which could not be defined are inferred by using the algorithm proposed in Sec. 4. The resulting complete graph is segmented using NCut spectral clustering.

6 Experiments

6.1 Synthetic Data

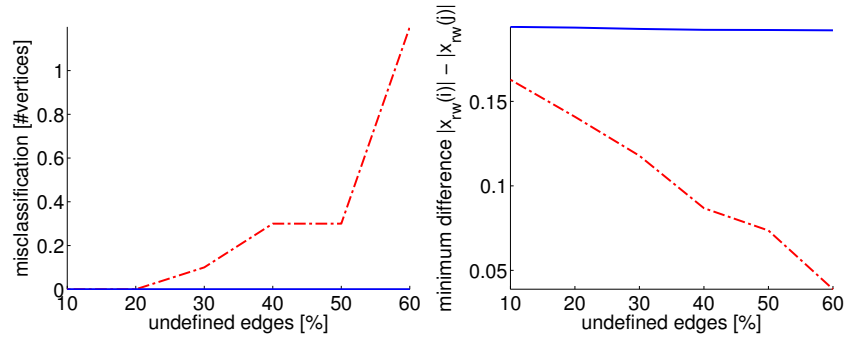


Fig. 1. Dash-dotted red line: spectral clustering (undefined edge weights are set to zero). Solid blue line: proposed graph completion followed by NCut. Left plot: the number of incorrectly classified graph vertices. Right plot: minimum absolute sum between any two of the entries $|x_2(i)|$ and $|x_2(j)|$ of x_2 . i and j indicate vertices of different clusters. Larger values of the minimum absolute sum indicate better separability of the two clusters.

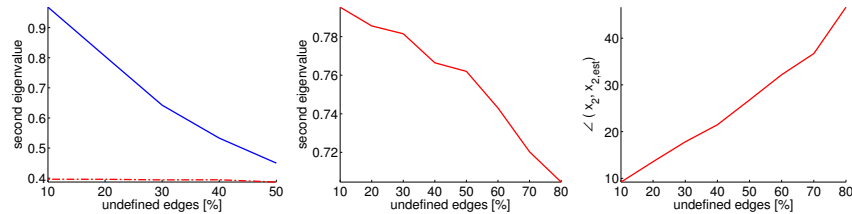


Fig. 2. Left plot: The solid blue line indicates the predicted upper bound on the eigenvalue corresponding to the second smallest eigenvalue of the Laplacian while the dash-dotted red line indicates the true value. The noise was fixed to 2% and $c_1 = 0.8$, $c_2 = 0.2$. Middle plot: The solid red line shows the measured second eigenvalue for $c_1 = 0.6$ and $c_2 = 0.4$. Right plot: The solid line shows the angle between the corresponding eigenvector and the ground truth.

The algorithm proposed in Sec. 5 was evaluated using an artificial graph consisting of $n_1 = 50$ and $n_2 = 50$ nodes. While the data is ordered, neither the algorithm proposed in Sec. 5 nor the spectral clustering have any knowledge about the labeling.

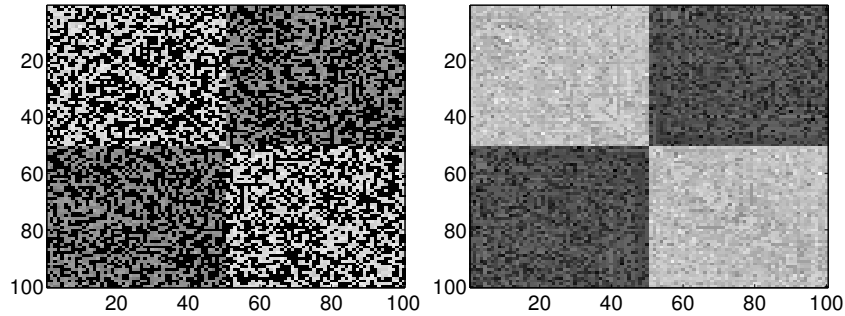


Fig. 3. The weight matrix W shown in the left image was created by taking $c_1 = 0.6$, $c_2 = 0.4$, setting the noise level to a standard deviation 3% and randomly removing 60% of all edges. The right image shows the recovered graph by the proposed algorithm. The recovered eigenvector to the second smallest eigenvalue of the Laplacian L differed from the ground truth by approximately 2.76° whereas that vector corresponding to the right image differed by about 36.87° .

The effect which undefined but “zeroed” edge weights have on the spectral clustering can be seen by the dash-dotted red curve in the left plot of Fig. 1. Here, for fixed, normally distributed noise applied to the non-zero entries of W , fixed separation $c_1 - c_2$ and gradually increasing percentage of zeroed weights, the misclassification (measured by the number of misclassified vertices) increases (shown is the average of ten trials with different noise).

For this experiment, noise and separation were fixed to $\sigma = 2\%$ and $c_1 = 0.7$, $c_2 = 0.3$, respectively. The dash-dotted red lines indicate the results using a traditional spectral clustering, ie. the entries of the weight matrix W are simply set to zero. The solid blue lines indicate the results after the proposed graph completion followed by the spectral clustering.

The right plot in Fig. 1 shows the minimum

$$\min |x_2(i)| - |x_2(j)|, \quad \forall i, j \quad (20)$$

of the eigenvector x_2 to the second smallest eigenvalue of L between any two of its entries $|x_2(i)|$ and $|x_2(j)|$ where i and j indicate vertices of different clusters. Larger values indicate better separability of the two clusters.

As can be seen, the amount of undefined edge weights decreases the separation between the two clusters for standard spectral clustering (dash-dotted red line). The proposed graph completion is not affected that strongly.

The left plot of Fig. 2 shows the average (of ten trials) of the predicted upper bound on the eigenvalue (solid blue line) corresponding to the second smallest eigenvalue of the Laplacian while the dash-dotted red line indicates the true value. The noise was fixed to 2% and $c_1 = 0.8$, $c_2 = 0.2$. It can be seen that for increasing noise both the theoretical upper bound and the measured values decrease. The middle plot in the same figure shows the average of the measured

second eigenvalue for $c_1 = 0.6$ and $c_2 = 0.4$. Apparently, the eigenvalue decreases more strongly. The right plot shows the angle between the corresponding eigenvector and the ground truth. This perturbation causes the misclassification.

The left image of Fig. 3 shows an example of a weight matrix W if 60% of all edges are randomly removed. The right image in the same image shows the graph recovered by the proposed algorithm.

The three plots in Fig. 4 show averages of the Frobenius norm between the ground truth weight matrix and the recovered one of ten trials with different, normally distributed noise each time. In the left plot, the standard deviation was increased from 0 to 0.03 in steps of 0.005. The solid blue line indicates 10% randomly removed edges, the dash-dotted red line 30%, and the dashed green one 60%.

In the middle plot, the percentage of missing edges was increased. Here, the solid blue line was obtained by setting $c_1 = 1$ and $c_2 = 0$, the dash-dotted red line $c_1 = 0.8$ and $c_2 = 0.2$; the dashed green line $c_1 = 0.6$ and $c_2 = 0.4$. The noise was kept fixed to 1%. It can be seen that the amount of missing edges is more important than the difference between c_1 and c_2 .

The right figure corresponds to an experiment where the difference between c_1 and c_2 was decreased. The solid blue line indicates 10% randomly removed edges, the dash-dotted red line 30%, and the dashed green one 60%. The noise was kept fixed to 1%. Since all of the three lines are relatively constant while $c_1 - c_2$ varies we can conclude that the determining factor is the amount of undefined edge weights.

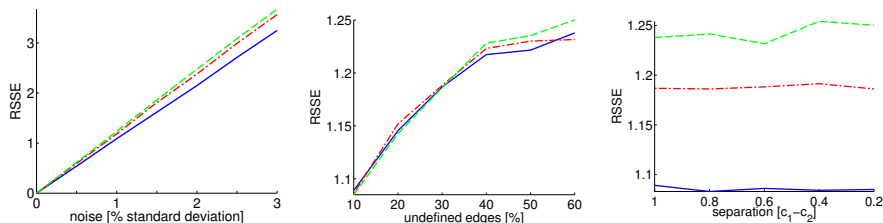


Fig. 4. Each of the three plots was obtained by varying a particular parameter: for the left plot, the noise was increased; edges were randomly removed for the middle one; the difference between c_1 and c_2 was decreased for the right plot. As error measure we used the Frobenius norm between the recovered weight matrix and the ground truth (root of sum of squared errors, RSSE). For the definitions of the different lines of each plot please see the explanation in Sec. 6.1.

6.2 Real Image Experiments

Two real image sequences, *cars9* and *farm01*, were selected from the extended Berkeley motion database³. These sequences were selected because they are

³ <http://lmb.informatik.uni-freiburg.de/resources/datasets>

longer (60 and 252 images, respectively) so not all trajectories co-occur at a given number of frame. That also is the reason why the sequences from the popular Hopkins benchmark were not considered here as they simply do not contain sufficiently many missing correspondences.

The software from [6]⁴ was used for tracking feature points. Tracks shorter than 8 images were discarded as the subspace distance measure is not reliable then.

For the *cars9* sequence 80% of the theoretically possible edges can be defined. The remaining ones correspond to trajectories that do not co-occur at sufficiently many images. For the *farm01* sequence, 40% of the trajectories do not overlap sufficiently.

Figures 5 and 6 show segmentation results on the *farm01* and the *cars9* sequence for a state-of-the-art algorithm for motion segmentation (left column) [6], the proposed algorithm for motion segmentation *without* graph completion before the spectral clustering (middle column), and the proposed algorithm *with* graph completion followed by spectral clustering (right column).

Differently than the algorithm of [6], the proposed algorithm does not merge multiple oversegmentations.

As can be seen, the proposed graph completion greatly improves segmentation results. The wheels of the tractor in the *farm01* sequence are somewhat mis-segmented as they define a separate rigid motion in terms of the rigid subspace measure as affinity.

7 Discussion

This article investigated the effect of undefined edge weights on spectral graph clustering. Upper bounds on the squared distance between two clusters were derived. It was possible to establish a lower bound on the second smallest eigenvalue of the Laplacian.

A practical algorithm was proposed to infer undefined edge weights. Its performance was evaluated using synthetic data. Using challenging sequences from a standard benchmark, it was shown that the proposed method outperforms both a state-of-the-art algorithm for motion segmentation and a spectral clustering without the graph completion.

Up to the best knowledge of the authors, this is the first work which considers the effect of undefined edges on the performance of spectral clustering.

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE TPAMI* **22** (2000) 888–905
2. Luxburg, U.: A tutorial on spectral clustering. *Statistics and Computing* **17** (2007) 395–416

⁴ The software is provided at <http://lmb.informatik.uni-freiburg.de/resources/binaries/>.

3. Maila, M., Shi, J.: A random walks view of spectral segmentation. In: AISTATS 2001. (2001)
4. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: Advances in Neural Information Processing Systems (NIPS). (2006) 1601–1608
5. Agarwal, S., Branson, K., Belongie, S.: Higher order learning with graphs. In: ICML. ICML '06, New York, NY, USA, ACM (2006) 17–24
6. Ochs, P., Brox, T.: Higher order motion models and spectral clustering. In: CVPR, Washington, DC, USA, IEEE Computer Society (2012) 614–621
7. Purkait, P., Chin, T.J., Ackermann, H., Suter, D.: Clustering with hypergraphs: the case for large hyperedges. In: ECCV. (2014)
8. van Gennip, Y., Hunter, B., Ahn, R., Elliott, P., Luh, K., Halvorson, M., Reid, S., Valasik, M., Wo, J., Tita, G.E., Bertozzi, A.L., Brantingham, P.J.: Community detection using spectral clustering on sparse geosocial data. *SIAM Journal of Applied Mathematics* **73** (2013) 67–83
9. Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV. ECCV'10, Berlin, Heidelberg, Springer-Verlag (2010) 282–295
10. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS. (2001) 849–856
11. Balakrishnan, S., Xu, M., Krishnamurthy, A., Singh, A.: Noise thresholds for spectral clustering. In: Advances in Neural Information Processing Systems (NIPS). (2011) 954–962
12. Chin, T.J., Yu, J., Suter, D.: Accelerated hypothesis generation for multistructure data via preference analysis. *TPAMI* **34** (2012) 625–638

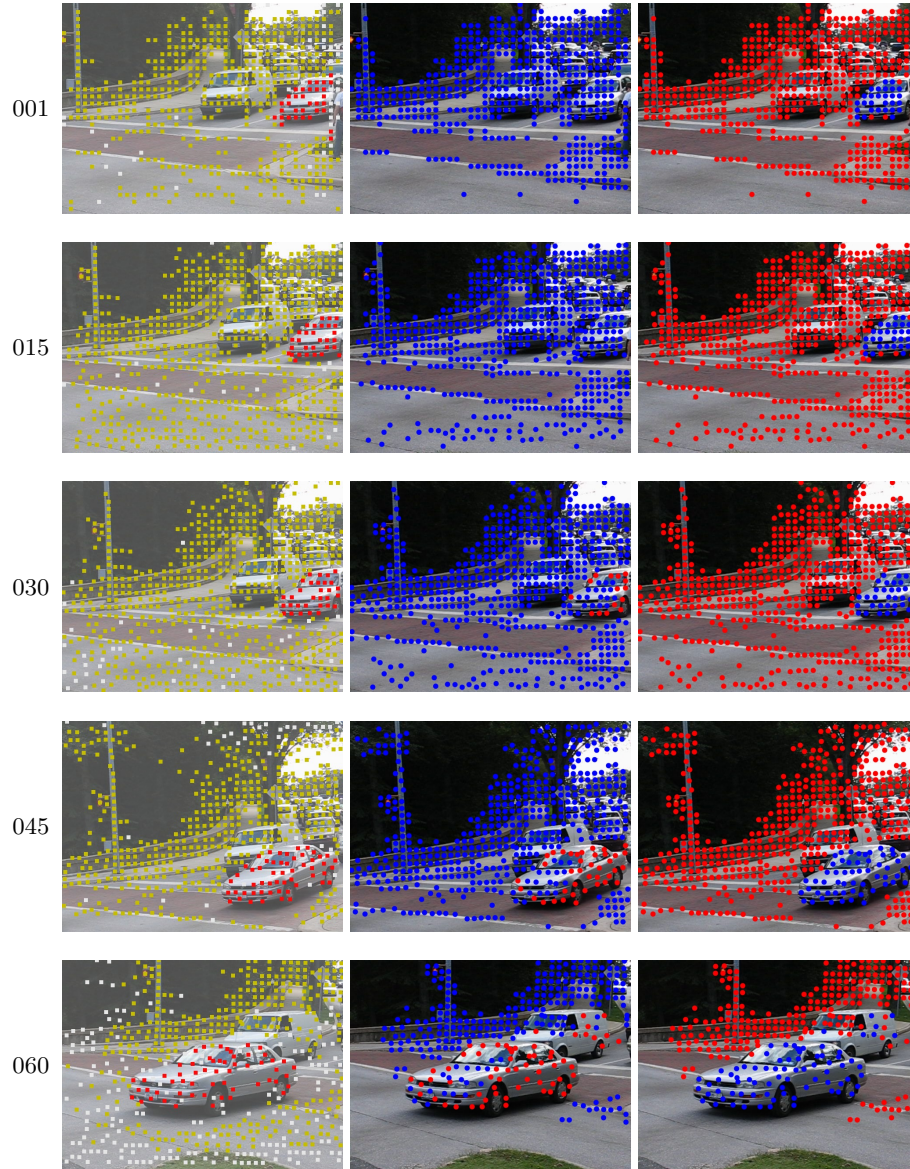


Fig. 5. *cars9* sequence from the extended Berkeley motion database. Images in the left column indicate segmentation results by the algorithm of [6]; middle column: spectral clustering followed by spectral clustering *without* graph completion before; right column: proposed (graph completion followed by spectral clustering)

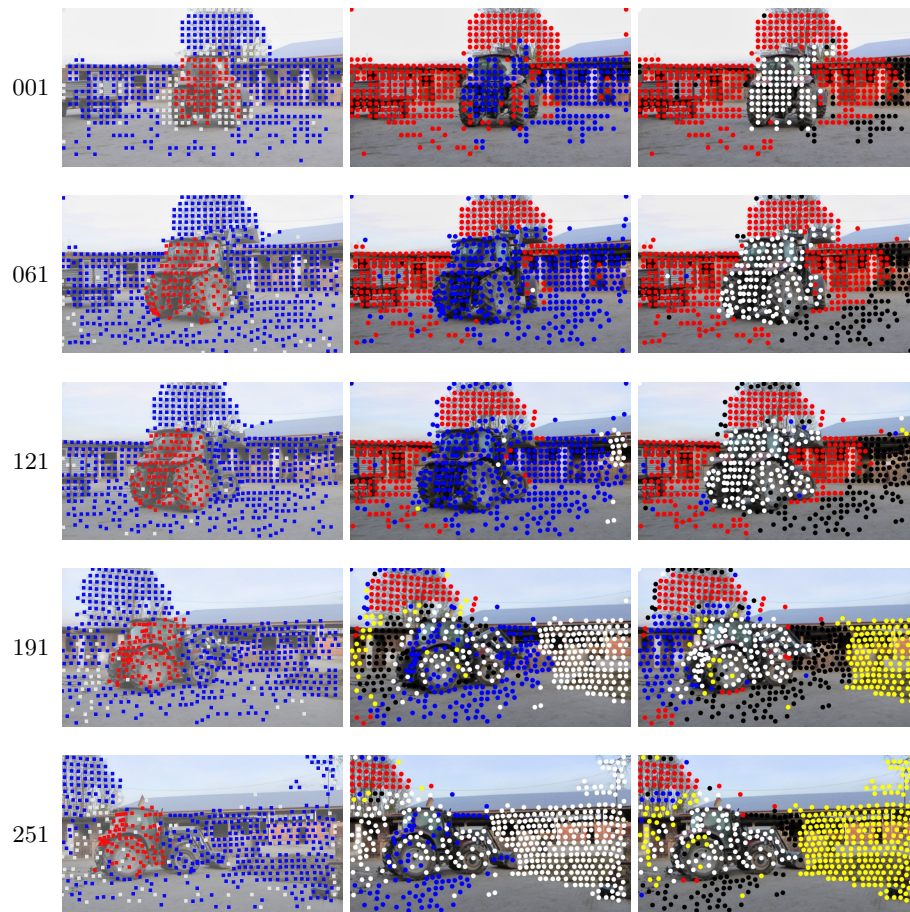


Fig. 6. *farm01* sequence from the extended Berkeley motion database. Images in the left column indicate segmentation results by the algorithm of [6]; middle column: spectral clustering followed by spectral clustering *without* graph completion before; right column: proposed (graph completion followed by spectral clustering)