Co-Clustering of Image Segments Using Convex Optimization Applied to EM Neuronal Reconstruction

Shiv N. Vitaladevuni
Janelia Farm Research Campus,
Howard Hughes Medical Institute
vitaladevunis@janelia.hhmi.org

Ronen Basri
Dept. of Computer Science and Applied Math,
Weizmann Institute of Science

Abstract

This paper addresses the problem of jointly clustering two segmentations of closely correlated images. We focus in particular on the application of reconstructing neuronal structures in over-segmented electron microscopy images. We formulate the problem of co-clustering as a quadratic semi-assignment problem and investigate convex relaxations using semidefinite and linear programming. We further introduce a linear programming method with manageable number of constraints and present an approach for learning the cost function. Our method increases computational efficiency by orders of magnitude while maintaining accuracy, automatically finds the optimal number of clusters, and empirically tends to produce binary assignment solutions. We illustrate our approach in simulations and in experiments with real EM data.

1. Introduction

We address the problem of jointly clustering two segmentations of an image or two closely correlated images, so as to ensure good matching between the clusters in the two segmentations. We pose this as an optimization problem and apply our approach to neuronal reconstruction from electron microscopy (EM) images.

Figures 1 illustrates the application with example EM images along with their ground-truth segmentation and the automatic segmentation that needs to be clustered. Notice the considerable deformation of cell boundaries across sections and the over-segmentation in the automatic segments. Our goal is to jointly cluster the automatic segments based on their likelihood to belong to the same neuron.

We refer to the joint grouping of segments in two segmentations as co-clustering, and formulate this as a quadratic optimization problem, specifically a Quadratic Semi-Assignment Problem (QSAP). Coefficients in the quadratic function encode whether pairs of segments should be placed in the same cluster or in distinct clusters.

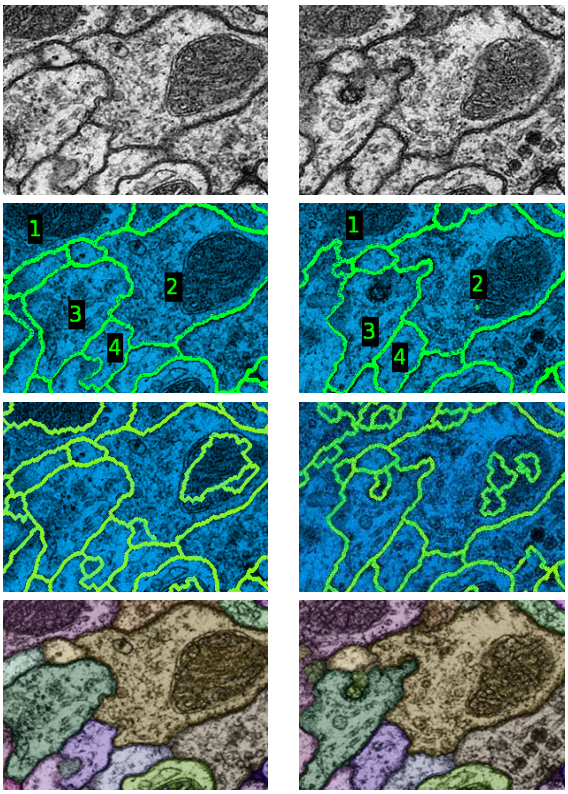


Figure 1. Row 1: Two consecutive sections of a neuronal tissue acquired using an electron microscope. Row 2: Ground-truth segmentation. Each segment corresponds to a region belonging to a distinct cell. The numbers show matches between segments of the same cell in the two sections. Row 3: Input automatic segmentation - note the over-segmentation relative to the ground-truth. Our objective is to cluster the automatic segments to recover the ground-truth. Row 4 shows the output of the proposed approach: colors indicate mapping between segments.

QSAP is NP-hard in general [10]. To make the problem tractable, we study two relaxations based on Semi-Definite Programming (SDP) and Linear Programming (LP), using the work of Charikar *et al.*[4]. The linkage application commonly requires solving systems with more than 2000 seg-

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

ments; we found the SDP solution to be impractical for such large systems. The LP formulation proposed in [4] also proved to be impractical for large systems. This formulation involves enforcing triangular inequalities, requiring therefore $O(n^3)$ constraints, where n is the number of segments. This results in billions of inequalities for typical systems. We address this by enforcing only the local metric constraints, bounding the number of constraints to $O(n^2)$. We found that this greatly reduces the time-complexity of the optimization without significantly affecting the accuracy.

We present an approach to learning the coefficients of the optimization function using a boosted classifier trained to minimize the Rand index error of the classification. This opens up the possibility of applying the results to other co-clustering problems in vision.

Computing globally optimal binary solutions from the real-valued solutions obtained from the SDP and LP is NP-hard. We present empirical results of LP's integrality, and theoretical discussions about the relation of the integrality of our LP solution to studies in optimization theory.

We compare the performance of the LP formulation with the SDP version, normalized cuts, and connected components with learned cost function. Experiments indicate the utility of the proposed approach.

2. Co-clustering problem

Let $\Omega \subset \Re^2$ denote a discrete 2D domain of points (pixels), and let P and Q denote two segmentations of Ω; $P = \{p_i\}_{i=1}^N$ and $Q = \{q_i\}_{i=1}^M$ are two non-overlapping partitions of Ω ($p_i \cap p_j = \emptyset$, $q_k \cap q_l = \emptyset$ and $\cup_i p_i = \cup_j q_j = \Omega$). Such segmentations can be obtained by applying two different segmentation algorithms to an image or by segmenting two closely correlated images. Our objective is to cluster the segments in P and Q simultaneously, so as to obtain a good match between clusters in P and Q. We further assume that P and Q are over-segmentations of Ω, and hence we do not need to consider splitting a segment to achieve a good match.

Let $\mathcal{C}$ be the set of clusters, such that each cluster $C \in \mathcal{C}$ is a subset of $P \cup Q$. We associate a cost with $\mathcal{C}$ defined as

$$H(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{r,r' \in C} F(r, r') \qquad (1)$$

with $F(r, r') \in \mathbb{R}$ and express $H(\mathcal{C})$ as a quadratic function as follows. Let $\mathbf{x}_C \in \{0,1\}^N$ denote a vector indicating which of the segments in P belongs to cluster C. Similarly, let $\mathbf{y}_C \in \{0,1\}^M$ be the indicator vector for segments in Q that belong to C. Let $\mathbf{z}_C \in \{0,1\}^{N+M}$ be a vector obtained by concatenating $\mathbf{x}_C$ and $\mathbf{y}_C$ (i.e., $\mathbf{z}_C = [\mathbf{x}_C^T \; \mathbf{y}_C^T]^T$). $H(\mathcal{C})$ is then expressed by a general quadratic function on pairs of elements in the clusters, as $H(\mathcal{C}) = \sum_{C \in \mathcal{C}} \mathbf{z}_C^T \; F \; \mathbf{z}_C$. Our objective is to compute the clustering that minimizes the cost $H(\mathcal{C})$, stated as

$$\begin{aligned} &\text{Min}: \; H(\mathcal{C}) = \sum_{C \in \mathcal{C}} \mathbf{z}_C^T \; F \; \mathbf{z}_C \\ &\text{s.t.} \; \; \mathbf{z}_C \in \{0,1\}^{N+M} \; \text{ and } \; \sum_C (\mathbf{z}_C)_i = 1 \; \forall i, \end{aligned} \qquad (2)$$

where the constraints ensure that each segment in P and Q participates in exactly one cluster. Note that while we focus here on co-clustering, the formulation and its solutions are applicable to the general problem of clustering as well.

When the number of clusters $|\mathcal{C}|$ is known, the optimization (2) is equivalent to finding the minimal k-cut in a weighted graph $\mathcal{G}$ with the edge weight matrix set to $-F$. Here, each cluster corresponds to a component in $\mathcal{G}$ and $k = |\mathcal{C}|$ is the known number of clusters. For the equivalence, notice that when the edge matrix is set to $-F$, Min k-cut minimizes $\sum_C (\mathbf{1} - \mathbf{z}_C)^T (-F) \mathbf{z}_C$, which is the same as minimizing $\mathbf{z}_C^T F \mathbf{z}_C$, as all the vertices in the graph are forced to belong to exactly one component in $\mathcal{G}$.

While the problem of finding the minimal k-cut with non-negative weights and with a fixed number of clusters has a polynomial solution (with complexity $O(n^{k^2})$), the problem of finding the minimal cut with negative weights or with an unknown number of clusters is NP-hard [7]. More generally, this problem can be viewed as a Quadratic Semi-Assignment Problem (QSAP) which is also NP-hard [10]. The focus of our work is to construct tractable relaxations that exhibit good empirical performance in the EM reconstruction application. An SDP relaxation, along with an LP relaxation based on Charikar *et al.* [4], form the starting point for our work.

Image segmentation has been addressed with graph theoretic algorithms, most notably with Shi and Malik's Normalized-cuts [14]. Yu and Shi modified N-cuts to handle negative weights and used their modification to model "pop-out" in perceptual grouping [17]. Cosegmentation of an object occuring in two images has been studied in [12] and [11].

3. Convex Relaxations

DeBie and Cristianini presented a fast SDP relaxation of N-cuts and Max-cut [2]. Also, Xing and Jordan study SDP relaxation of k-Ncuts [16]. QSAP is a special case of Quadratic Assignment Problem (QAP). Many problems, such as the traveling salesman problem and graph partitioning, can be formulated as QAP, which is also NP-hard and difficult to even approximate [10]. Schellewald et al. study convex relaxations of QAP for feature matching in [13]. A closely related work of Kumar *et al.* deals with convex relaxations of quadratic optimization problems [9].

3.1. Semi-Definite Program (SDP) relaxation

Let $Z = [\mathbf{z}_1 \dots \mathbf{z}_{|\mathcal{C}|}]$, and $R = ZZ^T$, Z is $(N+M)\times|\mathcal{C}|$ and R is $(N+M)\times(N+M)$. Here, R is akin to the cluster co-occurrence matrix; it is 1 when two segments belong to the same cluster and 0 otherwise. In the spirit of Goemans and Williamson's approximation algorithm for the maximum cut [6] we relax R to obtain real-valued entries and restate (2) as a Semi-Definite Program (SDP):

$$\begin{aligned} \text{Min}: \quad & \text{Trace}(F^T R) \\ \text{s.t.}: \quad & R \succeq 0,\ R_{ij} \in [0,1]\ \forall i,j, \text{ and } R_{ii} = 1\ \forall i. \end{aligned} \tag{3}$$

(Note that we assume that F has negative weights, or else the identity matrix would form a trivial solution.) The relaxed problem is convex for arbitrary F, and it is possible to compute a globally optimal real-valued solution. Another advantage is that the optimization does not require apriori knowledge of the number of clusters.

It is generally NP-hard to convert the obtained real-valued solution to a globally optimal binary solution. A common heuristic to obtaining a binary solution [6] is to factorize the optimal R to QQ^T using Cholesky decomposition and then convert Q to binary values, e.g., by assigning 1 in each row to the coordinate with the maximal value in that row. We refer to this post-processing approach by SDP-fact. Another possibility is to consider R as an adjacency matrix of a weighted undirected graph. Clustering is then computed by finding connected components in a sub-graph obtained by considering edges with weights above a threshold. This post-processing approach is referred to as SDP-thresh. We pursued both approaches in our experiments.

We observed that current implementations of SDP solvers, including SDPT3 [15], SeDuMi and SDPLR [3], have difficulty solving problems constructed for the EM application when the number of variables was in the 1000's, i.e. $N + M \approx 2000$. The theoretical computational complexity of general SDP solvers is high, growing as $O((\sharp\text{var})^2(\text{size SDP})^{2.5})$ [2]. In practice, the solvers improve their efficiency by exploiting the structure of specific problems. However, current solvers find our application challenging.

3.2. Linear Program (LP) relaxation

Given the practical issues with SDP, we use a Linear Programming relaxation based on [4]. Notice that R is a matrix of inner products between vectors, $R_{ij} = \mathbf{v}_i \cdot \mathbf{v}_j$ where the vectors $\mathbf{v}_i = [\mathbf{Z}_{iC}]_{C\in\mathcal{C}}$ populate the rows of Z and indicate for each segment i its cluster association. Let $d_{ij} = \frac{1}{\sqrt{2}}\|\mathbf{v}_i - \mathbf{v}_j\|_2$, where $\sqrt{2}d_{ij}$ denotes the ℓ_2 distance between $\mathbf{v}_i$ and $\mathbf{v}_j$ in the cluster space. Since we require $\|\mathbf{v}_i\| = 1$ for all i, we have

$$d_{ij}^2 = \frac{1}{2}\left(\|\mathbf{v}_i\|_2^2 + \|\mathbf{v}_j\|_2^2 - 2\mathbf{v}_i\cdot\mathbf{v}_j\right) = 1 - R_{ij}. \tag{4}$$

Let $D = [d_{ij}]$; our objective then is to maximize $\text{Trace}(F^T D)$. We relax the entries in D from being ℓ_2 distances to being a metric, i.e., nonnegative, symmetric and follow the triangular inequality. This constraint is valid because a clustering induces an equivalence relation, and hence by transitivity $d_{ik} = d_{kj} = 0$ implies that also $d_{ij} = 0$ while if $d_{ik} + d_{kj} \in \{1, 2\}$ the triangular inequality is satisfied trivially. The LP formulation then is

$$\begin{aligned} \text{Max}: \quad & \textstyle\sum_{i,j} F_{ij} d_{ij} \\ \text{s.t.}: \quad & 0 \le d_{ij} \le 1\ ,\ d_{ij} = d_{ji}\ \forall i,j,\ d_{ii} = 0\ \forall i, \\ & \text{and } d_{ij} \le d_{ik} + d_{kj}\ \forall i,j,k. \end{aligned} \tag{5}$$

After optimizing this cost function we post-process the solutions by thresholding the matrix D and considering the connected components obtained by eliminating edges with weights exceeding a specified threshold.

Although the metric constraints used with the LP formulation seem to be weaker than the SDP formulation, we will show empirically cases in which the LP optimization performs better than the SDP. Note that while in general SDP problems may be expressed as linear programs with infinite number of constraints [8], this proposed LP formulation has only a finite number of constraints. However, the number of triangular inequalities in the LP grows as $O(n^3)$ where n is the number of variables. For a typical EM co-clustering problem with 2000 variables, we will have more than $3 \times 8 \times 10^9$ inequalities! We found this to be impractical – we tested this using lp_solve, a state-of-the-art open source solver. We therefore seek to reduce the number of constraints while maintaining empirical performance.

3.3. LP-R - LP with local metric constraints

We further relax our LP formulation by constraining the distances in D to satisfy the triangular inequality only locally. Consider a graph $\mathcal{G}_P$ consisting of nodes corresponding to segments, $p_i \in P$, and edges connecting spatially adjacent segments. Similarly, a graph $\mathcal{G}_Q$ is defined for segmentation Q. Let $\mathcal{G}$ be a graph constructed by combining $\mathcal{G}_P$ and $\mathcal{G}_Q$ and adding edges between overlapping segment pairs $p_i \in P$ and $q_j \in Q$. The distances in D are constrained to be locally metric within all cliques in $\mathcal{G}$. Let $E = \{e_{ij}\}$ be the set of edges in $\mathcal{G}$. The reduced LP version, referred to as LP-R, is

$$\begin{aligned} \text{Max}: \quad & \textstyle\sum_{i,j} F_{ij} d_{ij} \\ \text{s.t.} \quad & 0 \le d_{ij} \le 1\ ,\ d_{ij} = d_{ji}\ \forall i,j,\ d_{ii} = 0\ \forall i, \text{ and} \\ & d_{ij} \le d_{ik} + d_{kj}\ \forall e_{ij}, e_{ik} \text{ and } e_{jk} \in E. \end{aligned} \tag{6}$$

The number of constraints in LP-R is 3 times the number of 3-cliques in $\mathcal{G}$. Let us first count the number of 3-cliques within $\mathcal{G}_P$ and $\mathcal{G}_Q$ separately. The segments in P and Q are constrained to be connected regions in the image

plane. Therefore, $\mathcal{G}_P$ and $\mathcal{G}_Q$ are planar graphs, and hence the number of 3-cliques within $\mathcal{G}_P$ is $O(n)$, and similarly for $\mathcal{G}_Q$. For counting the 3-cliques that extend across $\mathcal{G}_P$ and $\mathcal{G}_Q$, consider without loss of generality 3-cliques with two nodes in $\mathcal{G}_P$ and one in $\mathcal{G}_Q$. There are $O(n)$ edges in $\mathcal{G}_P$, and each adjacent pair can be simultaneously adjacent to $O(n)$ nodes in $\mathcal{G}_Q$. Therefore, the number of such cliques is bounded by $O(n^2)$. Thus, the number of inequalities in LP-R grows as $O(n^2)$. We observed empirically that the number of inequalities grows almost linearly with respect to the number of segments.

3.4. Non-integrality of LP solutions

Our goal in clustering is to assign exactly one cluster to every segment from $P \cup Q$. We therefore want our optimization to yield binary solutions. The LP formulation, however, only limits the variables to lie within the $[0, 1]$ range and a binary solution is not guaranteed.

One way to demonstrate that an LP system achieves binary solutions is to examine the vertices of the feasibility polytope. If all of the vertices are binary then a binary optimal solution must exist. To examine the vertices, consider the set of hyperplanes that compose the inequality constraints in our LP system (5). Denote these hyperplanes by $Ad = b$. Every vertex in the polytope lies at the intersection of a subset of these hyperplanes. A sufficient condition for the vertices to be binary is if the matrix A is *totally unimodular* (TUM), that is, if the determinant of every square subset of the rows and columns of A belongs to $\{-1, 0, 1\}$, and if b is integer.

Unfortunately, our constraint matrix is *not* totally unimodular. The matrix A can be viewed as the node-hyperedge incidence matrix of a directed hypergraph $\mathcal{H}$. The nodes in $\mathcal{H}$ correspond to the variables d_{ij} (columns of A), and each row of A that corresponds to a triangular inequality constraint represents a hyperedge with one head and two tails. For example, the inequality $d_{12} + d_{13} \geq d_{23}$ forms a directed hyperedge connecting the tail nodes d_{12} and d_{13} to the head node d_{23}. Such a hypergraph is called 2-LDH (Leontief Directed Hypergraph). Coullard and Ng [5] showed that the incidence matrix of a 2-LDH $\mathcal{H}$ is totally unimodular if and only if there exists no odd pseudo-cycle in $\mathcal{H}$. (A pseudo-cycle in $\mathcal{H}$ is a sequence of nodes where each node is connected to its successor in the cycle by a distinct hyperedge and that contains at least one tail-tail arc. An odd pseudo-cycle is a pseudo-cycle with odd number of tail-tail arcs.) We next construct an odd pseudo-cycle and use this construction to find a vertex with non-binary values.

Consider a system with 4 segments, denoted 1, 2, 3, and 4, and denote the unknown distances between those segments by d_{ij}, $1 \leq i, j \leq 4$. Consider a vertex of the feasibility polytope obtained by the following subset of constraints:

$$\tilde{A}\tilde{d} = \begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_{12} \\ d_{13} \\ d_{14} \\ d_{23} \\ d_{24} \\ d_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

$\tilde{A}$ is a sub-matrix of A with the first three rows representing triangular inequalities between subsets of the six variables and the last three rows representing the constraints that limit variables to take values ≤ 1. Indeed, the cycle $d_{12} \rightarrow d_{13} \rightarrow d_{14} \rightarrow d_{12}$ forms a 3-pseudo-cycle (and hence the determinant of the top-left 3×3 sub-matrix of $\tilde{A}$ is -2) implying that A is not totally unimodular. Consequently these six constraints intersect at a non-binary point $\langle d_{12} = d_{13} = d_{14} = 1/2, d_{23} = d_{24} = d_{34} = 1 \rangle$. In fact, the matrix A contains many more odd pseudo-cycles and hence, many vertices with rational values exist. Still, our simulations and experiments with real data demonstrate that typically we can find solutions in which many of the variables obtain binary values.

4. Learning the cost function

Quadratic cost functions can be used to encode a variety of criteria for clustering. For example, in Section 5 we demonstrate properties of our algorithm using a cost function that relies on the shape of the segments. Specifically, we construct a function that seeks clusters of segments in P and Q whose symmetric difference is minimal. Likewise, criteria that involve intensities and texture can also be incorporated. Furthermore, for a variety of applications it may make sense to learn the cost function from examples. In this section we describe a simple procedure for learning the cost function from training data. This procedure is later used in Section 6 to link regions corresponding to neuronal cells across EM sections.

Our objective is, given training data, to learn the components of F from properties of the segments in P and Q. In training, we assume that we are given the results of two automatic segmentations $P = \{p_i\}$ and $Q = \{q_i\}$ along with the ground-truth clustering, i.e., the desired partition of the set of segments $P \cup Q$. Let R^{gt} denote the co-occurrence matrix for the ground-truth clustering (as in Section 3.1, $R^{gt}_{ij} = 1$ if segments i and j belong to the same ground truth cluster and zero otherwise). Let R denote the co-occurrence matrix computed using a co-clustering formulation. To evaluate the accuracy of R we use the Rand index, which has been used extensively for evaluating clustering and segmentation algorithms, e.g. [1]. Denote respectively by $l_{gt}(r)$ and $l_{cc}(r)$ the cluster assigned to a segment r in the ground truth and in the computed co-clustering. The Rand index counts the number of false merges and splits in

the computed co-clustering as follows.

$$\sum_{r,r' \in P \cup Q} \mathbf{1}\left(l_{gt}(r) = l_{gt}(r') \wedge l_{cc}(r) \neq l_{cc}(r')\right) + \mathbf{1}\left(l_{gt}(r) \neq l_{gt}(r') \wedge l_{cc}(r) = l_{cc}(r')\right). \quad (7)$$

where $\mathbf{1}(.)$ denotes binary indicator function and $\wedge$ is the logical *and* operator. This measure can be expressed in terms of the co-occurrence matrices R^{gt} and R as follows.

$$\begin{aligned} \text{Rand}(R) &= \textstyle\sum_{i,j}[1 - R_{ij}]R^{gt}_{ij} + \sum_{i,j} R_{ij}[1 - R^{gt}_{ij}] \\ &= \textstyle\sum_{i,j}[1 - 2R^{gt}_{ij}]R_{ij} + c, \end{aligned}$$

where $c = \sum_{ij} R^{gt}_{ij}$ is a constant independent of R. Compare this with the SDP and LP formulations (3) and (5), if F was defined as $1 - 2R^{gt}$, minimizing the Rand error can be posed as a QSAP problem. We therefore seek a function that assigns to F_{ij} the values -1 and $+1$ to pairs of to-be-linked and not-to-be-linked segments in $P \cup Q$ respectively.

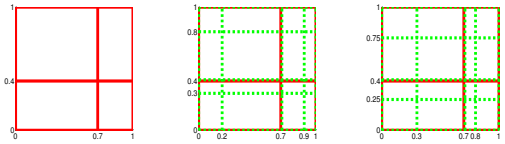
There have been numerous studies on trainable clustering and segmentation algorithms. Here we use a boosted classifier to learn the cost function for the optimization, and use intensity histograms as features. For pairs of segments belonging to the same section ($p, p' \in P$ and $q, q' \in Q$) we compute their histograms of intensities along their boundary interface. For pairs of segments that belong to adjacent sections, $p \in P$ and $q \in Q$, we compute the intensity histogram in the region of overlap ($p \cap q$), and concatenate it to the histograms of the individual segments p and q. We chose to use intensity histograms despite their simplicity as they have been used effectively in a variety of applications such as object recognition and tracking. The Gentle-Boost classifier outputs a "soft" confidence value. As we seek to rely on the information for which the classification is more confident we allow F to attain real values, and so we directly use the values returned by the Gentle-Boost classifier to populate F. We further trade-off between false split and false merge errors by assigning different costs during training to errors in classifying samples with ground-truth labels $+1$ and -1.

In the rest of the text, the co-clustering formulations combined with boosting are referred to as B-SDP, B-LP and B-LP-R.

5. Simulation experiments

We illustrate the co-clustering problem and the proposed formulation through simulation experiments. The results highlight (a) the relative sparsity of the optimization solutions, (b) the similarity in the results obtained with the SDP and LP formulations, and (c) the need for caution when defining the objective function to avoid trivial solutions.

For the simulations, we define the following problem. We are given as input two segmentations of an image that are noisy refinements of some ground-truth segmentation. Our objective is to co-cluster the segments to recover the ground-truth segmentation. We assume that the two segmentations are computed with two independent algorithms, so that they coincide, with some noise, on the ground truth segment boundaries, while they are uncorrelated on the false boundaries, see Figure 2 for an illustration.



(a) (b) (c)

Figure 2. Illustration of the input for simulation: (a) Ground-truth consisting of 4 segments. (b) and (c) Each segment is perturbed and split into sub-segments to simulate automatic segmentations. Boundaries of the automatic segmentation are shown in dashed, green lines.

We attempt to recover the ground truth segments by seeking clusters of P and Q whose pixel-wise symmetric difference is minimal. Let $C \subseteq P \cup Q$. As P and Q are non-overlapping partitions of the image we can measure the symmetric difference in C by

$$\sum_{p \in P \cap C} |p| + \sum_{q \in Q \cap C} |q| - 2 \sum_{p \in P \cap C} \sum_{q \in Q \cap C} |p \cap q|, \quad (8)$$

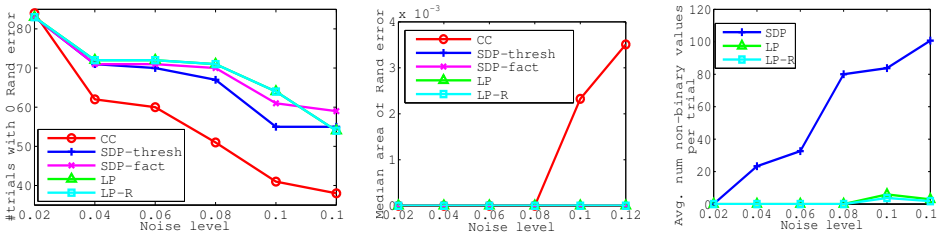
where $|p|$ denote the area of p. Since the diagonal values of F do not affect the optimization, minimizing the symmetric difference is equivalent to simply maximizing the area of overlap between segments, i.e. $\max \sum_C \mathbf{z}_C^T F \mathbf{z}_C$ with $F(p, q) = |p \cap q|$.

Notice that for this QSAP a trivial solution is optimal. In this solution all the segments of P and Q are put in a single cluster, i.e., $R = \mathbf{1}\mathbf{1}^T$, where $\mathbf{1} = (1, ..., 1)^T$. This makes intuitive sense because for a cluster containing all the segments the symmetric difference vanishes. More generally, the $\mathbf{1}\mathbf{1}^T$ trivial solution exists whenever F has all non-positive entries. One approach to avoiding this trivial solution is to add penalties, e.g., as the sum of squares of the cluster-sizes so as to favor smaller clusters in the optimization. We achieve this by modifying F as follows.

$$F = \begin{bmatrix} F_P & F_{PQ} \\ F_{PQ} & F_Q \end{bmatrix}, \quad (9)$$

with $F_{PQ}(p, q) = |p \cap q|$ and $F_P(p, p') = F_Q(q, q') = -\lambda$, and $\lambda > 0$ is a constant adjusting the relative cost for cluster size. We solve this optimization problem with SDP (3), LP (5) and LP-R (6). We then assign two segments to the same cluster if $R_{ij} > 0.5$ in the case of SDP-thresh and $D_{ij} < 0.5$ in the case of LP and LP-R.

In our simulation experiments, the image is a unit square and is divided into four rectangular ground-truth segments.



(a) (b) (c)

Figure 3. Results on simulated data: (a) Number of trials in which 0 Rand index error was achieved by co-clustering with the SDP, LP and LP-R formulations, and the strawman minimum pairwise distance based clustering. The x-axis is the noise level relative to the average automatic segment size. (b) The median area of segments classified as errors by the Rand index averaged over trials. (c) The average number of values in the solutions for the SDP and LP that are in the range $[0.1, 0.9]$ for varying noise levels. Both in terms of counts of Rand errors and the area of the errors, LP and LP-R perform almost the same as the SDP formulation, and all three of these outperform the strawman algorithm. However, the SDP solutions have many more "soft" values compared to those of LP and LP-R.

The automatic segmentations P and Q are produced by randomly splitting each ground truth segment into four subsegments. We further perturb the boundaries of the segments by an amount drawn from a uniform distribution in the range $[-\alpha, \alpha]$ with $\alpha > 0$ measured relative to the average size of segments in the automatic segmentation.

We compare our optimization with a strawman approach that applies a threshold to the area of overlap and forms clusters from the obtained connected components, referred to as CC. We further give extra leeway to this strawman algorithm by allowing it to choose a different, optimal threshold in each trial. Of course in real applications one expects to use a single constant threshold for all trials. We use the Rand Index (7) to evaluate our results.

Figure 3 shows the experimental results. Co-clustering with the SDP, LP and LP-R formulations clearly outperforms the strawman. Furthermore, the solutions computed by the LP and LP-R formulations have predominantly binary values, even for high levels of noise in the range of 10% of the segment sizes. Finally, both post-processing approaches to SDP produce very similar results.

6. The Linkage problem

We illustrate our co-clustering formulation with an application to neuronal reconstruction using electron microscopy (EM) images.

EM is a proven approach to reconstructing brain circuits. The most popular reconstruction technique involves serial section transmission electron microscopy (ssTEM) in which tissue blocks are cut into very thin sections and photographed using EM. There is significant physical deformation of the sections during this process, making it difficult to precisely align hundreds of sections into one 3D stack after imaging. Moreover, there is an order of magnitude difference in z and $x - y$ resolution - the sections are typically $40 - 50$nm thick, whereas the image resolution may be 4nm. Consequently, the practical approach is to segment each 2D section-image into regions belonging to individual cells and then link the 2D segments across sections to form 3D volumes. We will focus on the 3D linkage application: Given the segmentations of two adjacent sections, link the segments that are likely to belong to the same neuron.

There are two challenges in 3D linkage. *(1) Misalignment between sections*: The sections get physically deformed during EM imaging, and the tissue structure may change considerably across sections. Consequently, the cell boundaries may not align perfectly across sections. *(2) Incorrect segmentation*: EM image segmentation is a challenging problem and is a topic of active research. Currently, it is difficult to segment neuronal images such that each segment would correspond to a distinct neuron. This problem can be partially ameliorated by setting the segmentation parameters such that there are very few false mergers.

6.1. Evaluation method

We tested the co-clustering approach on an EM stack of 10 sections taken from the fly larva. Each section's image was 1600×1600 pixels, with approximately 1000 segments in each section. A round robin protocol was employed with 5 sections used for training and the other 5 for testing and vice-versa. The task was to compute co-clustering of segmentations of pairs of adjacent sections.

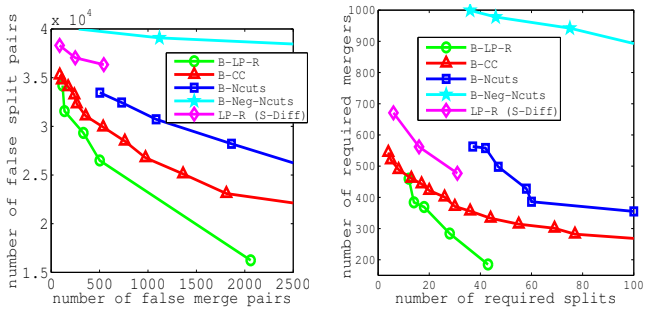
We compare four methods; in each case, a weighted undirected graph $\mathcal{G} = (V, E, W)$ is constructed with nodes corresponding to segments and weights computed by the method in question. To compute the final clustering, a subgraph $\mathcal{G}' = (V, E')$ is constructed by thresholding the edge weights such that $E'(\Delta) = \{e | e \in E \wedge w(e) \leq \Delta\}$. The clusters are defined to be the connected components in $\mathcal{G}'$. Applying a sequence of thresholds results in error curves. We also varied the relative weight given to false-merge and false-split errors when training the boosted classifier. For clarity, we only plot the lower envelope of the results for each method.

1. **B-LP-R**: The output of the boosted classifier is used to populate F, and the LP-R formulation in (6) is used to compute the distance matrix $D = [d_{ij}]$. The edge weight, W_{ij}, between two segments i and j is defined as d_{ij}.

2.**LP-R(S-Diff)**: LP-R is used to optimize the symmetric difference cost function (9) with $\lambda = 1000$. The weights W are set according to D.

3. **B-Ncuts**: Normalized-cuts was used to cluster the segments with the weighted adjacency matrix defined to be $-F$, the output of the boosted classifier. As the original version of N-cuts was defined for non-negative weights [14], we tried the following variations: (a) **B-Ncuts**: ignore the negative weights, (b) **B-Norm-Ncuts**: normalize the weights to the $[0, 1]$ range, and (c) **B-Neg-Ncuts**: use an extension of N-cuts to handle negative weights [17]. We found B-Ncuts and B-Neg-Ncuts to be better than B-Norm-Ncuts. In case of B-Neg-Ncuts, we also tried varying the relative weight given to positive and negative values in F. The output of Ncuts is a set of eigenvectors whose components define the coordinates of the segments in the cluster space. We compute pairwise distance between the segments in this cluster space and use them to define the edge-weights, W in $\mathcal{G}$.

4. **B-CC**: Clustering by connected components. The edge weights, W, in $\mathcal{G}$ are simply the output of the boosted classifier.



(a) (b)

Figure 4. Linkage results using co-clustering with LP, connected components and N-cuts on intensity histogram/boost coefficients, and symmetric difference. (a) Segment-wise Rand index shown as a trade-of between false splits and false merges of segment pairs. (b) Edit score showing the number of required splits and merges by a model proofreader. Co-clustering with B-LP-R outperforms the other methods.

The evaluation is performed at the granularity of segments. The reason is that the EM reconstructions are proofread by experts to correct mistakes, and the corrections are performed by regrouping the member segments. Our objective is to reduce the manual effort involved in proofreading. Therefore, the evaluation measure has granularity at the level of segments, and not pixels. Another evaluation aspect peculiar to EM reconstruction is the asymmetry between false merge and false split errors. Our experience has shown that when reconstructing 100's of EM sections, false merge errors add up very quickly. Proofreaders find it difficult to proofread volumes with large merge errors. Therefore, false merge errors are assigned a much higher cost relative to false split errors during the evaluation.

Two evaluation measures are employed: The Rand index and an edit score. The Rand index is computed as per the definition in (7), except that we count the first and second terms in the summation separately as false split and false merge errors, respectively.

The edit score is designed to mimic the number of clicks that would be required by a proofreader to correct the linkage errors. Drawing from our experience, we model the proofreading as a three step process. First, each cluster from the automatic linkage is assigned to ground-truth segment cluster (according to the proofreader's judgement) based on the number of overlapping segments. Next, for each automatically generated cluster, all the false merged segments are split away one at a time and placed in a distinct new set. Counting the number of such segments gives the number of required splits. At this point only merge operations are required. Counting the number of segment-clusters that should be joined together provides the number of required mergers. Notice that the Rand error grows quadratically with the number of segments that are linked erroneously, whereas the edit score grows linearly with this number.

6.2. Evaluation results

Figure 4(a) shows the number of false merge and false split pairs for the four investigated methods. Figure 4(b) shows the number of split and merge operations required according to the edit score criterion. B-LP-R clearly outperforms the other approaches. Compared to thresholding and connected component analysis, joint optimization improves the linkage. At the operating point typically used for reconstructions (number of required splits $\approx$ 30), B-LP-R produces a 15% decrease in the number of merge operations relative to B-CC for the same number of split operations. Similarly, B-LP-R produces a 7% reduction in the number of false split pairs relative to B-CC for the same number of false merge pairs. Figure 5 shows examples of linkage results obtained with B-CC and B-LP-R.

One possible reason for the inferior performance of N-cuts could be that the EM linkage applications requires the generation of a large number of small clusters. E.g., for a system with 2000 segments, the number of ground-truth clusters is around 500. This may put the normalization in N-cuts at a disadvantage.

6.3. SDP vs. LP-R

We tested the SDP-based co-clustering on a smaller EM dataset of 10 sections with approximately 150 segments per section, see Figure 6. We found that B-LP-R performs slightly better than B-SDP and B-LP, and significantly better than B-Ncuts and B-Neg-Ncuts. However, the results should be interpreted conservatively because the total number of segments in the dataset is only about 1500.

We ran our optimization on a 2.66GHz machine with 32GB RAM. For the small EM dataset, the SDP optimiza-

tion using SDPT3 took roughly 1800 seconds. The LP with all triangular inequalities took roughly 200 seconds, while the LP-R took less than 1 second. Neither SDP nor LP could not be used on the larger dataset with ≈1000 segment per section; only LP-R was feasible and took 5 to 7 seconds.

We found the results of LP-R to be very sparse and with few non-binary values. For systems with F and D approximately of size 1800×1800, the number of non-zero values in $1 - D$ was approximately 4000 ($\approx 0.1\%$) and the number of values in the range $[0.1, 0.9]$ was around 60 ($\approx 2 \times 10^{-3}\%$). For the smaller EM dataset with F and D of size 150×150, LP-R solutions had no values in the range $[0.1, 0.9]$. For the same systems, SDP had around 400 values ($\approx 2\%$) in the range $[0.1, 0.9]$.

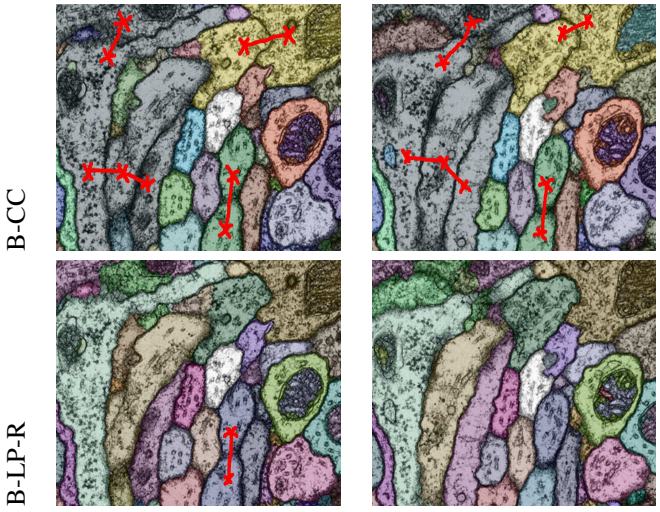


Figure 5. Linkage results for B-CC and B-LP-R at an operating point with number of false split pairs $\approx 2.9 \times 10^4$, or equivalently the number of required mergers ≈ 380. False mergers marked in red. B-LP-R has fewer false mergers.

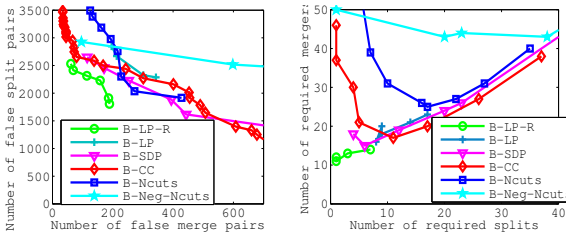


Figure 6. Results of linkage on small EM dataset for co-clustering-LP, co-clustering-LP-R, co-clustering-SDP, Ncuts with positive weights only, and N-cuts with positive and negative weights. LP-R clearly outperforms the other approaches.

7. Conclusion

We presented a formulation of co-clustering of two segmentations as a QSA problem and studied convex relaxations based on SDP and LP. In particular, we demonstrated how to modify the formulation in [4] for practical applications by sparsifying the set of constraints and by learning the cost function. Experiments indicate that the approach outperforms connected components and Ncuts. Moreover, the LP-R's solutions are sparse and mostly binary. We also highlighted the relationship between LP-R and total unimodularity of 2-LDHs. This suggests a possibility of forcing integrality by modifying the LP to ensure acyclicity of the corresponding 2-LDH.

Acknowledgements

This work was funded by Chklovskii Lab., JFRC, HHMI. We are thankful to Dmitri Chklovskii and David Jacobs for helpful discussions about the work. The EM data was collected in collaboration with Shinya Takemura, Wayne Pereanu, Rick Fetter, Zhiyuan Lu and Ian Meinertzhagen.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR-2009*, 2009. 4

[2] T. D. Bie and N. Cristianini. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *Jnl. Machine Learning Research*, 7:1409–1436, 2006. 2, 3

[3] S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming Ser. A*, 103:427–444, 2005. 3

[4] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *FOCS*, 2003. 1, 2, 3, 8

[5] C. R. Coullard and P. H. Ng. Totally unimodular Leontief directed hypergraphs. *Linear Algebra Applications*, 230:101–125, 1995. 4

[6] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability probelms using semidefinite programming. *Journal of ACM*, 42:1115–1145, 1995. 3

[7] O. Goldschmidt and D. Hochbaum. Polynomial algorithm for the k-cut problem. *FOCS*, pages 444–451, 1988. 2

[8] K. Krishnan and J. Mitchell. A linear programming approach to semidefinite programming problems. http://www4.ncsu.edu/~kksivara/publications/cutsdpbundle.pdf. 3

[9] M. P. Kumar, V. Kolmogorov, and P. H. S. Torr. An analysis of convex relaxations of MAP estimation. In *NIPS-2007*, 2007. 2

[10] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Jnl. Operations Research*, 176:657–690, 2007. 1, 2

[11] L. Mukherjee, V. Singh, and C. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009. 2

[12] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006. 2

[13] C. Schellewald, S. Roth, and C. Schnorr. Evaluation of a convex relaxation to a quadratic assignment matching approach for relational object view. *Image and Vision Computing*, 25:1301–1314, 2007. 2

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. and Machine Intell.*, 22(8):888–905, 2000. 2, 7

[15] R. H. Tutuncu, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Ser. B*, 95:189–217, 2003. 3

[16] E. P. Xing and M. I. Jordan. On semidefinite relaxation of normalized *k*-cut and connections to spectral clustering. Technical Report UCB/CSD-3-1265, Computer Science Div., U. C. Berkeley, Berkeley, CA, June 2003. 2

[17] S. X. Yu and J. Shi. Understanding popout through repulsion. In *CVPR-2001*, volume 2, pages 752–757, 2001. 2, 7