Electrical Network Approximation to Constrained Minimum k-Cut (Multiway Cut)

Guillermo A. Angeris and Nikhil Garg

I. Introduction & Related work

We study the following problem. Suppose there is an undirected, weighted graph $G = (V, E)$, with N nodes. There are k vertices $x_1, \ldots, x_k$ which are constrained to be in k different partitions[1] and each cluster is allowed to be of any given size. In general, we seek to minimize the total cut size, that is, allow $S_1, \ldots, S_k$ be the given partition of V, then our objective is

$$\text{minimize } \varphi(S) = \sum_i E(S_i, V - S_i)$$

where $E(C, V - C)$ is the total sum of weights of edges in the cut C, and the constraints are met. This problem, typically called Multiway cut in literature, is a particular version of constrained graph partitioning.

A. Spectral Clustering

Suppose there is a graph $G = (V, E)$ with (weighted, symmetric) adjacency matrix A. The standard spectral clustering approach to partition the graph into k clusters is as follows:

1) Find affinity matrix F from A, where the affinity matrix encodes how similar two points are. For example, one could use the Guassian kernel, $F = -\exp\left\{\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$, where x_i is the ith column of A, and σ^2 is a parameter encoding the width of similar regions. $F = A$ is also often used.
2) Calculate graph Laplacian $L = D - F$, where D is the diagonal matrix encoding the weighted degree of each node.
3) Compute the first k (smallest) eigenvectors of L, not counting the first eigenvector. Form the $N \times k$ matrix U from these eigenvectors. Represent each point i as the ith row of matrix U.
4) Cluster the points with constrained k-means on U, or by random cuts in this smaller dimension space.

Variations include different affinity matrices and ways to normalize the Laplacian, with corresponding changes to later steps as well [1]. More generally, graph embedding approaches create an embedding of the graph in some low dimensional space, and then cluster/cut the nodes in that space. We study various embedding approaches.

B. Constrained Graph Partitioning (Mulitway cut)

Constrained graph partitioning, including constrained spectral clustering, is a well studied problem in literature, and the authors in [2] provide a good overview of the field [2]–[8]. These methods often operate by modifying the affinity matrix F to match the pairwise constraints, such as in [6], where vertices in different clusters are modified to have 0 affinity, and [5], [8], [9] take similar approaches. Other works include modifying the objective in optimization-based spectral methods.

Multiway Cut, in the general case where the clusters have to be $|S_i| \leq b$ for some $b \leq |V|$ is well-known to be NP-hard and is NP-complete for the case where $k = 2$ and $b \leq |V|/2$ [10]. There are some good linear programming-based approximation methods for this problem [11] which, in our experience, do surprisingly well with the LP giving only integral solutions to the problem for almost all instances we ran. The program is as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{(u,v)\in E} w(u,v)||x_u - x_v||_1 \\ \text{subject to} \quad & \\ & x_u \in \triangle_k, \forall u \in V \\ & x_t = e_t, \forall t \in T \end{aligned}$$

Where $w(u, v)$ is the weight on the edge (u, v), T is the set of vertices constrained to be in different partitions, $|T| = k$, $\triangle_k$ is the k-dimensional simplex, and e_t is the standard basis vector with 1 in the tth index. Once $x_u, \forall u \in V$ is found, partitions are created using either the random cut in Algorithm 1 or through a provided derandomization scheme in the paper. This algorithm was originally shown to be a $1.5 - 1/k$-approximation, with later work showing an even stronger bound using the same algorithm.

[1] In general, if we have $> k$ such vertices, we can contract vertices within the same group for additive measures, perform constrained clustering there and then expand them again.

Algorithm 1 Random Cut

procedure RANDOMNETWORKCUT($G, \{f_i\}$)
  $V_i \leftarrow \varnothing$ for each i
  KOrder $\leftarrow \begin{cases} \{1 \dots k-1\} & \text{w.p. } 1/2 \\ \{k-1 \dots 1\} & \text{w.p. } 1/2 \end{cases}$
  $p \leftarrow$ uniform random draw from $[0, 1]$
  for $v \in V(G)$ **do**
    for $k \in$ KOrder **do**
      if $f_k(v) > p$ **then**
        $V_k \leftarrow V_k \cup \{v\}$
      end if
    end for
  end for
  $V_k \leftarrow V \setminus \{\cup_{i \in \{1 \dots k-1\}} V_i\}$
  return $\{V_i\}$
end procedure

II. ALGORITHMS

A. Dirichlet Problems and Computational Approximations

A Dirichlet problem on a graph is defined as a problem of the following form: let $G = (V, E)$ be an undirected graph, and consider some set S with its boundary defined as $\partial S = \{v \in V - S \,|\, \exists q \in S, q \sim v\}$. In general, allow $\partial S \neq \varnothing$ and $S \cup \partial S = V$ and define a function $g : \partial S \to [0, 1]$ which we call the problem's *boundary conditions*. Additionally, define a second function $w : V \times V \to [0, 1]$ such that

$$\sum_{v \sim q} w(q, v) = 1, \, \forall q$$

which we call the 'normalized' edge-weights. Note that $w_{ii} = 0$ since we define by convention that $i \not\sim i$. We call $f : V \to [0, 1]$ a *solution* to the Dirichlet problem if it satisfies $\sum_{v \sim q} w(q, v) f(v) = f(q)$ and $f(v) = g(v), \forall v \in \partial S$. This function is said to be *harmonic* over S following the convention for harmonic functions on manifolds.[2] It is well-known that such a solution exists and is unique by the maximal principle [12]. In general, let $g|_V$ be the trivial extension of g onto the complete graph,[3] and define A as the following

$$A_{ij} = \begin{cases} w_{ij} & i \notin \partial S \\ 0 & \text{otherwise} \end{cases}$$

and note that the solution to the Dirichlet problem is a fixed point of the system $Af + g|_V$, that is

$$f = Af + g|_V$$

which is everywhere a contraction mapping. We use this fact in order to efficiently compute a solution to the Dirichlet problem on large graphs, by beginning with an arbitrary vector, say $f = 0$, and repeatedly applying the mapping until we achieve a given tolerance.

B. Cutting by Electrical Networks

Let V^C be the set of constrained vertices for the weighted graph $G = (V, E)$, with weights $C : V \times V \to \mathbb{R}^{\geq 0}$. Let I_i be the indicator function for cut i, such that $I_i(v) = 1$ if $v \in C_i$ and zero otherwise, where C_i is the set of vertices constrained to the i-th cluster. Now, allow f_i to be the solution to the Dirichlet problem for the i-th cluster, stated as follows, with $R_{jk} = 1/C_{jk}$, $i \sim j$ and $R_j = \sum_{k \sim j} R_{jk}$

$$w^i(j, k) = R_{jk}/R_j$$

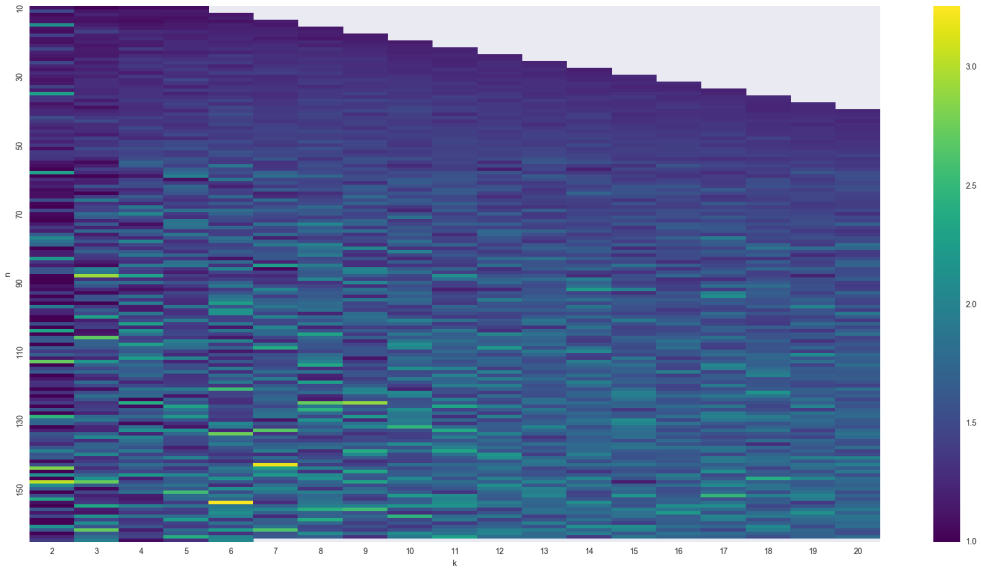
which defines our weights for the problem—note that normalization is guaranteed here by construction, assuming the graph has no isolated vertices—and the boundary condition is defined as

$$g^i(v) = I_i(v), v \in \bigcup_j C_j$$

[2] In fact, the laplacian quadratic form on the graph $f^T L f$ is exactly analogous to the laplacian on a manifold $\int_M \|\nabla h\|^2$. Similar theorems for both hold, including the maximal principle and variational characterization of solutions as energy minimizers.

[3] That is, extend g such that $g|_V(v) = 0, \forall v \in V - \partial S$.

Fig. 1. $ElectricalAlg/LiteratureApprox$ for various n, k.



where $\partial S = \bigcup_i C_i$ and $S = V - \partial S$ for every cluster i. Additionally, note that, in an intuitive sense, our solution $f_i(v)$ gives an idea of how likely[4] it is for a point to belong to a given cluster, i.

We tried two simple algorithms for cutting as follows, the first greedy cut (Algorithm 2) received relatively poor cuts relative to the best approximation algorithms but it was fairly decent for simplicity.

Algorithm 2 Greedy Cut

procedure GREEDYNETWORKCUT($G, \{f_i\}$)
  $V_i = \varnothing$ for each i
  for $v \in V(G)$ **do**
    $\ell \leftarrow \arg\min_i f_i(v)$
    $V_\ell \leftarrow V_\ell \cup \{v\}$
  end for
  return $\{V_i\}$
end procedure

The second cutting method used in the electrical cut is the same as given by [13] and is described in Algorithm 1.

III. EXPERIMENTAL RESULTS

We run experiments comparing our algorithm the the LP in literature, using Watts-Strogatz random graph generation as it very often produces graphs where each partition is non-trivial, unlike Erdos-Renyi or other common graph generation techniques.

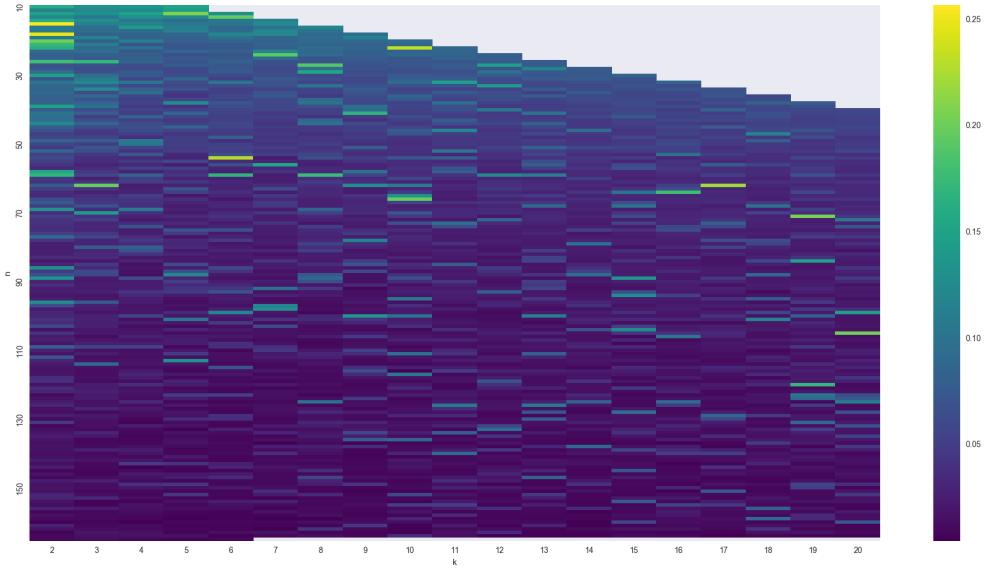
A. Approximation Performance

In our experiments, we found that the LP from literature almost always produced integral solutions. Furthermore, when comparing to the brute force solution for small problems, we never found a case where it did not produce the optimal solution. Our algorithm produces solutions near this approximation algorithm. Figure 1 shows the ratio of our solution over the LP from literature as a function of n and k. Especially for small to medium size n, k, we approximate the solution well, though we have no theoretical bounds.

We note that the solution of the algorithm described in the previous section is feasible except for the constraint of the vector for each x being in the simplex. Depending on the cutting technique (such as if assigning a vertex to the cluster with the

[4]This shouldn't imply that we're strictly probabilistically speaking, we use the term rather loosely here to give an idea. In general, though, this corresponds to the probability (assuming that the probability of walking from any one node i to any other j is given by R_ij/R_i) of returning back to the original node of the given cluster.

Fig. 2. $Time(ElectricalAlg)/Time(LiteratureApprox)$ for various n, k.



highest value, i.e. Algorithm 2), a transformation of the solution can lie on the simplex without affecting the final partition. Thus, it is not surprising that our algorithm probabilistically produces a value that is no better than the solution of the algorithm in literature.

B. Runtime

Figure 2 shows the relative time of our algorithm vs the one in literature. In general, we note that the run-time of our given algorithm is, empirically, much faster than the one proposed in the given paper. In general convergence for the fixed-point method requires $k \approx |V|$, and overall, a matrix-vector multiply takes $|V|$ operations on sparse graphs (assuming $|E| \in O(|V|)$), such that the algorithm requires at most around $O(|V|^2)$ iterations until convergence. The randomization scheme takes $O(k|V|)$, and in practice, the algorithm proposed here ranges is around 10 times faster than the algorithm proposed in [11]. The empirical gap of the given algorithm is comparable to the one given, but we believe that a better rounding scheme might perform much better than the current state.

IV. Conclusion

We studied the Multicut problem, which is a particular version of constrained graph partitioning. Motivated by Dirichlet problems and electrical network analogies for graphs, we developed an iterative approximation algorithm. Our algorithm does not beat the best known approximation algorithm (and can be shown to be worse than it depending on the cutting scheme utilized), though empirically it performs much faster. Overall, we believe this method has some potential to be a candidate for a faster solution that is approximately as accurate as the current best-approximations, yet is also more transparent. In general, it's not hard to believe that searching for further possible methods of making cuts based on this technique might yield far better results than those given above.

References

[1] U. v. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, Dec. 2007.
[2] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, pp. 1–30, Jan. 2014.
[3] G. Wacquet, . Poisson Caillault, D. Hamad, and P.-A. Hbert, "Constrained spectral embedding for K-way data clustering," *Pattern Recognition Letters*, vol. 34, pp. 1009–1017, July 2013.
[4] T.-B.-H. Dao, K.-C. Duong, and C. Vrain, "Constrained clustering by constraint programming," *Artificial Intelligence*, May 2015.
[5] Z. Lu and M. A. Carreira-Perpinan, "Constrained spectral clustering through affinity propagation," pp. 1–8, June 2008.
[6] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, "Spectral learning," in *International Joint Conference of Artificial Intelligence*, Stanford InfoLab, 2003.
[7] K. Wagstaff, *Constrained spectral clustering under a local proximity structure assumption*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2005.

[8] F. Wang, C. H. Ding, and T. Li, "Integrated KL (K-means-Laplacian) Clustering: A New Clustering Approach by Combining Attribute Data and Pairwise Relations.," in *SDM*, pp. 38–48, SIAM, 2009.
[9] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine Learning*, vol. 74, pp. 1–22, Jan. 2009.
[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.
[11] G. Clinescu, H. Karloff, and Y. Rabani, "An Improved Approximation Algorithm for MULTIWAY CUT," *Journal of Computer and System Sciences*, vol. 60, pp. 564–574, June 2000.
[12] P. G. Doyle and J. L. Snell, "Random Walks and Electric Networks," *American Mathematical Monthly*, vol. 94, no. January, p. 202, 2000.
[13] G. Ca, H. Karloff, and Y. Rabani, "An Improved Approximation Algorithm for M," *Journal of Computer and System Sciences*, vol. 574, pp. 564–574, 2000.