When Can We Cluster Data?

Brendan Ames

Department of Mathematics
The University of Alabama

AI / Machine Learning Seminar Series
Sponsored by the UA Cyber Initiative

Tuesday February 2, 2021

Agenda

Present a semidefinite relaxation for the graph clustering problem based on decomposition of graph into densest union of disjoint subgraphs.

Give a probabilistic model for "clusterable" data and graphs, and theoretical recovery guarantees.

Open problems.

Joint with Polina Bombina, UA and Aleksis Pirinen, Lund University.

Clustering

Clustering: partition data so that items in each cluster are similar to each other and items not in the same cluster are dissimilar.

Fundamental problem in statistics and machine learning:

- pattern recognition, computational biology, image processing/computer vison, network analysis.

No consensus on what constitutes a **good** clustering; depends heavily on application.

Intractable: usually modeled as some NP-hard problem (e.g., clique, normalized cut, k-means).

A sanity check

Clustering seems to be a very difficult/ill-posed problem.

Many heuristics seem to work well in practice.

Question: can we show that we can cluster "clusterable" data? How do we model clusterable data?

The Weighted Similarity Graph

Given data and affinity function f indicating similarity between any two items.

Can model the data as weighted similarity graph $G_S = (V, E, \boldsymbol{W})$ as follows:

- Each item is represented by a node in V.
- We add an edge between each pair of two nodes i, j with edge weight $w_{ij} = f(i, j)$.
- w_{ij} is large if i and j are highly similar.

Example: Rehnquist Supreme Court

Data drawn from U.S. Supreme Court decisions (from 1994-95 to 2003-04).

First consider by Hubert and Steinley 2005.

Assign edge-weights corresponding to fraction of decisions on which Justices agreed:

	St	**Br**	**Gi**	**So**	**Oc**	**Ke**	**Re**	**Sc**	**Th**
St	**1**	**0.62**	**0.66**	**0.63**	0.33	0.36	0.25	0.14	0.15
Br	**0.62**	**1**	**0.72**	**0.71**	0.55	0.47	0.43	0.25	0.24
Gi	**0.66**	**0.72**	**1**	**0.78**	0.47	0.49	0.43	0.28	0.26
So	**0.63**	**0.71**	**0.78**	**1**	0.55	0.5	0.44	0.31	0.29
Oc	0.33	0.55	0.47	0.55	**1**	**0.67**	**0.71**	**0.54**	**0.54**
Ke	0.36	0.47	0.49	0.5	**0.67**	**1**	**0.77**	**0.58**	**0.59**
Re	0.25	0.43	0.43	0.44	**0.71**	**0.77**	**1**	**0.66**	**0.68**
Sc	0.14	0.25	0.28	0.31	**0.54**	**0.58**	**0.66**	**1**	**0.79**
Th	0.15	0.24	0.26	0.29	**0.54**	**0.59**	**0.68**	**0.79**	**1**

The Densest k-Disjoint Clique Problem

To cluster the data we want to partition the graph into cliques with heavy support.

A **k-disjoint-clique subgraph** of a graph G is a subgraph of G induced by k disjoint cliques.

Densest k-disjoint-clique problem (KDC): find a k-disjoint-clique subgraph such that the sum of the densities of the k complete subgraphs induced by the cliques is maximized.

Density of complete subgraph induced by C:

$$d(C) = \frac{1}{|C|} \sum_{i \in C} \sum_{j \in C} w_{ij} = \frac{\boldsymbol{v}^T \boldsymbol{W} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}}$$

where $\boldsymbol{v}$ is the characteristic vector of C.

Lifting procedure for KDC

Let $\{C_1, \ldots, C_k\}$ define a k-disjoint-clique subgraph with characteristic vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$

Lift the k characteristic vectors $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$ to the rank-k matrix variable $\boldsymbol{X}$:

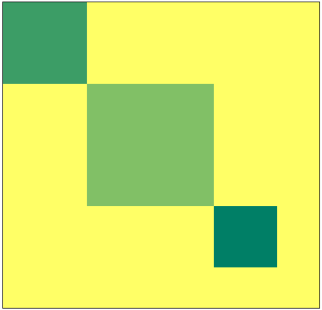
$$\boldsymbol{X} = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{\|\boldsymbol{v}_i\|^2} = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i \boldsymbol{v}_i^T}{|C_i|}$$

Want to find $\boldsymbol{X}$ that maximizes

$$\mathrm{tr}(\boldsymbol{W}\boldsymbol{X}) = \sum_{i=1}^{k} \frac{\boldsymbol{v}_i^T \boldsymbol{W} \boldsymbol{v}_i}{\|\boldsymbol{v}_i\|^2} = \sum_{i=1}^{k} d(C_i)$$

Lifted solutions

Lifted solution $\boldsymbol{X}$ must satisfy:



Inlier rows sum to 1. Outlier rows equal 0: $\boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{e}$

$\boldsymbol{X}$ is symmetric doubly nonnegative: $\boldsymbol{X} \geq \mathbf{0}, \quad \boldsymbol{X} \succeq \mathbf{0}$

$\mathrm{rank}(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}) = k$

plus other combinatorial constraints

SDP Relaxation

Ignoring rank constraint and relaxing combinatorial constraints on $\boldsymbol{X}$ gives the semidefinite program:

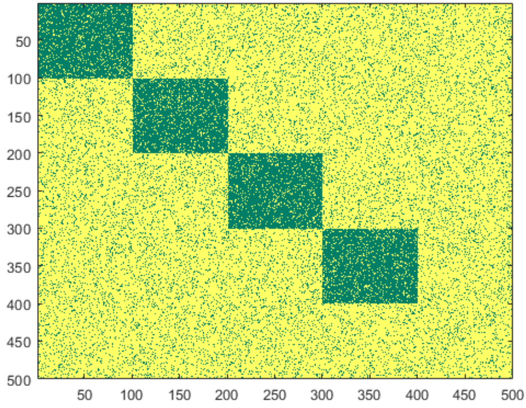
$$\begin{aligned} \max \quad & \mathrm{tr}(\boldsymbol{W}\boldsymbol{X}) \\ \mathrm{s.\,t.} \quad & \boldsymbol{X}\boldsymbol{e} \leq \boldsymbol{e} \\ & \mathrm{tr}(\boldsymbol{X}) = k \\ & \boldsymbol{X} \geq \mathbf{0}, \;\; \boldsymbol{X} \succeq 0. \end{aligned}$$

Question: When does the optimal solution of this relaxation recover underlying cluster structure in similarity graph?

The Stochastic Block Model

Stochastic Block Model (SBM): generate random graph containing k clusters of size r:

- edges within clusters are added independently with probability p
- edges between-clusters are added with probability $q < p$.



Recovery Guarantees under the SBM

Chen/Xu (2014) characterize when graphs sampled from the SBM are:

- trivial to cluster,
- easy to cluster (have polynomial-time algorithm),
- hard to cluster (via NP-hard max likelihood estimation)
- impossible to cluster (data has no meaningful cluster structure).

n-node graph sampled from SBM is **easy** to cluster if

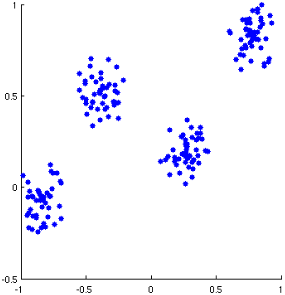
$$\frac{(p-q)^2}{q(1-q)} = \Omega\left(\frac{n}{r^2}\right).$$

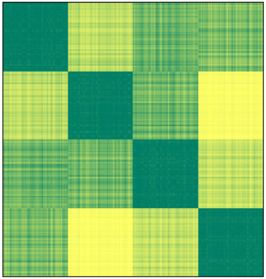
Example: Clustered Euclidean data

Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.

Distance within clusters will be small compared to the distance between clusters if centers are well-separated.

Choose $w_{ij} = \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2)$.





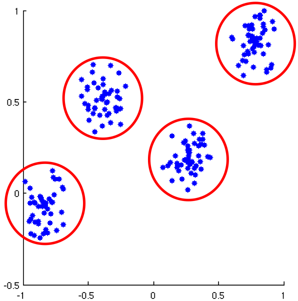
DOES NOT FIT STOCHASTIC BLOCK MODEL!!

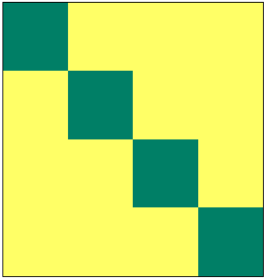
Example: Clustered Euclidean data

Suppose each data point in the ith cluster C_i is placed uniformly at random in a ball centered at $c_i \in \mathbf{R}^d$.

Distance within clusters will be small compared to the distance between clusters if centers are well-separated.

Choose $w_{ij} = \exp(-\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2)$.





DOES NOT FIT STOCHASTIC BLOCK MODEL!!

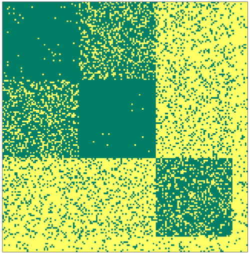
The Heterogeneous Planted Cluster Model

Assume that each node belongs to one of k clusters $C_1, C_2, \ldots, C_k$.

For each $u \in C_i$ and $v \in C_j$ we sample edge weight $w_{uv} = w_{vu}$ from distribution Ω_{ij} with

$$\boldsymbol{E}[w_{uv}] = \mu_{ij} \qquad \mathrm{Var}[w_{uv}] = \sigma_{ij}^2 \qquad 0 \leq w_{uv} \leq 1.$$

Weights within the same block are i.i.d., but weight might not be identically distributed across blocks.



Rethinking the Gap Condition

In the stochastic block model, we have perfect recovery if the gap constant $\gamma = q - p$ is sufficiently large.

In the heterogeneous case, we have perfect recovery if the **weak assortativity constant**

$$\gamma = \min_{\substack{q,s=1,2,\ldots,k \\ q \neq s}} \{\mu_{qq} - \mu_{qs}\}$$

is sufficiently large.

The Recovery Guarantee

Theorem (Pirinen-Ames 2018)
Let $\hat{\sigma} := \max_q \sigma_{qq}$ and $\tilde{\sigma} := \max_{q,s} \sigma_{q,s}$.

Let $\hat{r}$ denote size of the **smallest** planted cluster and r_{k+1} denote the number of outlier nodes.

Then there exists constant $c > 0$ such that if

$$\gamma \hat{r} \geq c \max\left\{ \sqrt{\tilde{\sigma}^2 n},\ \sqrt{\tilde{\sigma}^2 \hat{r} \log n},\ \sqrt{\hat{\sigma}^2 k r_{k+1}},\ \sqrt{k r_{k+1} \log n / \hat{r}},\ \mu_{k+1,k+1} r_{k+1},\ \log n \right\}.$$

then we have **perfect recovery with high probability**.

Signal-to-noise ratio

Suppose that the edge weight is homogeneous: $\alpha = \mu_{qq}$, $\beta = \mu_{qs}$ for all $q \neq s$.
We can recover the planted clusters w.h.p. if

$$\frac{(\alpha - \beta)^2}{\tilde{\sigma}^2} = \Omega\left(\frac{n}{\hat{r}^2}\right).$$

The left-hand side acts as a signal-to-noise ratio: ratio of difference between expected edge weights to noise variance.

This agrees with/generalizes the easy regime for cluster recovery proposed by Chen and Xu (2014), and Jalali et al. (2015).

The relaxation is mostly parameter free: SDP needs number of clusters k but doesn't need estimate of cluster sizes r_i, gap statistic $\alpha - \beta$, etc., seen in similar theoretical guarantees.

Special Case: Stochastic Block Models

Suppose Ω_1 and Ω_2 are Bernoulli distributions with probability of adding an edge p and q respectively ($p > q$) with no outliers ($r_{k+1} = 0$).

Dense case: p, q constant (independent of n).

Have exact recovery w.h.p. if $\hat{r} \geq \hat{c}\sqrt{n}$ for some scalar $\hat{c}$ (depending on p, q).

Sparse case: p constant, $q \leq \frac{\log n}{n}$.

Have exact recovery w.h.p. if $\hat{r} \geq \tilde{c} \log n$ for some constant $\tilde{c}$.

Rehnquist Supreme Court

- Data drawn from U.S. Supreme Court decisions (from 1994-95 to 2003-04).
- First consider by Hubert and Steinley 2005.
- Assign edge-weights corresponding to fraction of decisions on which Justices agreed:

	St	**Br**	**Gi**	**So**	**Oc**	**Ke**	**Re**	**Sc**	**Th**
St	1	0.62	0.66	0.63	0.33	0.36	0.25	0.14	0.15
Br	0.62	1	0.72	0.71	0.55	0.47	0.43	0.25	0.24
Gi	0.66	0.72	1	0.78	0.47	0.49	0.43	0.28	0.26
So	0.63	0.71	0.78	1	0.55	0.5	0.44	0.31	0.29
Oc	0.33	0.55	0.47	0.55	1	0.67	0.71	0.54	0.54
Ke	0.36	0.47	0.49	0.5	0.67	1	0.77	0.58	0.59
Re	0.25	0.43	0.43	0.44	0.71	0.77	1	0.66	0.68
Sc	0.14	0.25	0.28	0.31	0.54	0.58	0.66	1	0.79
Th	0.15	0.24	0.26	0.29	0.54	0.59	0.68	0.79	1

Rehnquist Supreme Court (2)

- Solve KDC with $k = 2$ to get the following partition of the Supreme court:

1: “Liberal”	2: “Conservative”
Stevens (St)	O’Connor (Oc)
Breyer (Br)	Kennedy (Ke)
Ginsberg (Gi)	Rehnquist (Re)
Souter (So)	Scalia (Sc)
	Thomas (Th)

Rehnquist Supreme Court (3)

	St	Br	Gi	So	Oc	Ke	Re	Sc	Th
St	**1**	**0.62**	**0.66**	**0.63**	0.33	0.36	0.25	0.14	0.15
Br	**0.62**	**1**	**0.72**	**0.71**	0.55	0.47	0.43	0.25	0.24
Gi	**0.66**	**0.72**	**1**	**0.78**	0.47	0.49	0.43	0.28	0.26
So	**0.63**	**0.71**	**0.78**	**1**	0.55	0.5	0.44	0.31	0.29
Oc	0.33	0.55	0.47	0.55	**1**	**0.67**	**0.71**	**0.54**	**0.54**
Ke	0.36	0.47	0.49	0.5	**0.67**	**1**	**0.77**	**0.58**	**0.59**
Re	0.25	0.43	0.43	0.44	**0.71**	**0.77**	**1**	**0.66**	**0.68**
Sc	0.14	0.25	0.28	0.31	**0.54**	**0.58**	**0.66**	**1**	**0.79**
Th	0.15	0.24	0.26	0.29	**0.54**	**0.59**	**0.68**	**0.79**	**1**

Rehnquist Supreme Court (4)

- Algorithm is sensitive to choice of k.
- Solve with $k = 3$:

Cluster 1	Cluster 2	Cluster 3
Thomas (Th)	O'Connor (Oc)	Stevens (St)
Scalia (Sc)	Kennedy (Ke)	Breyer (Br)
	Rehnquist (Re)	Ginsberg (Gi)
		Souter (So)

Current projects: Generalization of SBMs

More realistic planted models are needed:

- Overlapping clusters/communities;
- Finding largest of several planted clusters, possibly overlapping (without finding all clusters);
- Random graphs with **dependent** edges;
- Time-varying graphs; etc.,

Future work: Generalization to machine learning

Most machine learning **algorithms** are actually **heuristics**.

Approximately solve model problem for learning task (usually non-convex) and use approximate solution for inference process.

Would be extremely beneficial to have better understanding of the structure of local optima and optimization landscape of these model problems.

- Would allow better choices of initial solutions and heuristic parameters.
- Would encourage greater public trust in methods, more interpretability of results/predictions, etc.

Examples: Compressed Sensing

Compressed sensing / LASSO: can find **sparsest** solution of underdetermined linear system by solving convex relaxation

$$\min\{\|\boldsymbol{x}\|_1 : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\},$$

where $\|\boldsymbol{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n|$, under certain assumptions about $\boldsymbol{A}$.

Rank minimization: can find **minimum rank** solution of linear system $\mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}$ by solving

$$\min\{\|\boldsymbol{X}\|_* : \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}\},$$

under certain assumptions about $\mathcal{A}$, where $\|\boldsymbol{X}\|_*$ is the matrix nuclear norm.

Example: Combinatorial Optimization

Maximum Clique Problem: Ames/Vavasis 2011 showed that the **maximum clique** of graph $G = (V, E)$ can be found by solving the relaxation

$$\min \left\{ \|\boldsymbol{X}\|_* : \sum_{ij} x_{ij} = k,\ x_{ij} = 0\ \forall\ ij \notin E \right\}$$

if G sampled from **planted clique model**. Recovery guarantee improved in Bombina/Ames 2020.

Similar average case recovery guarantees exist for **sparse PCA**, **nonnegative matrix factorization**, among other NP-hard problems.

Current projects: biclustering

Given set of objects and features, **biclustering** or **co-clustering** aims to partition both simultaneously so objects in bicluster strongly exhibit same features.

Want to obtain groups of objects similar with respect to a particular subset of features, while simultaneously grouping features.

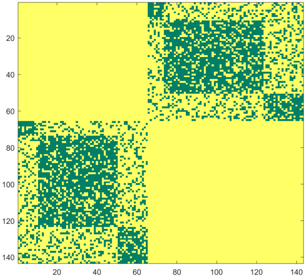
Applications:

- identifying subsets of genes exhibiting similar expression patterns across subsets of experimental conditions in analysis of gene expression data,
- grouping documents by topics in document clustering, and
- grouping customers according to their preferences in collaborative filtering and recommender systems, etc.

The Biclustering SDP

Model the problem as **densest k-disjoint biclique problem**.

Let $G = ((U, V), E)$ be a bipartite graph. Want collection of k-densest bipartite subgraphs, corresponding to k biclusters.



$$\begin{aligned} \max \quad & \mathrm{tr}(\boldsymbol{W}\boldsymbol{Z}) \\ \text{s.t.} \quad & \boldsymbol{Z}_{U,U}\boldsymbol{e} \leq \boldsymbol{e}, \quad \boldsymbol{Z}_{V,V}\boldsymbol{e} \leq \boldsymbol{e} \\ & \mathrm{tr}(\boldsymbol{Z}_{U,U}) = k = \mathrm{tr}(\boldsymbol{Z}_{V,V}) \\ & \boldsymbol{Z} \geq 0, \ \boldsymbol{Z} \in \Sigma_+^{|U|+|V|} \end{aligned}$$

Ames 2014 establishes conditions for perfect recovery in dense homogeneous case.

Would like to generalize to sparse heterogeneous case.

Current projects: Improved Numerical Methods

Current state of the art for solving clustering SDP requires $O(n^3)$ floating point operations for singular value decomposition each iteration; algorithm converges linearly.

Cannot solve large-scale problem instances.

Investigating intermediate relaxation via non-convex quadratic programming (QP).

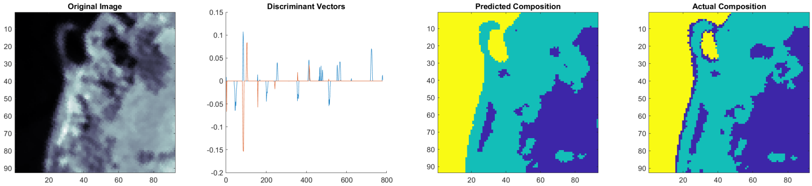
Solve QP using linearized ADMM, with much lower iteration complexity.

- When do we have perfect recovery?
- When does our algorithm converge?

Current projects: Applied Data Analysis

Ford et al. 2021: applied novel classification algorithm to identify comprehension of language via EEG.

Hyperspectral segmentation. Remote sensing (land-cover classification), biomedical samples (malignant vs. benign cells), geological samples (compositional/chronometric analysis)



Thank you!

A. Pirinen and B. Ames. *Exact clustering of weighted graphs via semidefinite programming.* Journal of Machine Learning Research. Year: 2019, Vol: 20, Issue: 30, pp. 1-34.
http://jmlr.org/beta/papers/v20/16-128.html

Software available from bpames.people.ua.edu/software

Thanks. B. Ames supported by

- University of Alabama RGC grants RG14678 and RG14838,
- UA Cyberseed Grant SP14572, and
- NSF Grant #2012554