Clustering using Sparse Recovery

Ming-Jun Lai and Daniel McKenzie
Department of Mathematics, University of Georgia

Why Consider Cluster Extraction?

- Given a graph $G = (V, E)$, finding clusters $C_1, C_2, \ldots, C_k \subset V$ is of interest in data science. Each $C_i$ should have many internal edges, few edges to rest of graph. It is natural to assume that vertices in the same cluster share important properties.
- Typical algorithms (spectral clustering, GeoLouvain, hierarchical clustering) assume that $C_1, \ldots, C_k$ do not overlap and $V = C_1 \cup \ldots \cup C_k$ but real-world graphs are more complicated.
- Would like to allow for overlapping clusters, as well as for background vertices that do not belong to any cluster.
- Real-world graphs are also large. If one is only interested in a certain cluster (e.g., the community containing a specified user in a social network) it can be computationally wasteful to find all clusters.

Definition 1 (Cluster Extraction Problem) Given a graph $G = (V, E)$ and a small set of seed vertices $F \subset V$, find a good cluster $C_F$ containing $F$.

Cluster extraction is agnostic about structure of $V \setminus C_F$. Could be background, other clusters etc.

Figure 1: The college football network of [5]. Clusters (indicated by color) correspond to the different conferences. There are five teams, indicated in black, that are independents and should not be assigned to any cluster.

Totally Perturbed Sparse Recovery

For $x^* \in \mathbb{R}^n$, let $\|x^*\|_0 = \lfloor \|x^*\|_0 \rfloor$ be the $\ell_0$-norm of $x^*$. If $\|x^*\|_0$ is small relative to $n$, we say that $x^*$ is sparse. Given $y = \Phi x^* + \epsilon \in \mathbb{R}^m$, one would like to recover $x^*$ as an sparse solution to linear system $y = \Phi x^*$. Formally:

$$\text{argmin}_{x} \|x\|_0 \text{ such that } \Phi x = y$$

(1)

In compressed sensing $m < n$ so the linear system is underdetermined. Problem (1) is highly non-convex, so either study the convex relaxation ($\ell_1$ minimization) or use greedy approach to solve:

$$\text{argmin}_{x} \|x\|_1 \text{ such that } \Phi x = y$$

(2)

He and Stracke [6, Li] [4] and others, study problem (2) in presence of additive and multi-modal noise. That is, suppose $y = (\Phi + M)x^* + \epsilon$ and $x^*$ is the solution to (2) found using, e.g., subspace pursuit or OMP. Is $x^* = \hat{x}$?

Theorem 2 (Cor. 1 in [4], simplified) Let $\Phi = \Phi + M$ and $\hat{y} = \Phi x^*$ where $\|x^*\|_0 = s$. Suppose that input $y = \Phi x^* + \epsilon$ as received. Define:

$$c_y = \frac{\|x^*\|_2}{\|\epsilon\|_2} \text{ and } c_M = \|M\|_{2\rightarrow 2}$$

Let $x^\Phi$ denote the solution to the following problem, found using subspace pursuit:

$$\text{argmin}_{x} \|\Phi x - y\|_2 \text{ such that } \|x\|_0 \leq s$$

Assume $\delta_{yr} = 4_2, 4_2 > 0.4579$ then:

$$\|x^\Phi - x^*\|_2 \leq C_\delta (c_y, e, c_M) \|x^*\|_2$$

(3)

Turning Cluster Extraction into Sparse Recovery

$L = I - D^{-1} A$ denotes the (normalized) Laplacian of $G$. Let $L^{M}$ denote the Laplacian of $G^M \subset G$ where $G^M$ is obtained by deleting all edges between clusters. Note that clusters $C_1, \ldots, C_k$ of $G$ are now connected components of $G^M$.

Let $L^{M} \mu_i$ denote the indicator vector of $C_i$, then theorem in spectral graph theory states that $L^{M} \mu_i = \mu_i$. Importantly: $L^{M} \mu_i = \mu_i$ hence if $|C_i|$ is small relative to $|V|$, $L^{M}$ is sparse. Assume wlog, that $\mu_i \in C_i$. We can find $L^{M} \mu_i$ as solution to:

$$\text{argmin}_{x} \|L^{M} x\|_0 \text{ subject to } \|x\|_0 \leq \|C_i\|$$

(4)

Of course $L^{M}$ is unknown. In [3] we show that $L = L^{M} + M (\|M\|_{2\rightarrow 2} + 1)$ small. One would hope that if $x^\Phi \approx L^{M} \mu_i$ then:

$$\text{argmin}_{x} \|L x\|_0 \text{ subject to } \|x\|_0 \leq \|C_i\|$$

Then by Theorem 2 $x^\Phi \approx L^{M} \mu_i$. Unfortunately problem (6) turns out to be poorly conditioned. Thus we first use the seed vertices $F \subset C_F$ to find a rough approximation $C_1 \supset F$ and then solve a related sparse recovery problem to extract $C_1$ from $C_1$.

Semi-Supervised Cluster Pursuit (SSCP) [3]

1. Input: Adjacency matrix $A$ on $C$ and $n_i \approx |C_i|$
2. Compute $L = I + D^{-1} A$ and $b = \sum_{i} c_i$
3. Let $\nu = (L^+)^{-1} b$
4. Define $\Omega = \{x : v_i \text{ largest entries in } \nu \} \cup \Gamma$
5. Compute $L = I + D^{-1} A$ and $y = \sum_{i} c_i$
6. Find $x^\nu$ as the solution to $\text{argmin}_{x} (\|L x - y\|_1 \text{ such that } \|x\|_0 \leq 0.1 n_i)$
7. Let $W^\nu = (x^\nu - x)^2 > 0.5$
8. Output: $C_1^\nu = \Omega \setminus W^\nu$, an approximation to $C_1$

Remark 3 In step 6 we use subspace pursuit to take advantage of Theorem 2. Using other sparse recovery algorithms is certainly possible.

Theoretical Guarantees

We consider graphs drawn from the Symmetric Stochastic Block Model, $G \sim \text{SSBM}(m, \alpha, \beta, \gamma)$, where $G$ has $k$ disjoint, equally sized clusters $V = C_1 \cup \ldots \cup C_k$ and edge $(v_i, v_j)$ inserted with probability $p$ if $v_i \in C_i$ and $v_j \in C_j$ or $v_j \in C_i$ and $v_i \in C_j$ for $i \neq j$. Here $|V| = n$ and $|C_i| = n/k$. Using Theorem 2 we prove:

Theorem 4 (B) Suppose $G \sim \text{SSBM}(m, \alpha, \beta, \gamma)$ and $k$ constant, $\gamma = \log(n)/n$ and $p = \omega(\log(n)/n)$ for any $\omega \to \infty$. Let $C_1^\nu$ be the output of SSCP with inputs $A, \Gamma \subset C_1$ where $|\Gamma| = g_j(n)$ for any fixed $j \in [0, 1]$ and $n_j = |C_j|$ then:

$$\|C_1^\nu \cap C_j \cup C_1^\nu \cap C_j\| = o(1)$$

almost surely

Numerical Results

We compared the performance of SSCP against several state-of-the-art cluster extraction methods (Tables 1–3). Full experimental details are contained in [3]. Jaccard := $|C_1^\nu \cap C_j| / |C_1^\nu \cup C_j|$

Table 1: Results for $G \sim \text{SSBM}(m, 10, p, q)$ with $p$ and $q$ as in Theorem 4.

Table 2: Results for four social networks from the facebook1000 data set. Quantities displayed are averaged over ten independent trials per cluster and over all clusters.

Table 3: Results for 20 000 MNIST images, averaged over ten independent trials per digit and over all ten digits. Amount of labeled data varied from 1% to 5%.

Concluding Remarks

• I am currently extending this approach to Dynamic Graphs $G = \{G^{(1)}, \ldots, G^{(T)}\}$
• All code available at: danielmckenzie.github.io
• Questions or comments? daniel.mckenzie@uga.edu

References