Guaranteed biclustering via semidefinite programming

Brendan Ames

Institute for Mathematics and its Applications
University of Minnesota

CP18 Optimization, July 12
SIAM Annual Meeting
July 9-13, 2012

Clustering

- Clustering: Want to partition a given data set so that items in each cluster are similar to each other and items not in the same cluster are dissimilar.
- Intractable in general.
- If data is actually clustered, can we identify the clusters efficiently?

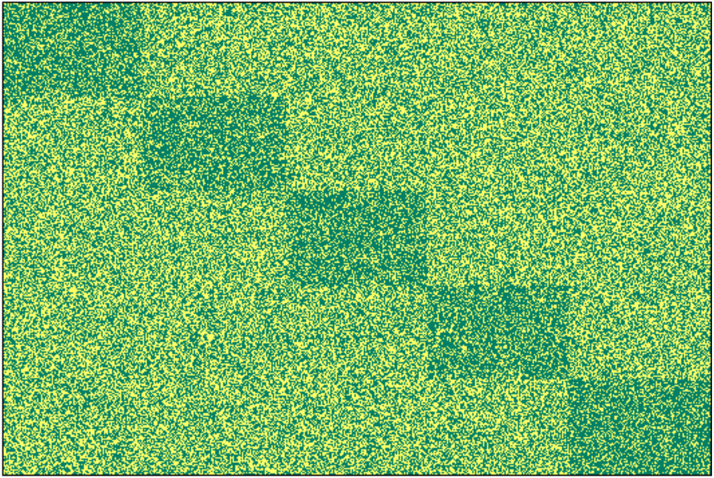
Clustering

- Clustering: Want to partition a given data set so that items in each cluster are similar to each other and items not in the same cluster are dissimilar.
- Intractable in general.
- If data is actually clustered, can we identify the clusters efficiently?
  - **Answer:** Yes, under appropriate assumptions on the data (Ng, Jordan, Weiss 2002, Ostrovsky et al. 2006, Ames-Vavasis 2010, Oymak-Hassibi 2011, Jalali et al 2011, Rohe et al 2011).

Biclustering

- Given a set of objects and features.
- Want to simultaneously partition objects and features so that each cluster of objects exhibit common features and each cluster of features is shared by a set of similar objects.
- Also known as *co-clustering*, *two-mode clustering*.

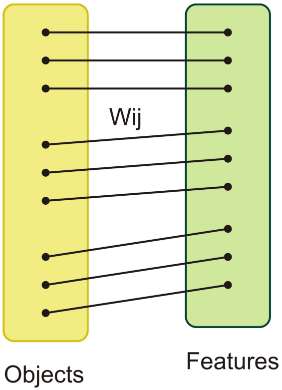
Biclustering example



Applications

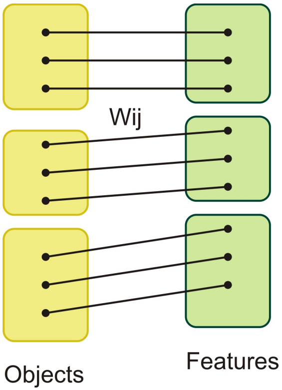
- Biclustering gene expression data:
  - Run a series of experiments where expression level of genes is measured under varying conditions.
  - Obtain matrix where rows are indexed by genes, and each column is the expression level of the genes in a single experiment.
  - Want to identify groups of genes similarly expressed in subsets of experimental conditions.
- Identifying topics in text-database: objects are articles in database, features are keywords, want to identify sets of articles that contain many instances of the same keywords (likely about same topic).
- Recommender systems: want to identify groups of customers and groups of items in catalogue that they prefer.

Biclustering as bipartite graph partitioning



- Consider similarity graph $G_S = (U, V, W)$.
- U is the set of objects
- V is the set of features
- Add edge uv with weight W_{uv} according to the expression level of object u of feature v.

Biclustering as bipartite graph partitioning



- Consider similarity graph $G_S = (U, V, W)$.
- U is the set of objects
- V is the set of features
- Add edge uv with weight W_{uv} according to the expression level of object u of feature v.

Dense subgraphs

- Objects strongly exhibit features within biclusters, relative to other features (and vice versa).
- Biclusters in data will correspond to dense subgraphs.
- Density of a subgraph H is the weight of the edges in H divided by the square root of the number of edges:

$$D_H = \sum_{ij \in E(H)} \frac{W_{ij}}{\sqrt{|U(H)||V(H)|}}.$$

- Want to partition the similarity graph into dense bipartite subgraphs.

The Densest k-disjoint-biclique problem

- A k-disjoint-biclique subgraph is a subgraph of a given graph consisting of k disjoint bipartite complete subgraphs.
- Want to identify the k-disjoint-biclique subgraph G^* maximizing the sum of the densities of the k subgraphs.
- Optimal subgraphs yield biclustering of the underlying data.

Proposed solutions

- Symmetrize W as

$$\tilde{W} = \frac{1}{2}\begin{pmatrix} 0 & W \\ W^T & 0 \end{pmatrix}$$

- Let G^* be a k-disjoint-biclique subgraph given by the bicliques $B_1 = (\mathbf{u}_1, \mathbf{v}_1), \ldots, B_k = (\mathbf{u}_k, \mathbf{v}_k)$.
- The matrix

$$Z^* = \begin{pmatrix} X^* & M^* \\ (M^*)^T & Y^* \end{pmatrix} := \sum_{i=1}^{k} \begin{pmatrix} \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} \\ \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \end{pmatrix} \begin{pmatrix} \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} \\ \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \end{pmatrix}^T$$

  has rank k with $Tr(\tilde{W}Z^*) = \sum_i D_{B_i}$.
- Moreover, X^* and Y^* blocks both have row/col sums at most 1, rank equal to k.

Semidefinite relaxation

- This suggests a relaxation to a rank-constrained semidefinite program:

$$\begin{aligned} max \quad & Tr(\tilde{W}Z) \\ st \quad & Z_{U,U}\mathbf{e} \leq \mathbf{e}, \quad Z_{V,V}\mathbf{e} \leq \mathbf{e} \\ & rank(Z_{U,U}) = k \\ & rank(Z_{V,V}) = k \\ & Z \geq 0, \quad Z \succeq 0. \end{aligned}$$

Semidefinite relaxation

- This suggests a relaxation to a rank-constrained semidefinite program:

$$\begin{aligned} max \quad & Tr(\tilde{W}Z) \\ st \quad & Z_{U,U}\mathbf{e} \leq \mathbf{e}, \quad Z_{V,V}\mathbf{e} \leq \mathbf{e} \\ & Tr(Z_{U,U}) = k \\ & Tr(Z_{V,V}) = k \\ & Z \geq 0, \quad Z \succeq 0. \end{aligned}$$

- Relax further to SDP by replacing rank constraint with trace constraint.

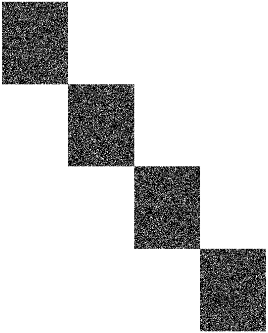
The Planted Bicluster Model

Randomly generate weights $W \in [0,1]^{M \times N}$ according to the following model:

- Start with biclusters (U_i, V_i) of size (m_i, n_i).

The Planted Bicluster Model

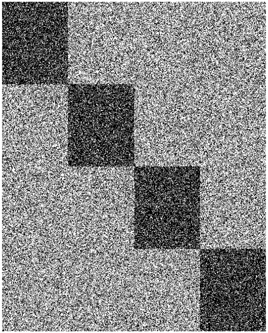
Randomly generate weights $W \in [0,1]^{M \times N}$ according to the following model:



- Start with biclusters (U_i, V_i) of size (m_i, n_i).
- Sample entries of $W(U_i, V_i)$ i.i.d. from probability distribution Ω_1 with mean α.

The Planted Bicluster Model

Randomly generate weights $W \in [0, 1]^{M \times N}$ according to the following model:



- Start with biclusters (U_i, V_i) of size (m_i, n_i).
- Sample entries of $W(U_i, V_i)$ i.i.d. from probability distribution Ω_1 with mean α.
- Sample remaining entries of W i.i.d. from distribution Ω_2 with mean $\beta << \alpha$.

Guaranteed recovery

- Suppose the biclusters are roughly the same size and there are not too many biclusters or outliers then they can be recovered from the optimal solution of the SDP.

- Z^* is the unique optimal solution of the SDP relaxation
- G^* is the unique densest k-disjoint-biclique subgraph
- In particular, if all biclusters are size $m_i = n_i = N^{2/3}$ have exact recovery if
  - $k = O(N^{1/3})$, (not too many biclusters)
  - $m_{k+1}, n_{k+1} = O(N^{1/3})$ (not too many outliers)

Final remarks

- Have presented a new semidefinite programming based heuristic for the biclustering problem.
- If data is sufficiently clusterable then the heuristic successfully recovers the correct biclusters.
- All results translate to the clustering problem as well.
- Open problems:
  - Computation: not practical for most large data sets. Requires solving an SDP with $\Omega(N^2)$ variables and $\Omega(N^2)$ constraints.
  - Clustering issues: choice of k, overlapping clusters/biclusters.
- Preprint: arxiv.org/abs/1202.3663