Semidefinite relaxation for the clustering and biclustering problems

Brendan Ames

Institute for Mathematics and its Applications
University of Minnesota

INFORMS Optimization Society Conference
February 24-26, 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).

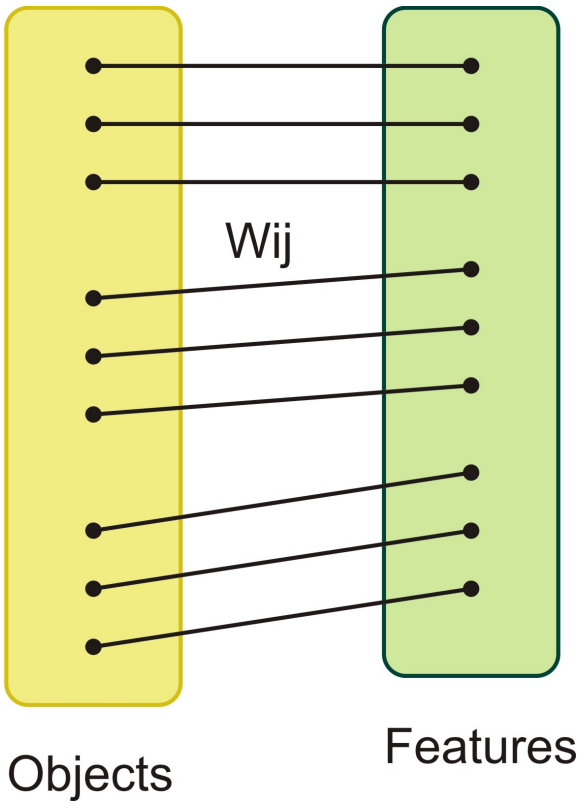
Biclustering

- 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*.

Applications

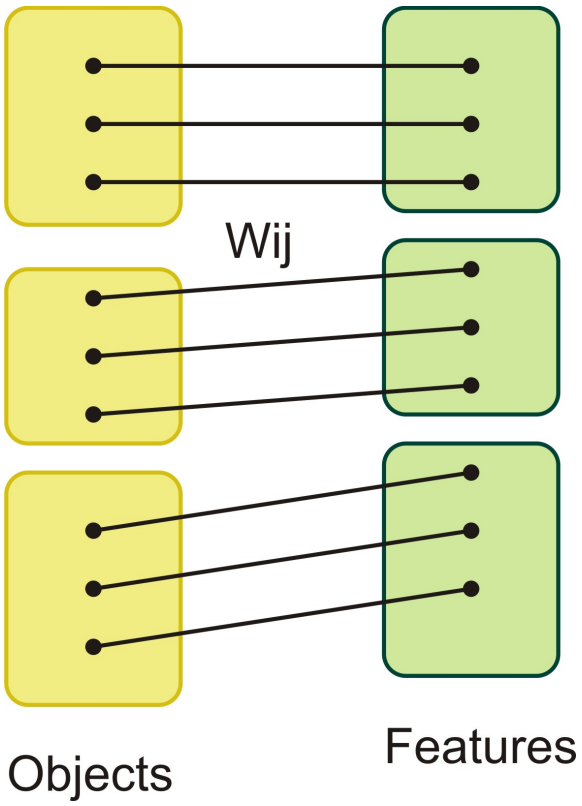
- Biclustering gene expression data:
  - Run a series of experiments where expression level of genes is measured under varying conditions.
  - Obtain matrix where rows are indexed by genes, and each column is the expression level of the genes in a single experiment.
  - Want to identify groups of genes 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.
- Want to partition the similarity graph into dense bipartite subgraphs.
- Biclusters in data will correspond to *dense* subgraphs.

Density of a subgraph $H = (\tilde{V}, \tilde{E})$

- Density of H is the "average degree" of a node in $\tilde{V}$:
  - Unweighted graph:
$$D_H = \frac{|\tilde{E}|}{|\tilde{V}|}$$
  - Weighted graph:
$$D_H = \sum_{ij \in \tilde{E}} \frac{W_{ij}}{|\tilde{V}|}$$
  - Weighted bipartite graph:
$$D_H = \sum_{ij \in \tilde{E}} \frac{W_{ij}}{\sqrt{|\tilde{U}||\tilde{V}|}}$$

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 such that the sum of the densities of the k subgraphs is maximized.
- Optimal subgraphs yield biclustering of the underlying data
- Objects strongly exhibit features within biclusters, relative to other features.

Normalized partition matrices and density

- Let $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ and $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be the characteristic vectors of disjoint subsets of U and V.
- The normalized "partition" matrices corresponding to these subsets are

$$X = \left[\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \; \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \; \cdots \; \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|} \right], \quad Y = \left[\frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \; \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} \; \cdots \; \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|} \right].$$

- The sum of the densities of the subgraphs induced by the $(\mathbf{u}_i, \mathbf{v}_i)$'s is equal to $Tr(X^T W Y)$.

Densest KDB as nonconvex QP

- The set of all normalized k-partition matrices of S is denoted by $npm(S, k)$.
- Then the densest k-disjoint-biclique problem can be posed as the optimization problem

$$\begin{array}{ll} max & Tr(X^T WY) \\ s.t. & X \in npm(U, k), \quad Y \in npm(V, k). \end{array}$$

- This is a nonconvex quadratic program, with combinatorial constraints.
- Likely hard to solve.

Relaxation to SDP

- Symmetrize W as

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

- Lift columns of $Z = [X; Y]$ to the matrix variable

$$Z \mapsto \tilde{Z} = \sum_{i=1}^{k} Z(i,:)Z(i,:)^T.$$

- The $\tilde{Z}_{U,U}$ and $\tilde{Z}_{V,V}$ blocks both have row/col sums at most 1, rank equal to k.

Relaxation to SDP, pt 2

- Matrix lifting yields a relaxation to a rank-constrained semidefinite program:

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

Relaxation to SDP, pt 2

- Matrix lifting yields a relaxation to a rank-constrained semidefinite program:

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

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

A proposed solution

- Every k-disjoint-biclique subgraph G^* with vertex sets $(\mathbf{u}_1, \mathbf{v}_1), \ldots, (\mathbf{u}_k, \mathbf{v}_k)$ defines a feasible solution by

$$Z^* = \begin{pmatrix} X^* & M^* \\ (M^*)^T & Y^* \end{pmatrix}$$

  where

$$X^* = \sum_{i=1}^{k} \frac{1}{m_i} \mathbf{u}_i \mathbf{u}_i^T, \quad Y^* = \sum_{i=1}^{k} \frac{1}{n_i} \mathbf{v}_i \mathbf{v}_i^T, \quad M^* = \sum_{i=1}^{k} \frac{1}{\sqrt{m_i n_i}} \mathbf{u}_i \mathbf{v}_i^T.$$

- Z^* is exactly the lifted solution corresponding to G^*.

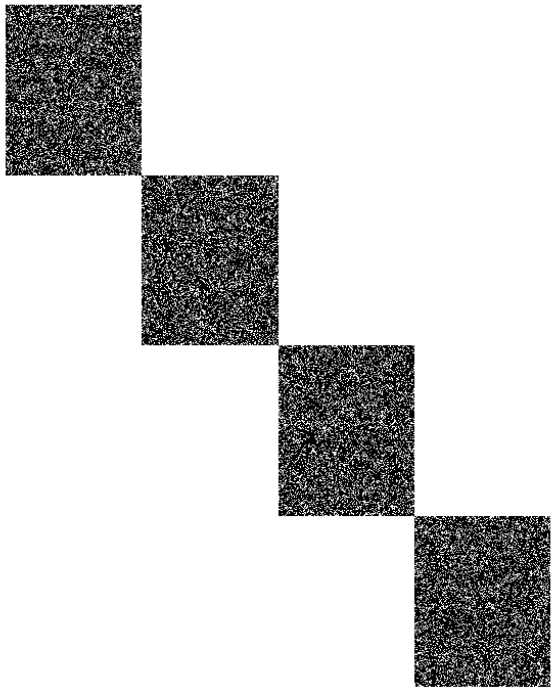
The Planted Bicluster Model

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

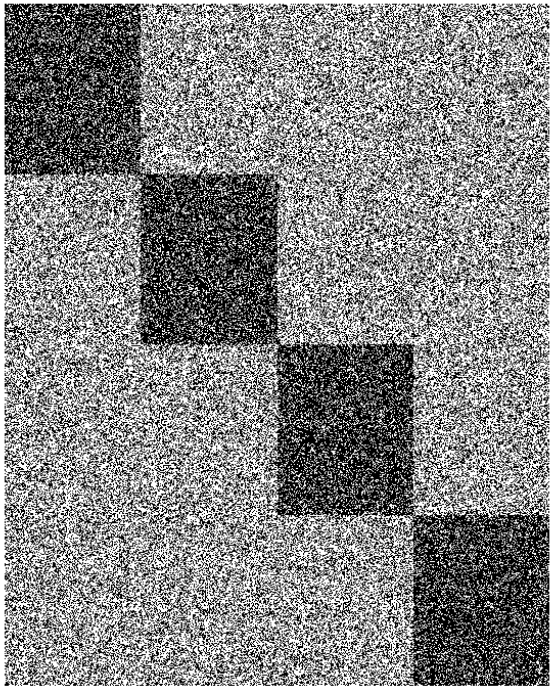
Randomly generate weights $W \in [0,1]^{M \times N}$ according to the following model:



- 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

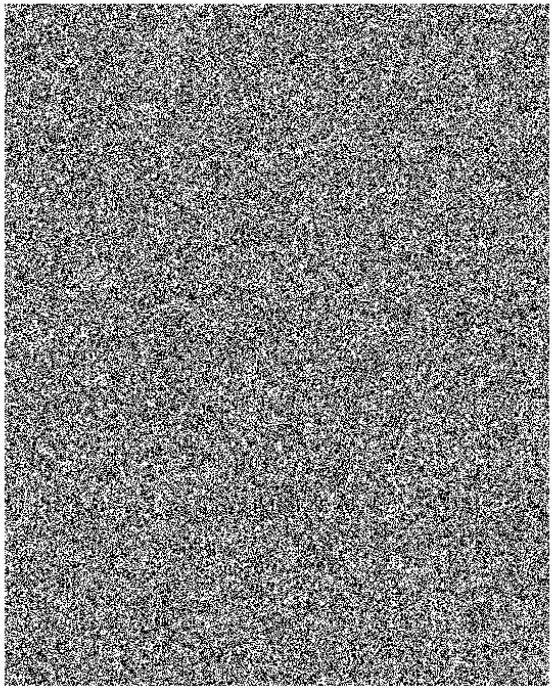
Randomly generate weights $W \in [0,1]^{M \times N}$ according to the following model:



- 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 $\beta << \alpha$.

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 $\beta << \alpha$.
- When can we recover the planted bicliques?

Guaranteed recovery

Theorem

There exist scalars $c_1, c_2, c_3, c_4 > 0$ *such that if*

- $m_i = \tau_i^2 n_i$ *for some scalar* τ_i.
- $\alpha > \beta$ *if* $m_{k+1} = n_{k+1} = 0$ *or* $\alpha > 2\beta$ *o/w,*
- $\alpha\tau_i > \beta\tau_j$ *for all* i, j.
- $n_i = c_1(\alpha - \beta)^2\hat{n}^2$, *where* $\hat{n} = \min_{i=1,\ldots,k} n_i$, *and*
- $c_2\,(k\sum n_i)^{1/2} + c_3(1+\sqrt{n_{k+1}})\sqrt{N} + \beta\tau_{k+1}n_{k+1} \leq c_4(\alpha-\beta)\hat{n}$

then

- Z^* *is the unique optimal solution of SDP relaxation*
- G^* *is the unique densest* k*-disjoint-biclique subgraph*

with probability tending exponentially to 1 *as* $\hat{n} \to \infty$.

What does the theorem mean?

- Theorem says that if biclusters are roughly the same size (smallest $\hat{n}$, biggest $\hat{n}^2$) and there are not too many biclusters or outliers then they can be recovered from the optimal solution of the SDP.

- In particular, if all biclusters are size $m_i = n_i = \hat{n} = N^{2/3}$ 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