Spectral Methods for Ranking and Constrained Clustering

Mihai Cucuringu

University of Oxford
Alan Turing Institute

April 14, 2017

Outline

- Synchronization Ranking
  - M. Cucuringu, *Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and SDP Synchronization*, IEEE Transactions on Network Science and Engineering (2016)
- Constrained clustering
  - M. C., I. Koutis, S. Chawla, G. Miller, and R. Peng, *Simple and Scalable Constrained Clustering: A Generalized Spectral Method*, AISTATS 2016

Ranking from pairwise information

Given n players, and a set of incomplete inconsistent pairwise comparisons of the form

$$\text{(ordinal)} \qquad \text{Player}_i \succ \text{Player}_j$$

or

$$\text{(cardinal)} \qquad \text{Player}_i \;\; 3 \; : \; 1 \;\; \text{Player}_j$$

the goal is to infer a global ranking $\pi(i)$ of the n players

$$\text{Player}_{\pi(1)} \succ \text{Player}_{\pi(2)} \succ \ldots \succ \text{Player}_{\pi(n)}$$

that is as consistent with the given data as best as possible.

- $r_1, r_2, \ldots, r_n$ ground truth ranking
- available pairwise comparisons are a proxy for $r_i - r_j$

Roadmap

- Related methods
  - Serial-Rank
  - Rank-Centrality
  - SVD-based ranking
  - Ranking via least-squares
- The group synchronization problem
- Ranking via angular synchronization
- Numerical experiments on synthetic and real data
- Rank aggregation; Ranking with hard constraints

Challenges

- in most practical applications, the available information
  - is usually incomplete, especially when n is large
  - is very noisy, most measurements are inconsistent with the existence of an underlying total ordering
- at sports tournaments ($G = K_n$) the outcomes always contain cycles: A beats B, B beats C, and C beats A
- aim to recover a total (or partial) ordering that is as consistent as possible with the data
- minimize the number of upsets: pairs of players for which the higher ranked player is beaten by the lower ranked one

Applications

Instances of such problems are abundant in various disciplines, especially in modern internet-related applications such as

- the famous search engine provided by Google
- eBay's feedback-based reputation mechanism
- Amazon's Mechanical Turk (MTurk) crowdsourcing system enables businesses to coordinate the use of human labor to perform various tasks
- Netflix movie recommendation system
- Cite-Seer network of citations
- ranking of college football/basketball teams
- ranking of math courses for students based on grades

Ranking Courses in the *UCLA Math Courses* data set

- n courses offered by the department
- enrolment data for m students
- ordered sequence of courses taken by student
- a partial ranking $\mathcal{S}_i$ on the set of n courses, $|\mathcal{S}_i| \leq n$, $i = 1, \ldots, m$
- extract pairwise comparisons from the partial rankings
- compute a total ordering most consistent with the data

Pure Mathematics

A $(n_s = 86)$	**C** $(n_s = 95)$
Discr. Struct.	Lin. Algebra I
Lin. Algebra I	Hist. of Math
Real Analysis I	Real Analysis I
Lin. Algebra II	Discr. Struct.
Algebra I	Algebra I
Real Analysis II	Ord. Diff. Eqn.'s
Ord. Diff. Eqn.'s	Complex Analysis
Complex Analysis	Game Theory

Very rich literature on ranking

- dates back as early as the 1940s (Kendall and Smith)
- **PageRank**: used by Google to rank web pages in increasing order of their relevance
- Kleinberg's **HITS algorithm**: another website ranking algorithm based on identifying good *authorities* and *hubs* for a given topic queried by the user

Traditional ranking methods fall short:

- developed with **ordinal** comparisons in mind (movie X is better than movie Y)
- much of the current data deals with **cardinal/numerical** scores for the pairwise comparisons (e.g., goal difference in sports)

Erdős-Rényi Outliers noise model

$r_1, \ldots, r_n$ denote the ground truth rankings of the n players

ERO(n, p, η): the available measurements are given by

$$C_{ij} = \begin{cases} r_i - r_j & \text{correct edge} & \text{w.p. } (1-\eta)p \\ \sim \text{Unif}[-(n-1), n-1] & \text{incorrect edge} & \text{w.p. } \eta p \\ 0 & \text{missing edge,} & \text{w.p. } 1-p \end{cases} \tag{1}$$

Multiplicative Uniform Noise model

MUN(n, p, η): noise is multiplicative and uniform

- for cardinal measurements, instead of the true rank-offset measurement $r_i - r_j$, we measure

$$C_{ij} = (r_i - r_j)(1 + \epsilon), \quad \text{where} \quad \epsilon \sim [-\eta, \eta]. \tag{2}$$

- cap the erroneous measurements at $n - 1$ in magnitude
- for ordinal measurements, $C_{ij} = \text{sign}\left((r_i - r_j)(1 + \epsilon)\right)$

E.g., $\eta = 50\%$, and $r_i - r_j = 10$, then $C_{ij} \sim [5, 15]$.

Serial-Ranking (NIPS 2014)

- Fogel et al. adapt the seriation problem to ranking
- propose an efficient polynomial-time algorithm with provable recovery and robustness guarantees
- perfectly recovers the underlying true ranking, even when a fraction of the comparisons are either corrupted by noise or completely missing
- more robust to noise that other classical scoring-based methods from the literature

Serial-Rank

- given as input a skew symmetric matrix C of size $n \times n$ of pairwise comparisons $C_{ij} = \{-1, 0, 1\}$, with $C_{ij} = -C_{ji}$

$$C_{ij} = \begin{cases} 1 & \text{if i is ranked higher than j} \\ 0 & \text{if i and j are tied, or comparison is not available} \\ -1 & \text{if j is ranked higher than i} \end{cases} \tag{3}$$

- for convenience, the diagonal of C is set to $C_{ii} = 1, \forall i = 1, 2, \ldots, n$

Serial-Rank

- the pairwise similarity matrix is given by

$$S_{ij}^{match} = \sum_{k=1}^{n} \left(\frac{1 + C_{ik}C_{jk}}{2} \right) \tag{4}$$

- $C_{ik}C_{jk} = 1$ whenever i and j have the same signs, and $C_{ik}C_{jk} = -1$ whenever they have opposite signs
- S_{ij}^{match} counts the number of matching comparisons between i and j with a third reference item k
- intuition: players that beat the same players and are beaten by the same players should have a similar ranking in the final solution
- final similarity matrix is given by

$$S^{match} = \frac{1}{2} \left(n\mathbf{1}\mathbf{1}^T + CC^T \right) \tag{5}$$

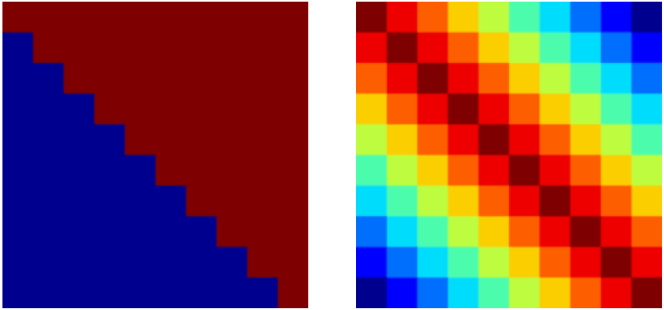
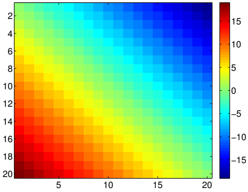


Figure: S^{match} similarity

Clean Rank-Offset:



Algorithm 1 Serial-Rank: an algorithm for spectral ranking using seriation, proposed by Fogel, d'Aspremont and Vojnovic

Require: A set of pairwise comparisons $C_{ij} \in \{-1, 0, 1\}$ or [-1,1]

1: Compute a similarity matrix as shown in (4)

2: Compute the associated graph Laplacian matrix

$$L_S = D - S \tag{6}$$

where D is a diagonal matrix $D = \mathbf{diag}\,(S\mathbf{1})$, i.e., $D_{ii} = \sum_{j=1}^{n} S_{i,j}$ is the degree of node i.

3: Compute the Fiedler vector of S (eigenvector corresponding to the smallest nonzero eigenvalue of L_S).

4: Output the ranking induced by sorting the Fiedler vector of S, with the global ordering (increasing or decreasing order) chosen such that the number of upsets is minimized.

Rank Centrality (NIPS 2012)

- Negahban, Oh, Shah, "*Rank Centrality: Ranking from Pair-wise Comparisons*", NIPS 2012
- algorithm for rank aggregation by estimating scores for the items from the stationary distribution of a certain random walk on the graph of items
- edges encode the outcome of pairwise comparisons
- proposed for the rank aggregation problem: a collection of sets of pairwise comparisons over n players, given by k different ranking systems

Rank Centrality; $k > 1$ rating systems

- the BTL model assumes that

$$\mathbb{P}(Y_{ij}^{(l)} = 1) = \frac{w_j}{w_i + w_j}$$

- w is the vector of positive weights associated to the players

- RC starts by estimating the fraction of times j defeated i

$$a_{ij} = \frac{1}{k} \sum_{l=1}^{k} Y_{ij}^{(l)}$$

- consider the symmetric matrix

$$A_{ij} = \frac{a_{ij}}{a_{ij} + a_{ji}} \tag{7}$$

$$P_{ij} = \begin{cases} \frac{1}{d_{max}} A_{ij} & \text{if } i \neq j \\ 1 - \frac{1}{d_{max}} \sum_{k \neq i} A_{ik} & \text{if } i = j, \end{cases} \tag{8}$$

where d_{max} denotes the maximum out-degree of a node.

Singular Value Decomposition (SVD) ranking

- for cardinal measurements, the noiseless matrix of rank offsets $C = (C_{ij})_{1 \leq i,j \leq n}$ is a skew-symmetric of even rank 2

$$C = \boldsymbol{r}\boldsymbol{e}^T - \boldsymbol{e}\boldsymbol{r}^T \tag{9}$$

  where $\boldsymbol{e}$ denotes the all-ones column vector.
- $\{v_1, v_2, -v_1, -v_2\}$: order their entries, infer rankings, and choose whichever minimizes the number of upsets
- ERO noise model

$$\mathbb{E}C_{ij} = (r_i - r_j)(1 - \eta)p, \tag{10}$$

- $\mathbb{E}C$ is a rank-2 skew-symmetric matrix

$$\mathbb{E}C = (1 - \eta)p(\boldsymbol{r}\boldsymbol{e}^T - \boldsymbol{e}\boldsymbol{r}^T) \tag{11}$$

- decompose the given data matrix C as

$$C = \mathbb{E}C + R \tag{12}$$

Ranking via Least-Squares

- $m = |E(G)|$
- denote by B the edge-vertex incidence matrix of size $m \times n$

$$B_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E(G), \quad \text{and } i > j \\ -1 & \text{if } (i,j) \in E(G), \quad \text{and } i < j \\ 0 & \text{if } (i,j) \notin E(G) \end{cases} \tag{13}$$

- w the vector of size $m \times 1$ containing all pairwise comparisons $w(e) = C_{ij}$, for all edges $e = (i,j) \in E(G)$.
- least-squares solution to the ranking problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad ||Bx - w||_2^2. \tag{14}$$

Synchronization over SO(2)

Estimate n unknown angles

$$\theta_1, \ldots, \theta_n \in [0, 2\pi),$$

given m noisy measurements Θ_{ij} of their offsets

$$\Theta_{ij} = \theta_i - \theta_j \mod 2\pi. \tag{15}$$

Challenges:

- amount of noise in the offset measurements
- only a very small subset of all possible pairwise offsets are measured

Synchronization over $\mathbb{Z}_2$

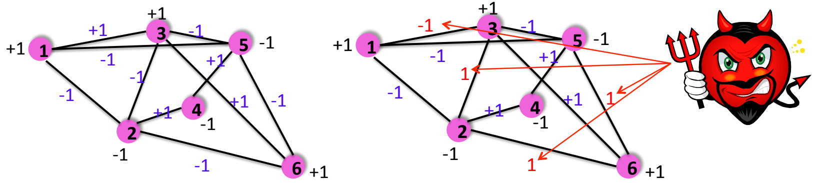


Figure: Synchronization over $\mathbb{Z}_2$ (left: clean, right: noisy)

$$Z_{ij} = \begin{cases} z_i z_j^{-1} & (i,j) \in E \text{ and the measurement is correct,} \\ -z_i z_j^{-1} & (i,j) \in E \text{ and the measurement is incorrect,} \\ 0 & (i,j) \notin E \end{cases}$$

- original solution: $z_1, \ldots, z_N$
- approximated solution: $x_1, \ldots, x_N$
- task: estimate $x_1, \ldots, x_N$ such that we satisfy as many pairwise group relations in Z as possible

Consider maximizing the following quadratic form

$$\max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} \sum_{i,j=1}^{N} x_i Z_{ij} x_j = \max_{x_1,\ldots,x_N \in \mathbb{Z}_2^N} x^T Z x,$$

whose maximum is attained when $x = z$ (noise-free data).
NP-hard problem, but relax to

$$\max_{\sum_{i=1}^{N} |x_i|^2 = N} \sum_{i,j=1}^{N} x_i Z_{ij} x_j = \max_{\|x\|^2 = N} x^T Z x$$

whose maximum is achieved when $x = v_1$, the normalized top eigenvector of Z

$$Z v_1 = \lambda_1 v_1$$

Alternative formulation

Start by formulating the synchronization problem as a least squares problem, by minimizing the following quadratic form

$$\begin{aligned}
\min_{x\in\mathbb{Z}_2^N} \sum_{(i,j)\in E} (x_i - Z_{ij}x_j)^2 &= \min_{x\in\mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + Z_{ij}^2 x_j^2 - 2Z_{ij}x_i x_j \\
&= \min_{x\in\mathbb{Z}_2^N} \sum_{(i,j)\in E} x_i^2 + x_j^2 - 2Z_{ij}x_i x_j \\
&= \min_{x\in\mathbb{Z}_2^N} \sum_{i=1}^{n} d_i x_i^2 - \sum_{(i,j)\in E} 2Z_{ij}x_i x_j \\
&= \min_{x\in\mathbb{Z}_2^N} x^T D x - x^T Z x \\
&= \min_{x\in\mathbb{Z}_2^N} x^T (D - Z) x
\end{aligned}$$

The Eigenvector Method - noiseless case

- $\mathcal{Z} = D^{-1}Z$
- Diagonal matrix Υ, $\Upsilon_{ii} = z_i$ (ground truth value)
- $A = (a_{ij})$ adjacency matrix of the measurement graph
- Write $Z = (z_{ij})$ as $Z = \Upsilon A \Upsilon^{-1}$, for noiseless data $z_{ij} = z_i z_j$
- $\mathcal{Z} = \Upsilon(D^{-1}A)\Upsilon^{-1}$.
- $\mathcal{Z}$ and $D^{-1}A$ all have the same eigenvalues.
- Normalized discrete graph Laplacian $\mathcal{L} = I - D^{-1}A$
- $I - \mathcal{Z}$ and $\mathcal{L}$ have the same eigenvalues
- $1 - \lambda_i^{\mathcal{Z}} = \lambda_i^{\mathcal{L}} \geq 0$, and $v_i^{\mathcal{Z}} = \Upsilon v_i^{\mathcal{L}}$
- G connected $\Rightarrow \lambda_1^{\mathcal{L}} = 0$ is simple, $v_1^{\mathcal{L}} = \mathbf{1} = (1, 1, \ldots, 1)^T$
- $v_1^{\mathcal{Z}} = \Upsilon \mathbf{1}$ and thus $v_1^{\mathcal{Z}}(i) = z_i$

Angular embedding

- A. Singer (2011), spectral and SDP relaxation for the angular synchronization problem
- S. Yu (2012), spectral relaxation; robust to noise when applied to an image reconstruction problem

- embedding in the angular space is significantly more robust to outliers compared to embedding in the usual linear space

The Group Synchronization Problem

- finding group elements from noisy measurements of their ratios
- synchronization over $SO(d)$ consists of estimating a set of n unknown $d \times d$ matrices $R_1, \ldots, R_n \in SO(d)$ from noisy measurements of a subset of the pairwise ratios $R_i R_j^{-1}$

$$\underset{R_1,\ldots,R_n \in SO(d)}{\text{minimize}} \sum_{(i,j)\in E} w_{ij} \|R_i^{-1} R_j - R_{ij}\|_F^2 \tag{16}$$

- w_{ij} are non-negative weights representing the confidence in the noisy pairwise measurements R_{ij}

SDP and Spectral Relaxations

- Least squares solution to synchronization over $R_1, \ldots, R_n \in SO(d)$ that minimizes

$$\underset{R_1,\ldots,R_n \in SO(d)}{\text{minimize}} \sum_{(i,j)\in E} w_{ij} \| R_i^{-1} R_j - R_{ij} \|_F^2 \tag{17}$$

$$\underset{R_1,\ldots,R_n \in SO(d)}{\text{maximize}} \sum_{(i,j)\in E} w_{ij} tr(R_i^{-1} R_j R_{ij}^T) \tag{18}$$

Rewrite objective as tr(G C)
with $G_{ij} = R_i^T R_j$, and

$$G = \bar{R}^T \bar{R}$$

where $\bar{R}_{d\times nd} = [R_1 R_2 \ldots R_n]$

$$C_{ij} = w_{ij} R_{ij}^T (w_{ji} = w_{ij}, R_{ji} = R_{ij}^T)$$

- SDP Relaxation (Singer, 2011)

$$
\begin{aligned}
\underset{G}{\text{maximize}} \quad & tr(GC) \\
\text{subject to} \quad & G \succeq 0 \\
& G_{ii} = I_d, \text{for } i = 1, \dots n \\
& [\text{rank}(G) = d] \\
& [\det(G_{ij}) = 1, \text{for } i,j = 1, \dots n]
\end{aligned}
\qquad (19)
$$

- Spectral Relaxation: via the graph Connection Laplacian L

Let $W \in \mathbb{R}^{nd \times nd}$ with blocks $W_{ij} = w_{ij} R_{ij}$
Let $D \in \mathbb{R}^{nd \times nd}$ diagonal with $D_{ii} = d_i I_d$ where $d_i = \sum_j w_{ij}$

$$L = D - W, \qquad \text{with} \quad L\bar{R}^T = 0$$

Recover the rotations from the bottom d eigenvectors of L + SVD for rounding in the noisy case.

The Graph Realization Problem

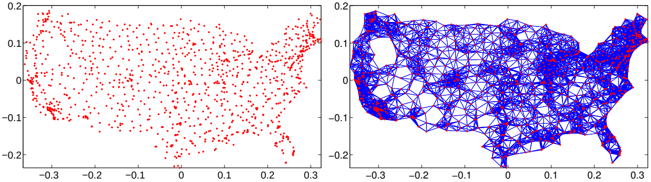


Figure: Original US map with $n = 1090$ and the measurement graph with sensing radius $\rho = 0.032$.

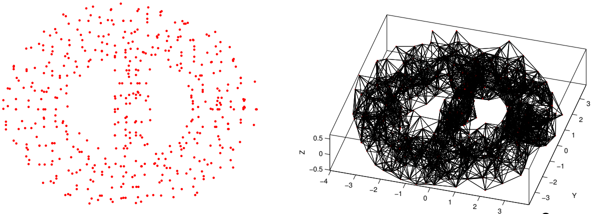


Figure: BRIDGE-DONUT data set of $n = 500$ points in $\mathbb{R}^3$ and the measurement graph of radius $\rho = 0.92$.

The Graph Realization Problem in $\mathbb{R}^d$

- Graph $G = (V, E)$, $|V| = n$ nodes
- Set of distances $l_{ij} = l_{ji} \in \mathbb{R}$ for every pair $(i, j) \in E$
- Goal: find a d-dimensional embedding $p_1, \ldots p_n \in \mathbb{R}^d$ s.t.

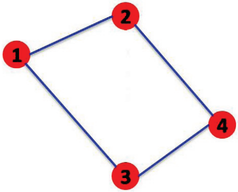
$$||p_i - p_j|| = l_{ij}, \text{ for all } (i, j) \in E$$

- If the solution is unique (up to a rigid motion), then graph is **globally rigid** (uniquely realizable)
- Noise $d_{ij} = l_{ij}(1 + \epsilon_{ij})$ where $\epsilon_{ij} \sim \mathit{Uniform}([-\eta, \eta])$
- Disc graph model with sensing radius ρ, $d_{ij} \le \rho$ iff $(i, j) \in E$

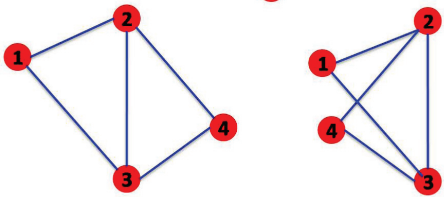
Practical applications:

- Input: sparse noise subset of pairwise distances between sensors/atoms
- Output: d-dimensional coordinates of sensors/atoms

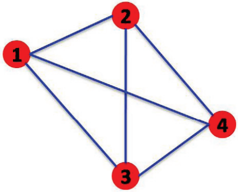
Local and Global Rigidity



Flexible
(not locally rigid)



Locally rigid
(not globally rigid)



Globally rigid

Breaking up the large graph into patches

- Find maximal globally rigid components in the 1-hop neighborhood graph (look for 3-connected components)

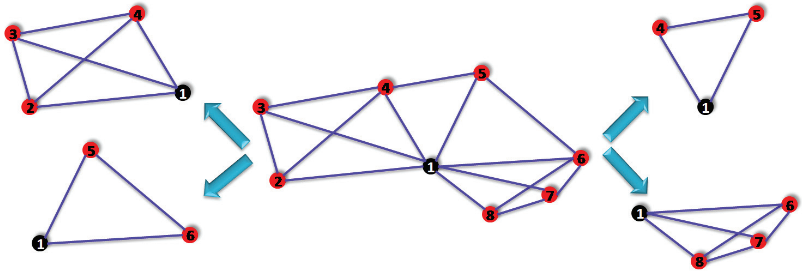


Figure: The neighborhood graph of center node 1 is split into four maximally 3-connected-components (patches): $\{1, 2, 3, 4\}, \{1, 4, 5\}, \{1, 5, 6\}, \{1, 6, 7, 8\}$.

Pairwise alignment of patches

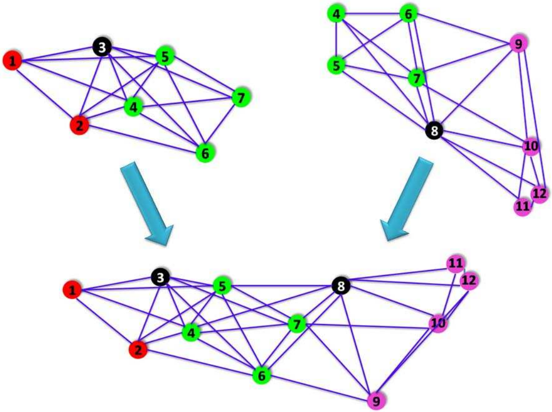
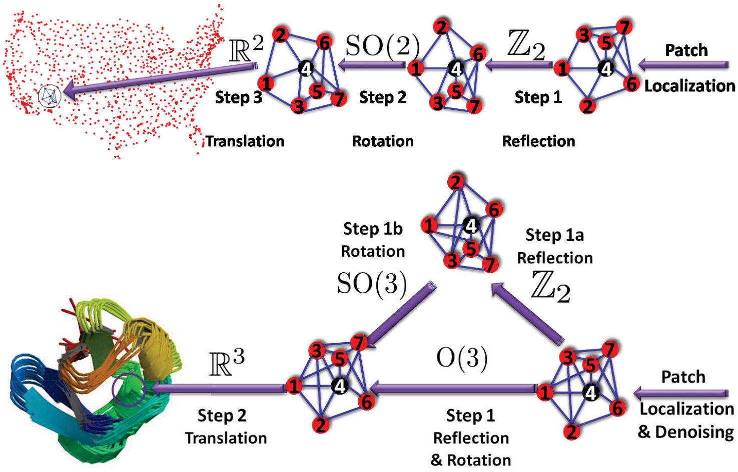


Figure: Optimal alignment of two patches that overlap in four nodes

Local frames and synchronization in $\mathbb{R}^d$



The rightmost subgraph is the embedding of the patch in its own local frame (stress minimization or SDP).

Noise model SO(2) (Singer 2011)

- measurement graph G is Erdős-Rényi $G(n, \alpha)$
- each available measurement is either correct with probably p or a random measurement with probability $1 - p$

$$\Theta_{ij} = \begin{cases} \theta_i - \theta_j & \text{for a correct edge} & \text{w.p. } p\alpha \\ \sim Uniform(S^1) & \text{for an incorrect edge} & \text{w.p. } (1-p)\alpha \\ 0 & \text{for a missing edge,} & \text{w.p. } 1-\alpha. \end{cases} \tag{20}$$

- for $G = K_n$ (thus $\alpha = 1$), the spectral relaxation for the angular synchronization problem
  - undergoes a phase transition phenomenon
  - top eigenvector of H exhibits above random correlations with the ground truth solution as soon as

$$p > \frac{1}{\sqrt{n}} \tag{21}$$

  - can be extended to the general Erdős-Rényi case

Spectral and SDP relaxations for SO(2)

- Build the $n \times n$ sparse Hermitian matrix H

$$H_{ij} = \begin{cases} e^{\imath\theta_{ij}} & \text{if } (i,j) \in E \\ 0 & \text{if } (i,j) \notin E. \end{cases} \tag{22}$$

- Singer considers the following maximization problem

$$\underset{\theta_1,\ldots,\theta_n \in [0,2\pi)}{\text{maximize}} \sum_{i,j=1}^{n} e^{-\imath\theta_i} H_{ij} e^{\imath\theta_j} \tag{23}$$

- gets incremented by $+1$ whenever an assignment of angles θ_i and θ_j perfectly satisfies the given edge constraint $\Theta_{ij} = \theta_i - \theta_j \mod 2\pi$ (i.e., for a *good* edge),
- the contribution of an incorrect assignment (i.e., of a *bad* edge) will be uniformly distributed on the unit circle

Spectral relaxation

Spectral relaxation given by

$$\underset{z_1,\ldots,z_n\in\mathbb{C};\ \sum_{i=1}^{n}|z_i|^2=n}{\text{maximize}} \sum_{i,j=1}^{n} \bar{z}_i H_{ij} z_j \tag{24}$$

- replaced the individual constraints $z_i = e^{\iota\theta_i}$ having unit magnitude by the much weaker single constraint

$$\sum_{i=1}^{n} |z_i|^2 = n$$

- maximization of a quadratic form

$$\underset{||z||^2=n}{\text{maximize}}\ z^* H z \tag{25}$$

solved for $z = v_1$, the top eigenvector of H

Spectral relaxation

- normalize H by the diagonal matrix D with $D_{ii} = \sum_{j=1}^{N} |H_{ij}|$
- define

$$\mathcal{H} = D^{-1}H, \tag{26}$$

- similar to the Hermitian matrix $D^{-1/2}HD^{-1/2}$ via

$$\mathcal{H} = D^{-1/2}(D^{-1/2}HD^{-1/2})D^{1/2}$$

- $\mathcal{H}$ has n real eigenvalues $\lambda_1^{\mathcal{H}} > \lambda_2^{\mathcal{H}} \geq \cdots \geq \lambda_n^{\mathcal{H}}$ and n orthogonal (complex valued) eigenvectors $v_1^{\mathcal{H}}, \ldots, v_n^{\mathcal{H}}$
- estimated rotation angles $\hat{\theta}_1, ..., \hat{\theta}_n$ using the top eigenvector $v_1^{\mathcal{H}}$ via

$$e^{\iota\hat{\theta}_i} = \frac{v_1^{\mathcal{H}}(i)}{|v_1^{\mathcal{H}}(i)|}, \quad i = 1, 2, \ldots, n. \tag{27}$$

- up to an additive phase, since $e^{i\alpha}v_1^{\mathcal{H}}$ is also an eigenvector of $\mathcal{H}$ for any $\alpha \in \mathbb{R}$

Semidefinite Programming relaxation

$$\sum_{i,j=1}^{n} e^{-\iota\theta_i} H_{ij} e^{\iota\theta_j} = \text{trace}(H\Upsilon), \tag{28}$$

- Υ is the (unknown) $n \times n$ Hermitian matrix of rank-1

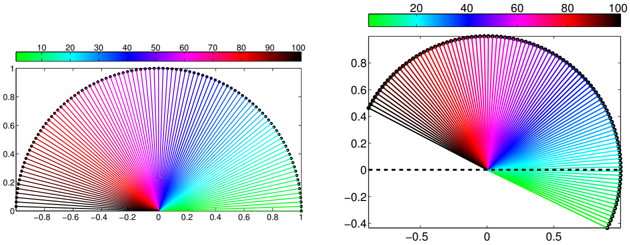
$$\Upsilon_{ij} = e^{\iota(\theta_i - \theta_j)} \tag{29}$$

  with ones in the diagonal $\Upsilon_{ii}, \forall i = 1, 2, \ldots, n$.
- dropping the rank-1 constraint

$$\begin{aligned} \underset{\Upsilon \in \mathbb{C}^{n \times n}}{\text{maximize}} \quad & \mathit{trace}(H\Upsilon) \\ \text{subject to} \quad & \Upsilon_{ii} = 1 \qquad i = 1, \ldots, n \\ & \Upsilon \succeq 0, \end{aligned} \tag{30}$$

- the recovered solution is not necessarily of rank-1
- estimator obtained from the best rank-1 approximation

Synchronization-based ranking



(a) (b)

Figure: (a) Equidistant mapping of the rankings $1, 2, \ldots, n$ around half a circle, for $n = 100$, where the rank of the i^{th} player is i. (b) The recovered solution at some random rotation, motivating the step which computes the best circular permutation of the recovered rankings, which minimizes the number of upsets with respect to the initially given pairwise measurements.

Modding out by the best circular permutation

- $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]$ the ordering induced by the angles recovered from angular synchronization, when sorting the angles from smallest to largest
- compute the pairwise rank offsets associated to the induced ranking

$$P_\sigma(\boldsymbol{s}) = \left(\sigma(\boldsymbol{s}) \otimes \mathbf{1}^T - \mathbf{1} \otimes \sigma(\boldsymbol{s})^T\right) \circ A \tag{31}$$

- $\otimes$ denotes the outer product of two vectors $x \otimes y = xy^T$
- $\circ$ denotes the Hadamard product of two matrices (entrywise product)
- A is the adjacency matrix of the graph G
- $(P_\sigma(\boldsymbol{s}))_{uv} = \sigma(s)_u - \sigma(s)_v$

$$\sigma = \underset{\sigma_1,\ldots,\sigma_n}{\arg\min} \; \frac{1}{2} \left\| \operatorname{sign}(P_{\sigma_i}(\boldsymbol{s})) - \operatorname{sign}(C) \right\|_1 \tag{32}$$

which counts the total number of upsets

The Sync-Rank Algorithm

Require: Matrix C pairwise comparisons

1: Map all rank offsets C_{ij} to an angle $\Theta_{ij} \in [0, 2\pi\delta)$

$$C_{ij} \mapsto \Theta_{ij} := 2\pi\delta \frac{C_{ij}}{n-1} \tag{33}$$

We choose $\delta = \frac{1}{2}$, and hence $\Theta_{ij} := \pi \frac{C_{ij}}{n-1}$.

2: Build the $n \times n$ Hermitian matrix H with $H_{ij} = e^{\imath\theta_{ij}}$, if $(i,j) \in E$, and $H_{ij} = 0$ otherwise

3: Solve the ASP via its spectral or SDP relaxation, and denote the recovered solution by $\hat{r}_i = e^{\imath\hat{\theta}_i} = \frac{v_1^R(i)}{|v_1^R(i)|}, \quad i = 1, 2, \ldots, n$

4: Extract the corresponding set of angles $\hat{\theta}_1, \ldots, \hat{\theta}_n \in [0, 2\pi)$ from $\hat{r}_1, \ldots, \hat{r}_n$, as in (27).

5: Compute the best circular permutation σ^* that minimizes the number of upsets in the induced ordering

6: Output as a final solution the ranking induced by the circular permutation σ^*.

Synchronization Ranking via Superiority Scores

- $C_{ij} = \pm 1, \forall ij \in E$, all angle offsets in synchronization have unit magnitude
- for an ordered pair (i, j), we define the *Superiority Score* of i with respect to j (number of witnesses favorable to i) as

$$W_{ij} = \mathcal{L}(i) \cap \mathcal{H}(j) = \{k \mid C_{ik} = -1 \text{ and } C_{jk} = 1\} \qquad (34)$$

- $\mathcal{L}(x)$ denotes the set of nodes defeated by x,
- $\mathcal{H}(x)$ denotes the set of nodes who defeated x
- the final score (rank offset), used as input for Sync-Rank

$$Z_{ij} = W_{ji} - W_{ij} \qquad (35)$$

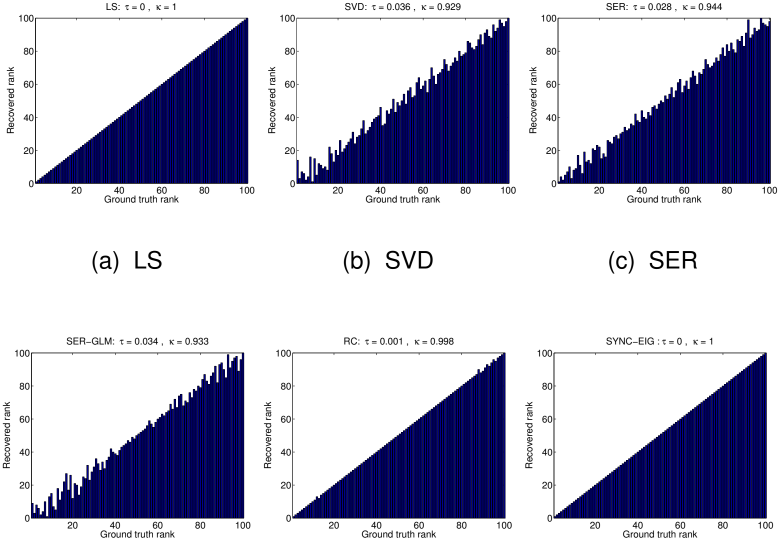
- set $Z_{ji} = -Z_{ij}$
- want Z_{ij} to reflect the true rank-offset between two nodes, a proxy for $r_i - r_j$

Comparison of several algorithms

Acronym	Name
SVD	SVD Ranking
LS	Least Squares Ranking
SER	Serial-Ranking (NIPS 2014)
SER-GLM	Serial-Ranking in the GLM model (NIPS 2014)
RC	Rank-Centrality (NIPS 2012)
SYNC	Sync-Rank via the spectral relaxation
SYNC-SUP	Sync-Rank Superiority Score (spectral relaxation)
SYNC-SDP	Sync-Rank via SDP relaxation

Table: Names of the algorithms we compare, and their acronyms.

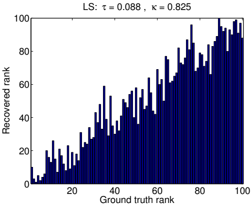
ERO, $\eta = 0$



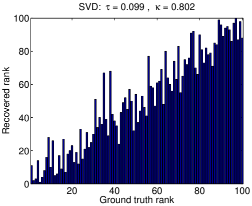
(a) LS (b) SVD (c) SER

(d) SER-GLM (e) RC (f) SYNC-EIG

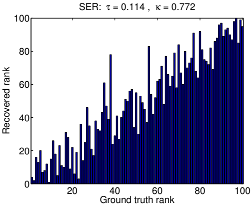
ERO, $\eta = 0.35$



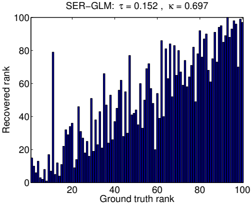
(a) LS



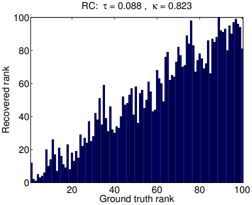
(b) SVD



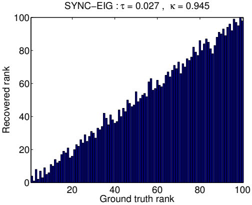
(c) SER



(d) SER-GLM

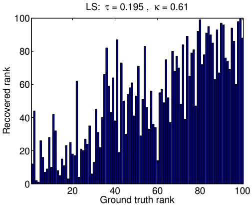


(e) RC

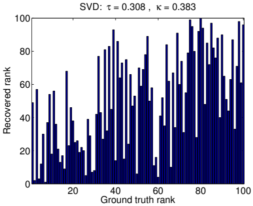


(f) SYNC-EIG

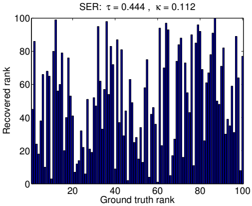
ERO, $\eta = 0.75$



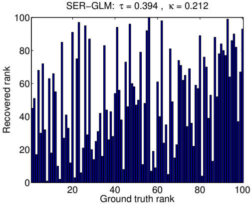
(a) LS



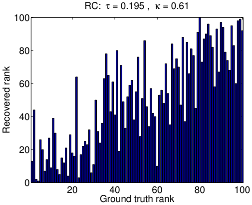
(b) SVD



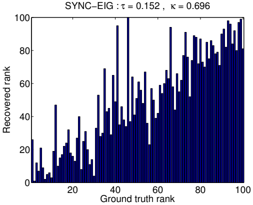
(c) SER



(d) SER-GLM



(e) RC



(f) SYNC-EIG

Kendall distance

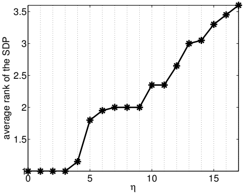
- measure accuracy using the popular Kendall distance
- counts the number of pairs of candidates that are ranked in different order (flips), in the two permutations (the original one and the recovered one)

$$\kappa(\pi_1, \pi_2) = \frac{|\{(i,j) : i < j, [\pi_1(i) < \pi_1(j) \ \wedge \ \pi_2(i) > \pi_2(j)] \ \}}{\binom{n}{2}}$$

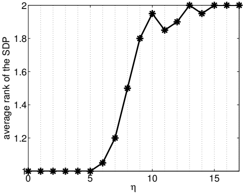
$$\frac{\vee \ [\pi_1(i) > \pi_1(j) \ \wedge \ \pi_2(i) < \pi_2(j)]\}|}{} = \frac{nr.flips}{\binom{n}{2}} \qquad (36)$$

- we compute the Kendall distance on a logarithmic scale

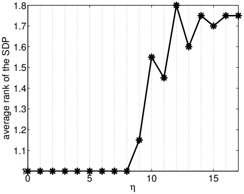
Rank of the SDP solution (cardinal measurements)



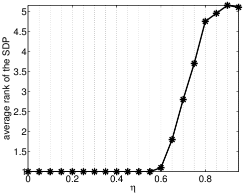
(a) $p = 0.2$, MUN



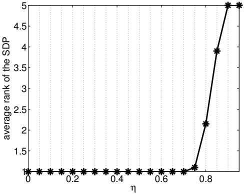
(b) $p = 0.5$, MUN



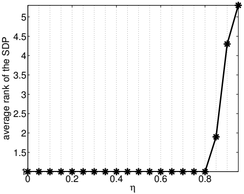
(c) $p = 1$, MUN



(d) $p = 0.2$, ERO

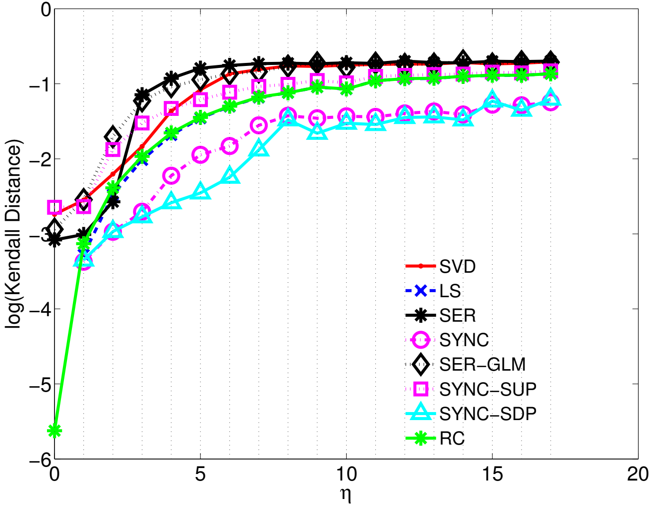


(e) $p = 0.5$, ERO

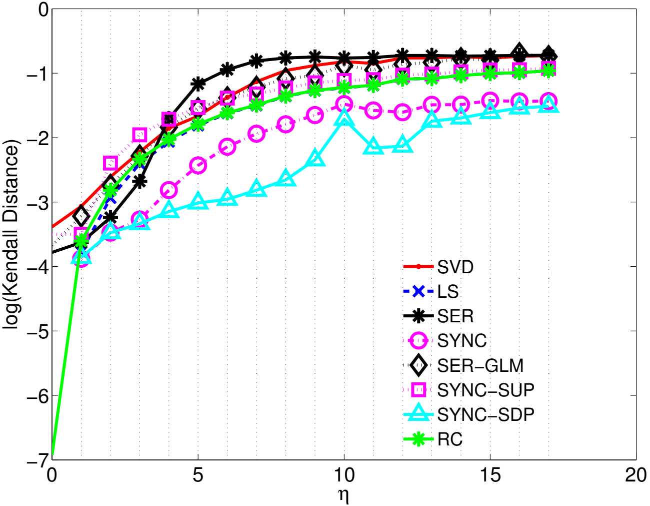


(f) $p = 1$, ERO

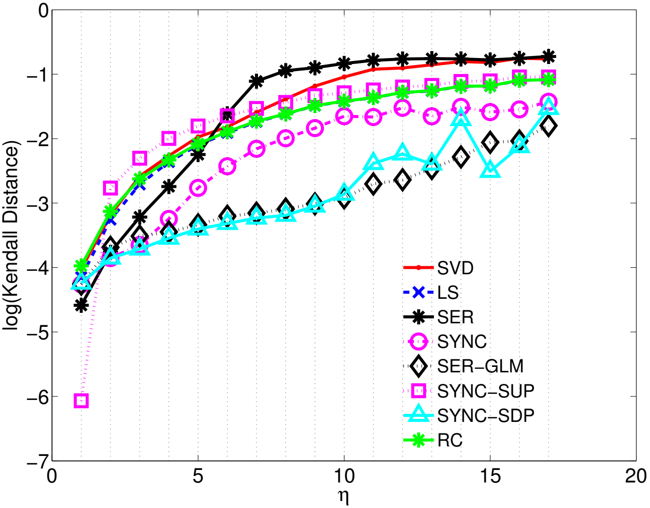
Multiplicative Uniform Noise MUN($n = 200, p = 0.2, \eta$)



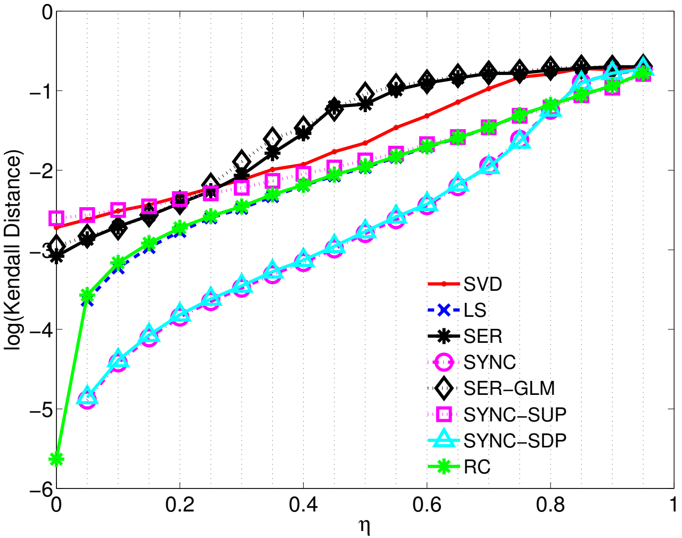
Multiplicative Uniform Noise MUN($n = 200, p = 0.5, \eta$)



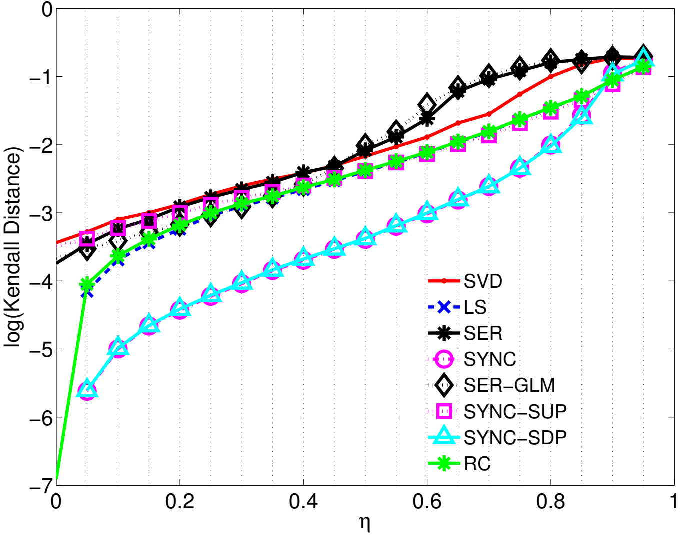
Multiplicative Uniform Noise MUN($n = 200, p = 1, \eta$)



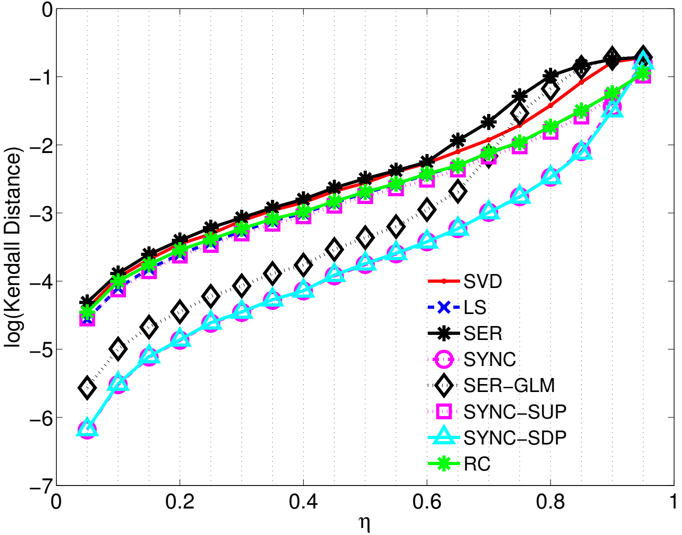
Erdős-Rényi Outliers ERO($n = 200, p = 0.2, \eta$)



Erdős-Rényi Outliers ERO($n = 200, p = 0.5, \eta$)



Erdős-Rényi Outliers ERO($n = 200, p = 1, \eta$)



Rank Aggregation

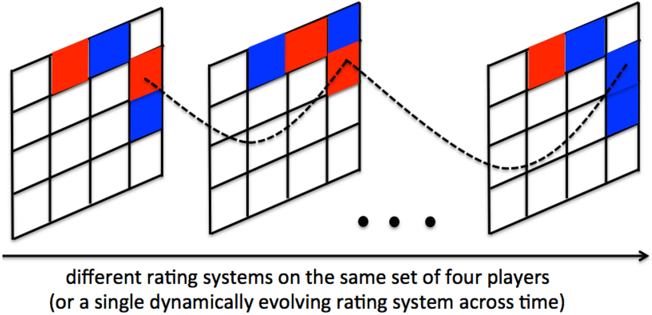


Figure: Multiple rating systems, each one providing its own pairwise comparisons matrix, $C^{(i)}, i = 1, \ldots, k$

Naive approach A: solve individually each of $C^{(i)}, i = 1, \ldots, k$, producing k possible rankings

$$\hat{r}_1^{(i)}, \ldots, \hat{r}_n^{(i)}, i = 1, \ldots, k \tag{37}$$

and average out the resulting rankings.

Naive approach B: first average out all given data

$$\bar{C} = \frac{C^{(1)} + C^{(2)} + \ldots + C^{(k)}}{k} \tag{38}$$

and extract a final ranking from $\bar{C}$.

Synchronization for rank aggregation

- consider the block diagonal matrix of size $N \times N$ ($N = nk$)

$$\boldsymbol{C}_{N \times N} = \textbf{blkdiag}\left(\boldsymbol{C}^{(1)}, \boldsymbol{C}^{(2)}, \ldots, \boldsymbol{C}^{(k)} \right) \tag{39}$$

- denote its Hermitian counterpart by $\boldsymbol{H}_{N \times N}$
- $\mathcal{A}_i, i = 1, \ldots, n$ the set of indices corresponding to player i
- for example, $\mathcal{A}_i = \{i, n+i, 2n+i, \ldots, (k-1)n+i\}$

$$\begin{aligned} & \underset{\theta_1,\ldots,\theta_N;\ \theta_i \in [0,2\pi)}{\text{maximize}} && \sum_{ij \in E(\boldsymbol{G})} e^{-\iota\theta_i}\, \boldsymbol{H}_{ij}\, e^{\iota\theta_j} \\ & \text{subject to} && \theta_i = \theta_j,\ \ \forall\, i,j \in \mathcal{A}_u,\ \ \forall u = 1,\ldots,k. \end{aligned} \tag{40}$$

$$\begin{aligned} & \underset{z=[z_1,\ldots,z_N]:\ z_i \in \mathbb{C},\ \ ||z||^2 = N}{\text{maximize}} && z^*\, \boldsymbol{H}\, z \\ & \text{subject to} && z_i = z_j,\ \ \forall\, i,j \in \mathcal{A}_u,\ \ \forall u = 1,\ldots,k. \end{aligned} \tag{41}$$

Synchronization for rank aggregation

$$\begin{array}{lll} \underset{\Upsilon \in \mathbb{C}^{N \times N}}{\text{maximize}} & Trace(\boldsymbol{H}\Upsilon) & \\ \text{subject to} & \Upsilon_{ij} = 1 & \text{if } i, j \in \mathcal{A}_u, \quad u = 1, \ldots, k \\ & \Upsilon \succeq 0, & \end{array} \tag{42}$$

which is expensive...
However, note that:

$$\begin{aligned} \sum_{u=1}^{k} \sum_{ij \in E(G)} e^{-\iota\theta_i} \, \Theta_{ij}^{(u)} \, e^{\iota\theta_j} &= \sum_{ij \in E(G)} e^{-\iota\theta_i} \left(\sum_{u=1}^{k} e^{\iota\Theta_{ij}^{(u)}} \right) e^{\iota\theta_j} \\ &= \sum_{ij \in E(G)} e^{-\iota\theta_i} \bar{H}_{ij} e^{\iota\theta_j} \end{aligned} \tag{43}$$

where

$$\bar{H} = \sum_{u=1}^{k} H^{(u)} \tag{44}$$

A cheaper synchronization for Rank Aggregation

$$\bar{H} = \sum_{u=1}^{k} H^{(u)} \tag{45}$$

$$\begin{aligned} \underset{\Upsilon \in \mathbb{C}^{n \times n}}{\text{maximize}} \quad & Trace(\bar{H}\Upsilon) \\ \text{subject to} \quad & \Upsilon_{ii} = 1 \qquad i = 1, \ldots, n \\ & \Upsilon \succeq 0 \end{aligned} \tag{46}$$

Alternatively, consider the spectral relaxation for large size problems.

Ranking with hard constraints via SDP

- *vertex constraints*, i.e., information on the true rank of a small subset of players
- *anchors* players; set denoted by $\mathcal{A}$
- *non-anchor* players; set denoted by $\mathcal{F}$

$$
\begin{aligned}
\underset{\Upsilon \in \mathbb{C}^{n \times n}}{\text{maximize}} \quad & Trace(H\Upsilon) \\
\text{subject to} \quad & \Upsilon_{ii} = 1, i = 1, \ldots, n \\
& \Upsilon_{ij} = e^{\iota(a_i - a_j)}, \quad \text{if } i, j \in \mathcal{A} \\
& \Upsilon \succeq 0
\end{aligned}
\tag{47}
$$

Ranking with hard constraints via SDP

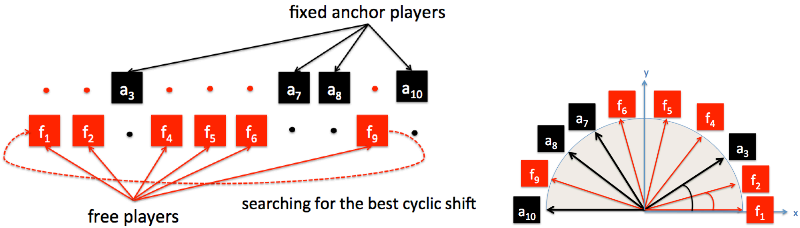
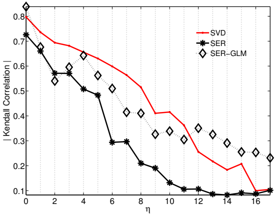
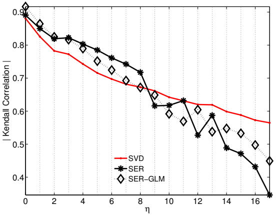


Figure: Left: Searching for the best cyclic shift of the free players, shown in red. The black nodes denote the anchor players, whose rank is known and stays fixed. Right: The ground truth ranking.

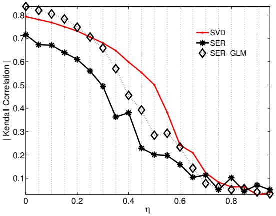
Signless Ranking ($C_{ij} = |r_i - r_j|$). Why does it work?



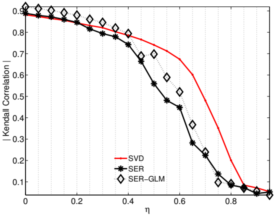
(a) $p = 0.2$, MUN



(b) $p = 0.5$, MUN

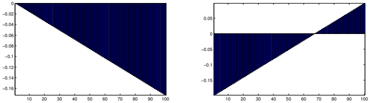


(c) $p = 0.2$, ERO

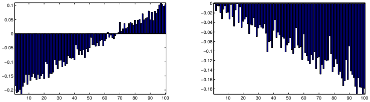


(d) $p = 0.5$, ERO

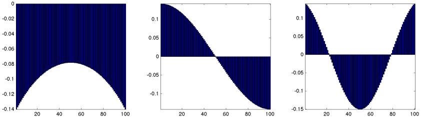
Signless Ranking: Erdős-Rényi Outliers (ERO(n, p, η))



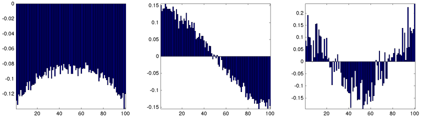
(a) $p = 1, \eta = 0, C = r_i - r_j$



(b) $p = 1, \eta = 0.25, C = r_i - r_j$



(c) $p = 1, \eta = 0, C = |r_i - r_j|$

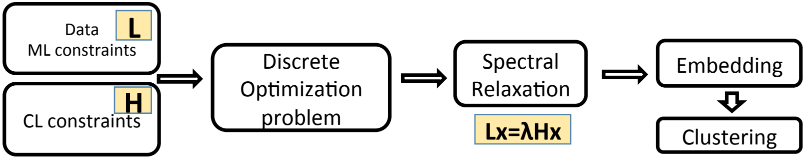


(d) $p = 1, \eta = 0.25, C = |r_i - r_j|$

Clustering with constraints

- important problem in machine learning and data mining
- information about an application task comes in the form of both data and domain knowledge
- domain knowledge is specified as a set of soft
  - must-link (ML) constraints
  - cannot-link (CL) constraints

A schematic overview of our approach



Our contribution

- a principled spectral approach for constrained clustering
- reduces to a generalized eigenvalue problem for which both matrices are Laplacians
- guarantees for the quality of the clusters (for $k = 2$)
- nearly-linear time implementation
- outperforms existing methods both in speed and quality
- can deal with problems at least two orders of magnitude higher than before
- solves data sets with millions of points in less than 2 minutes, on very modest hardware

Related literature

- Rangapuram, S. S. and Hein, M. (2012). Constrained 1-spectral clustering. In AISTATS, pages 1143-1151 (COSC)
- Wang, X., Qian, B., and Davidson, I. (2012). Improving document clustering using automated machine translation. CIKM 12
  - reduces constrained clustering to a generalized eigenvalue problem
  - resulting problem is indefinite and the method requires the computation of a full eigenvalue decomposition

Constrained clustering problem definition

1. G_D: data graph which contains a given number of k clusters that we seek to find
   - $G_D = (V, E_D, w_D)$, with edge weights $w_D \geq 0$ indicating the level of 'affinity' of their endpoints
2. G_{ML} knowledge graph; an edge indicates that its two endpoints should be in the same cluster
3. G_{CL} knowledge graph; an edge indicates that its two endpoints should be in different cluster

- weight of a constraint edge indicates the level of belief
- prior knowledge does not have to be exact or even self-consistent (hence *soft* constraints)
- to conform with prior literature, use existing terminology of 'must link' (ML) and 'cannot link' (CL) constraints

Goal: find k disjoint clusters in G_D obtained via

- cutting a small number of edges in the data graph G_D
- while simultaneously respecting as much as possible the constraints in the knowledge graphs G_{ML} and G_{CL}

Re-thinking constraints

- numerous approaches pursued within the constrained spectral clustering framework
- although distinct, they share a common point of view: constraints are viewed as entities structurally extraneous to the basic spectral formulation
- leads to modifications/extensions with mathematical features that are often complicated
- a key fact is overlooked:

  Standard clustering is a special case of constrained clustering with implicit soft ML and CL constraints.
- recall the optimization in the standard method (NCUT)

$$\phi = \min_{S \subseteq V} \frac{\mathrm{cut}_{G_D}(S, \bar{S})}{\mathrm{vol}(S)\mathrm{vol}(\bar{S})/\mathrm{vol}(V)}$$

Re-thinking constraints

- $\text{vol}(S)$: the total weight incident to the vertex set S
- $\text{cut}_G(S, \bar{S})$: the total weight crossing from S to $\bar{S}$ in G
- data graph G_D is an implicit encoding of soft ML constraints
- pairwise affinities: construed as ”local declarations” that such nodes should be connected rather than disconnected in a clustering
- let now d_i denote the total incident weight of vertex i in G_D
- **demand graph** K of implicit soft CL constraints, defined as

$$K_{ij} = d_i d_j / \text{vol}(V)$$

- easy to verify that $\text{vol}(S)\text{vol}(\bar{S})/\text{vol}(V) = \text{cut}_K(S, \bar{S})$

$$\min_{S \subseteq V} \frac{\text{cut}_{G_D}(S, \bar{S})}{\text{vol}(S)\text{vol}(\bar{S})/\text{vol}(V)} = \min_{S \subseteq V} \frac{\text{cut}_{G_D}(S, \bar{S})}{\text{cut}_K(S, \bar{S})}$$

- i.e., the NCUT objective can be viewed as

$$\min_{S \subseteq V} \frac{\text{weight of violated implicit ML constraints}}{\text{weight of satisfied implicit CL constraints}} \qquad (48)$$

Graph Laplacians

- $G = (V, E, w)$ is a graph with non-negative weights w
- the combinatorial Laplacian L of G is defined by

$$L = D - W : \qquad L_{ij} = \begin{cases} -w_{ij}, & \text{if } i = j \\ \sum_{j \neq i} w_{ij}, & \text{if} i \neq j \end{cases}$$

- L satisfies the following basic identity for all vectors x

$$x^T \, L \, x = \sum_{i,j} w_{ij}(x_i - x_j)^2 \tag{49}$$

- given a cluster $C \subseteq V$, define a cluster indicator vector $x_C(i) = 1$ if $i \in C$ and $x_C(i) = 0$ otherwise. We have:

$$x_C^T \, L \, x_C = \mathrm{cut}_G(C, \bar{C}) \tag{50}$$

- $\mathrm{cut}_G(C, \bar{C}) =$ the total weight crossing from C to $\bar{C}$ in G

The optimization problem

- input: two weighted graphs, the must-link constraints G, and the cannot-link constraints H
- objective: partition V into k disjoint clusters C_i.
- define an individual measure of badness for each cluster C_i

$$\phi_i(G,H) = \frac{cut_G(C_i,\bar{C}_i)}{cut_H(C_i,\bar{C}_i)} \tag{51}$$

- numerator = total weight of the violated ML constraints, because cutting one such constraint violates it.
- denominator = total weight of the satisfied CL constraints, because cutting one such constraint satisfies it
- the minimization of the individual badness is a sensible objective

The optimization problem

- find clusters $C_1, \ldots, C_k$ that minimize the maximum badness, i.e. solve the following problem

$$\Phi_k = \mathbf{min} \max_i \phi_i \qquad (52)$$

- can be captured in terms of Laplacians
- x_{C_i}: indicator vector for cluster i

$$\phi_i(G, H) = \frac{x_{C_i}^T L_G x_{C_i}}{x_{C_i}^T L_H x_{C_i}}$$

- solving the minimization problem in (52) amounts to finding k vectors in $\{0, 1\}^n$ with disjoint support
- optimization problem may not be well-defined if there are very few CL constraints in H
- can be detected easily and the user can be notified
- the merging phase takes automatically care of this case
- we assume that the problem is well-defined

Spectral Relaxation

- to relax the problem we instead look for k vectors in $y_1, \ldots, y_k \in \mathbb{R}^n$, such that for all $i \neq j$, we have $y_i L_H y_j = 0$
- these L_H-orthogonality constraints can be viewed as a relaxation of the disjointness requirement
- their particular form is motivated by the fact that they directly give rise to a generalized eigenvalue problem
- the k vectors y_i that minimize the maximum among the k Rayleigh quotients

  $$(y_i^T L_G y_i)/(y_i^T L_H y_i)$$

  are precisely the generalized eigenvectors corresponding to the k smallest eigenvalues of

  $$L_G x = \lambda L_H x$$

- if H is the demand graph K, the problem is identical to the standard problem $L_G x = \lambda D x$, where $D = \text{diag}(L_G)$, since
  - $L_K = D - dd^T/(d^T \mathbf{1})$
  - eigenvectors of $L_G x = \lambda D x$ are d-orthogonal, where d is vector of degrees in G

Spectral Relaxation

- this fact is well understood and follows from a generalization of the min-max characterization of the eigenvalues for symmetric matrices
- details can be found for instance in Stewart and Sun, *Matrix Perturbation Theory*, 1990
- note that H does not have to be connected
- since we are looking for a minimum, the optimization function avoids vectors that are in the null space of L_H
- $\Rightarrow$ no restriction needs to be placed on x so that the eigenvalue problem is well defined, other than it cannot be the constant vector (which is in the null space of both L_G and L_H), assuming without loss of generality that G is connected

The embedding

- let X be the $n \times k$ matrix of the first k generalized eigenvectors for $L_G x = \lambda L_H x$
- the embedding borrows ideas from

  Lee, Gharan, Trevisan, "Multi-way Spectral Partitioning and Higher-order Cheeger Inequalities", STOC '12

Embedding Computation (based on [Lee et al., 2012])

Input: X, L_H, d

Output: embedding $U \in \mathbb{R}^{n \times k}$, $l \in \mathbb{R}^{n \times 1}$

$u \leftarrow 1^n$

for $i = 1 : k$ **do**

  $x = X_{:,i}$

  $x = x - (x^T d / u^T d) u$

  $x = x / \sqrt{x^T L_H x}$

  $U_{:,i} = x$

end for

for $j = 1 : n$ **do**

  $l_j = ||U_{j,:}||_2$

  $U_{j,:} = U_{j,:} / l_j$

end for

Computing Eigenvectors

- spectral algorithms based on eigenvector embeddings do not require the exact eigenvectors, but only approximations
- the computation of the approximate generalized eigenvectors for $L_G x = \lambda L_H x$ is most time-consuming
- the asymptotically fastest known algorithm for the problem runs in $O(km \log^2 m)$ time, and combines
  - a fast Laplacian linear system solver (Koutis et al. 2011)
  - a standard power method (Glob and Loan 1996)
- in practice, we use the combinatorial multigrid solver (Koutis et al. 2011) which empirically runs in $O(m)$ time
- the solver provides an approximate inverse for L_G which in turn is used with the preconditioned eigenvalue solver LOBPCG (Knyazev 2001)

Merging constraints: a simple heuristic

- often, users provide unweighted constraints G_{ML} and G_{CL}
- construct two weighted graphs $\hat{G}_{ML}$ and $\hat{G}_{CL}$, as follows: if edge (i, j) is a constraint, we take its weight in the corresponding graph to be $d_i d_j / (d_{\min} d_{\max})$, where d_i denotes the total incident weight of vertex i,
- $d_{\min}, d_{\max}$ the minimum and maximum among the d_i's
- let $G = G_D + \hat{G}_{ML}$ and $H = K/n + \hat{G}_{CL}$
- K is the demand graph
- include small copy of K in H to render the problem well-defined in all cases of user input

A generalized Cheeger inequality

- success of standard spectral clustering is often attributed to the existence of non-trivial approximation guarantees
- $k = 2$: Cheeger inequality (Chung 1997)
- provides supporting mathematical evidence for the advantages of expressing constrained clustering as a generalized eigenvalue problem with Laplacians

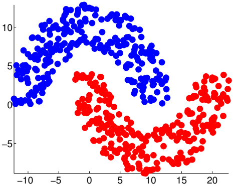
Theorem

Let G and H be any two weighted graphs and d be the vector containing the degrees of the vertices in G. For any vector x such that $x^T d = 0$, we have

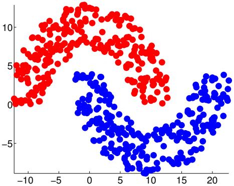
$$\frac{x^T L_G x}{x^T L_H x} \geq \phi(G, K) \cdot \phi(G, H)/4,$$

where K is the demand graph. A cut meeting the guarantee of the inequality can be obtained via a Cheeger sweep on x.

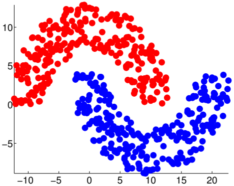
Double-Moons; $G \sim \text{NoisyKnn}(n = 500, k = 30, l = 3)$



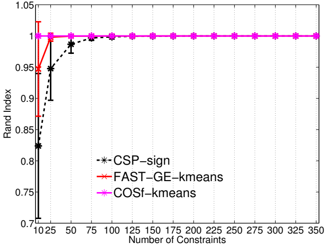
(a) CSP ($c = 75$)

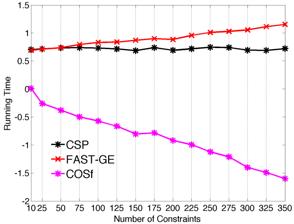


(b) COSf ($c = 75$)

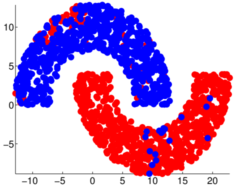


(c) FAST-GE ($c = 75$)

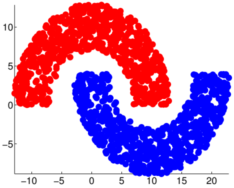




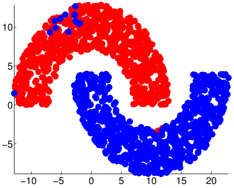
Double Moons - NoisyKnn($n = 1500, k = 30, l = 15$)



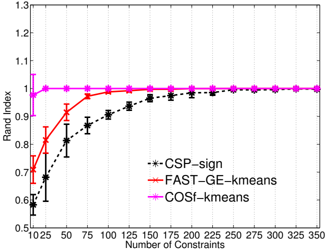
(a) CSP ($c = 75$)

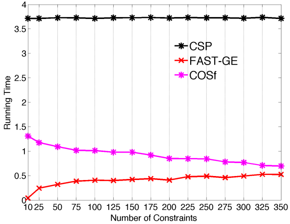


(b) COSf ($c = 75$)

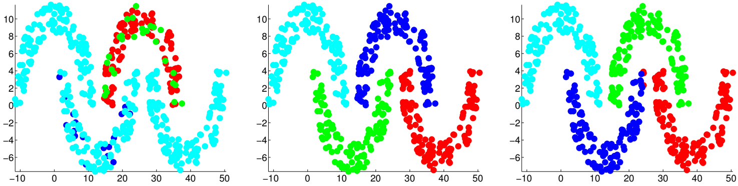


(c) FAST-GE ($c = 75$)





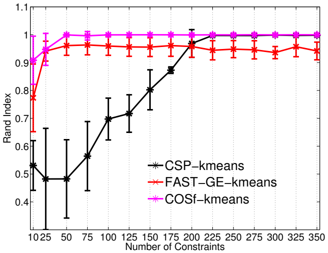
Four-Moons: NoisyKnn($n = 500, k = 30, l = 3$)

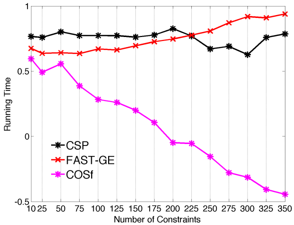


(a) CSP ($c = 75$)

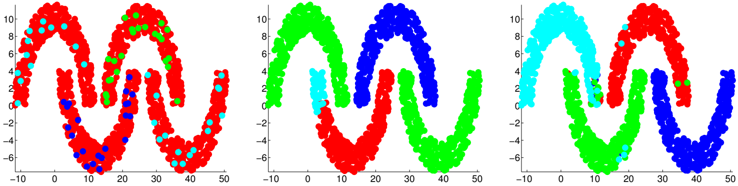
(b) COSf ($c = 75$)

(c) FAST-GE ($c = 75$)

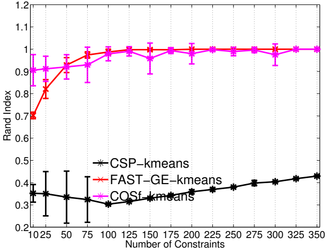


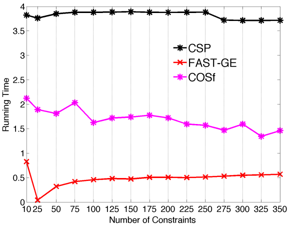


Four-Moons: NoisyKnn($n = 1500, k = 30, l = 15$)

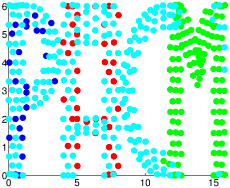


(a) CSP ($c = 75$) (b) COSf ($c = 75$) (c) FAST-GE ($c = 75$)

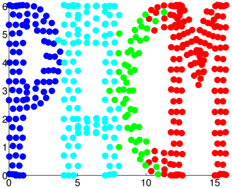




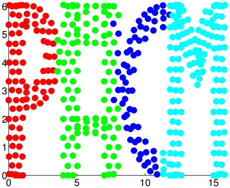
PACM: NoisyKnn($n = 426$, $k = 30$, $l = 15$)



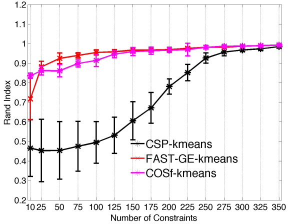
(a) CSP ($c = 125$)

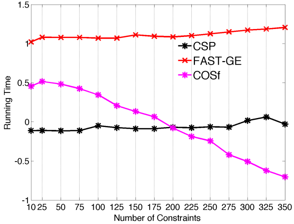


(b) COSf ($c = 125$)

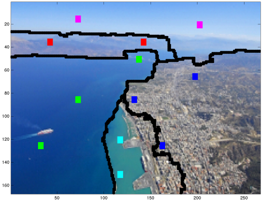


(c) FAST-GE ($c = 125$)

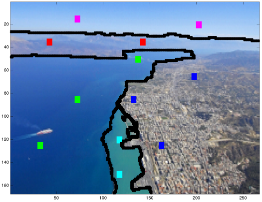




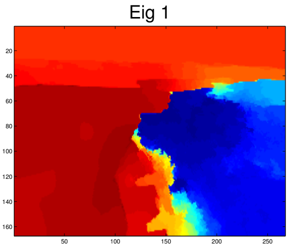
Patras image - 44K pixels ($k = 5$)

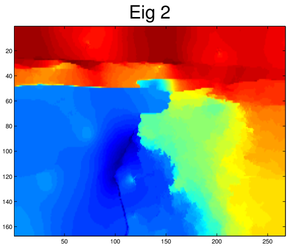


(a) FAST-GE (2.8 sec)



(b) COSf (40.2 sec)





A portion of the Patras image

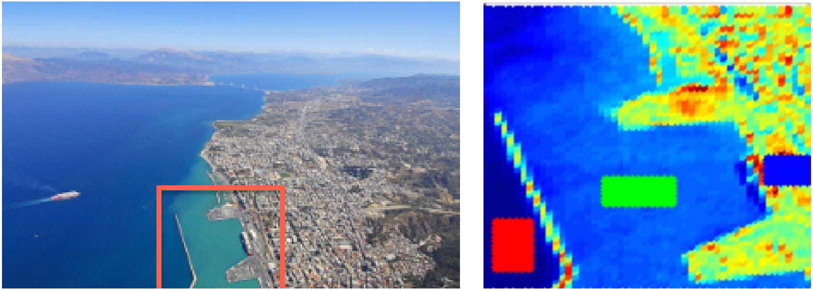
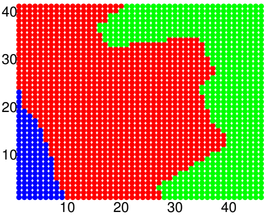
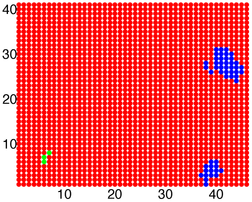


Figure: Cluster membership (rgb) in a sub-image.

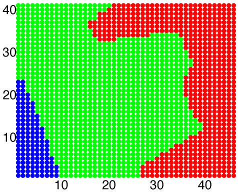
A comparison of all methods



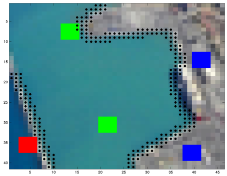
(a) COSC



(b) CSP

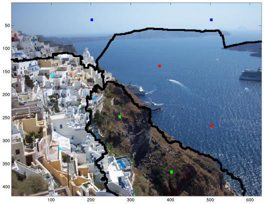


(c) FAST-GE

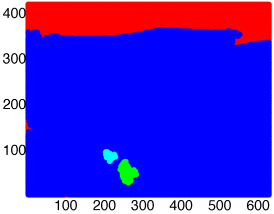


(d) FAST-GE

Santorini image - 250K pixels ($k = 4$)

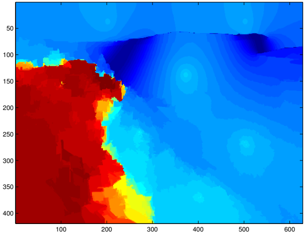


(a) FAST-GE (15.2 sec)

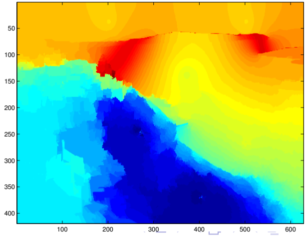


(b) COSf (263.6 sec)

Eig 1

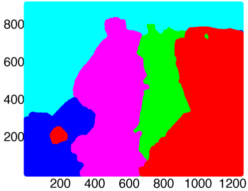


Eig 2

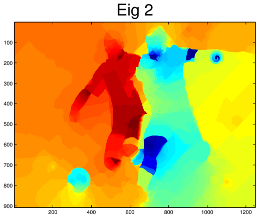


Soccer image - $\sim$ 1.1 million pixels ($k = 5$)

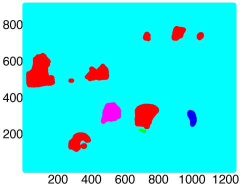
(a) Clean image



(b) FAST-GE (94 sec)



(c) FAST-GE



(d) COSf (25 min)