Hypergraph Clustering based on Game Theory

Ahmed Abdelkader, Nick Fung, Ang Li and Sohil Shah

University of Maryland

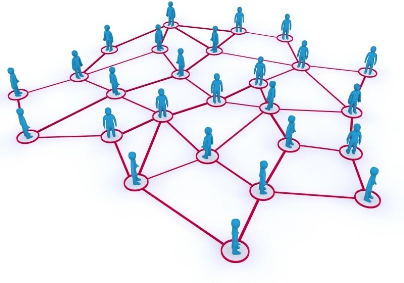
May 8, 2014

Overview

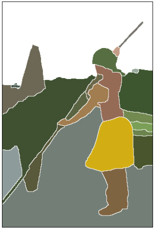
- Introduction
- Related Work
- Our Model
- Algorithm
- Conclusions

Clustering

(a) DNA



(b) Social Network

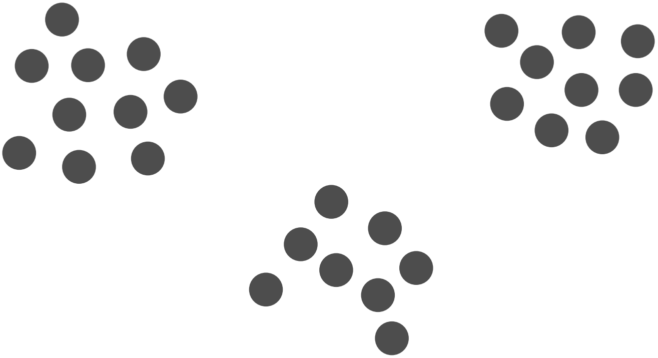


(c) Image

from Google Image

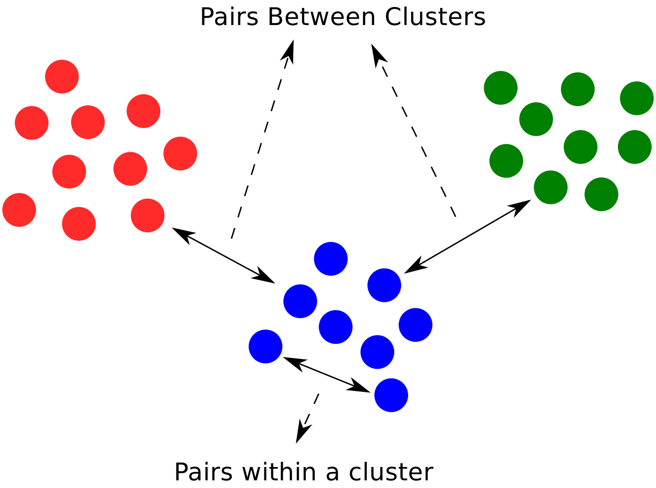
What is clustering?

Original data



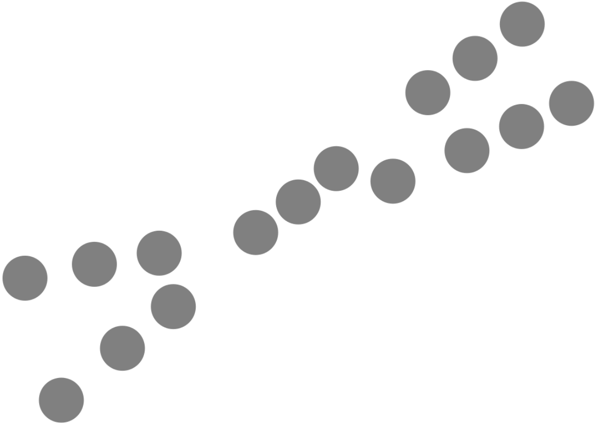
What is clustering?

Clustering using pairwise distances



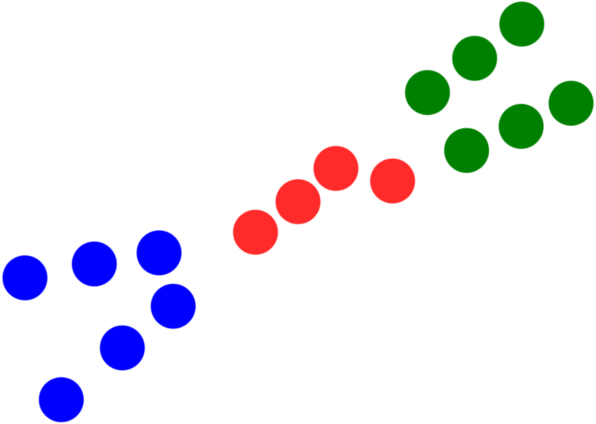
Pairwise distances are not enough

Another example



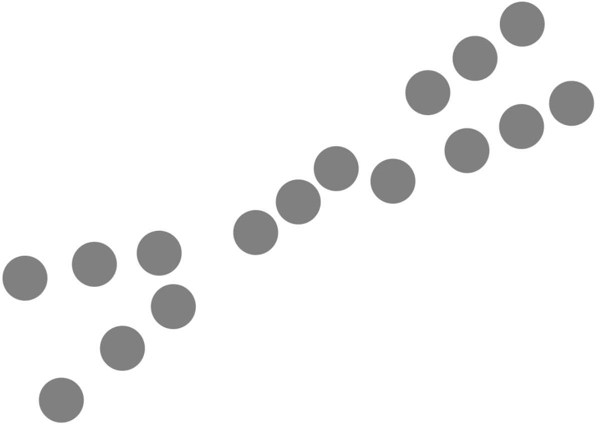
Pairwise distances are not enough

Clustering lines using pairwise distances



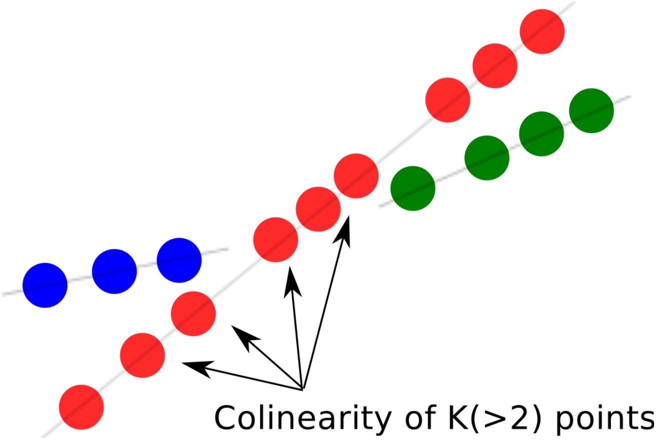
Pairwise distances are not enough

Another example



Pairwise distances are not enough

Clustering lines using measurement of more than 2 points



A general data representation: Hypergraph

Hypergraph is a generalization of a graph in which an edge can connect any number of vertices.

Definition (Hypergraph)

A hypergraph H is a pair $H = (V, E)$ where V is a set of elements called nodes or vertices, and E is a set of non-empty subsets of V called hyperedges or edges.

Definition (Weighted Hypergraph)

A weighted hypergraph $H(V, E, \omega)$ is a hypergraph where each hyperedge is associated with a weight defined by ω.

Definition (Weighted k-graph)

A weighted k-graph (aka k-uniform hypergraph) $H(V, E, \omega)$ is a weighted hypergraph such that all its hyperedges have size k.

The problem we address

Hypergraph Clustering

Given a k-graph $H(V, E, \omega)$ where for each vertex combinations $(v_1, v_2, \ldots, v_k) \in V$, the weight $\omega(v_1, v_2, \ldots, v_k) \in [0, 1]$ is defined by their similarity measure (the possibility that they come from the same cluster). The Hypergraph Clustering problem is to cluster the vertices from V into multiple clusters $\{C_1, C_2, \ldots\}$ (the total number of clusters is unknown) such that

1. each vertex belongs to one and only one cluster;
2. vertices from the same cluster have higher similarities;
3. vertices from different clusters have lower similarities.

Clustering is related to game theory

Non-cooperative games based approaches:

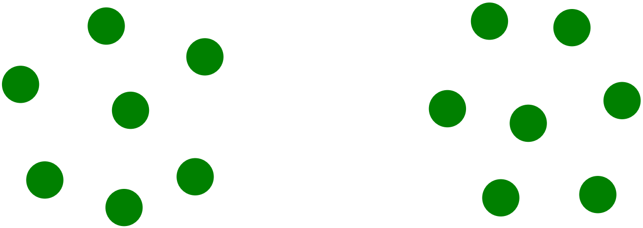
- Replicator dynamics
- Related works:
  - Rota Bulò and Pelillo, PAMI 2013 [1]
  - Donoser, BMVC 2013 [3]
  - Liu et al., CoRR 2013 [5]

Cooperative games based approaches:

- Shapley values
- Related works:
  - Garg et al., TKDE 2013 [4]
  - Dhamal et al., CoRR 2012 [2]

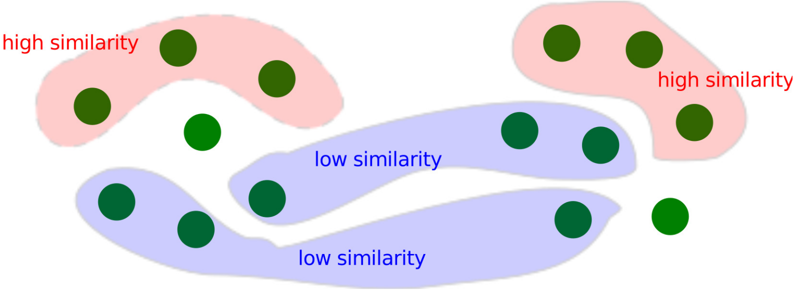
Non-cooperative games

Replicator Dynamics based approaches



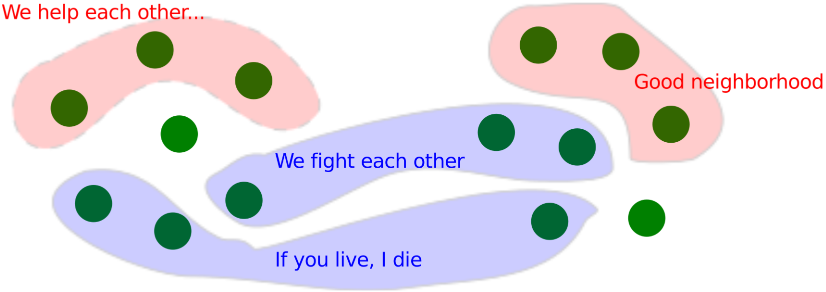
Non-cooperative games

Replicator Dynamics based approaches (Let $K = 3$)



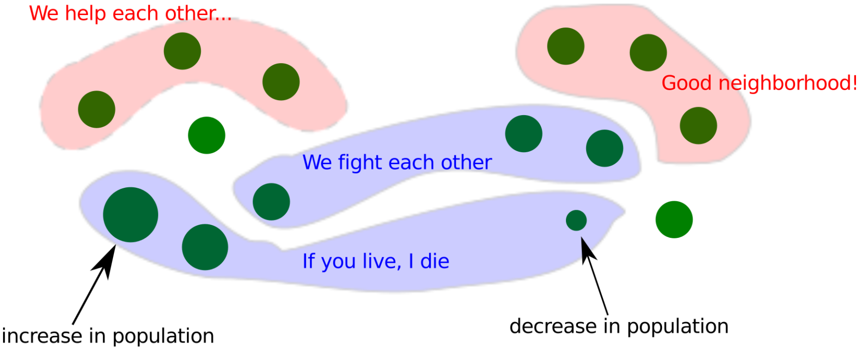
Non-cooperative games

Replicator Dynamics based approaches



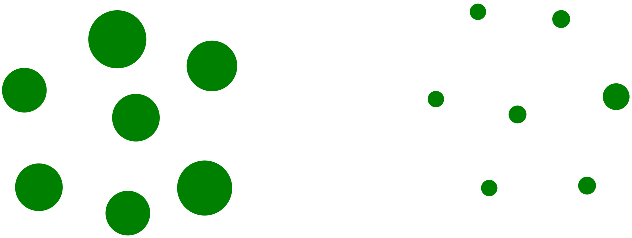
Non-cooperative games

Replicator Dynamics based approaches



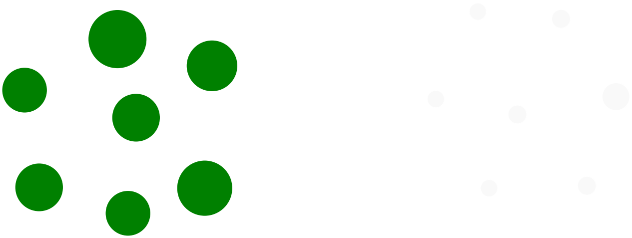
Non-cooperative games

Replicator Dynamics based approaches



Non-cooperative games

Replicator Dynamics based approaches



Hypergraph clustering

A general formulation

The problem of clustering a k-graph $H(V, E, \omega)$ can be mathematically defined as solving,

$$C^* = \arg\max_{\mathbf{C}} \; S(C) \tag{1}$$

$$\text{s.t.} \quad S(C) = \frac{1}{m^k} \sum_{e \in C : C \subseteq E} \omega(e) \tag{2}$$

where $S(C)$ is the cluster score.
This can be reformulated using an assignment vector,

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \; \sum_{e \in E} \omega(e) \prod_{v_i \in e} x_{v_i} \tag{3}$$

such that

$$\mathbf{x} \in \left\{0, \frac{1}{m}\right\}^N \quad \text{where} \quad \mathbf{x} = (x_1, x_2, \ldots, x_{|V|}) \tag{4}$$

- There are k players P $=\{1,2,,\ldots k\}$ each with N pure strategies S=$\{1,\ldots N\}$.
- The payoff function $\pi : S^k \mapsto \mathbb{R}$
- $\Delta = \{x \in \mathbb{R}^N : \sum_{j \in S} x_j = 1, x_j \geq 0, \forall j \in S\}$. Let $x^{(i)} \in \Delta$.
- The utility function of the game $\Gamma = (P, S, \pi)$ for any mixed strategy is given by,

$$u(x^{(1)}, \ldots x^{(k)}) = \sum_{(s_1, \ldots s_k) \in S^k} \pi(s_1, \ldots s_k) \prod_{i=1}^{k} x_{s_i}^{(i)} \tag{5}$$

Evolutionary Stable Strategy

- Find equillibrium $x \in \Delta$ s.t. every player obtains some expected payoff and no strategy can prevails upon others.
- For Nash equillibrium we get, $u(e^j, x^{[k-1]}) \leq u(x^{[k]}), \forall j \in S$
- Instead for any $y \in \Delta \setminus \{x\}$ and $w_\delta = (1-\delta)x + \delta y$ we need $u(y, w_\delta^{[k-1]}) < u(x, w_\delta^{[k-1]})$. This is ESS.

- Assumption: π is supersymmetric
- Payoff function

$$\pi(s_1, \ldots, s_k) = \frac{1}{k!}\omega(s_1, \ldots, s_k), \quad \forall \{s_1, \ldots, s_k\} \in E \qquad (6)$$

- Here N input data point is analogous to N pure strategies of k player game.
- The support of final ESS $\mathbf{x}$ correspond to the points belonging to that cluster.
- Solving for (3) is equivalent to finding maxima point of (5).(*)
- ESS cluster satisfies the two basic properties of cluster, Internal coherency and External incoherency.

Optimization Criteria

- Solving (5) optimally is NP-Hard.
- Observe that the function in (5) is homogeneous polynomial equation and thus it is a convex optimization problem.
- In [1], author proves that the Nash equilibria of game Γ are the critical points of $u(x^{[k]})$ and ESS are the strict local maximizers of $u(x^{[k]})$ over the simplex region.
- Performing **Projected** gradient ascent in Δ requires large number of iterations.

$$\left(u(x^{[k]}) = \sum_{(s_1,\ldots s_k)\in S^k} \pi(s_1,\ldots s_k) \prod_{i=1}^{k} x_{s_i} \right)$$

Baum-Eagon Algorithm

Any homogeneous polynomial f($\mathbf{x}$) in variable $x \in \Delta$ with nonnegative coefficients can be approximately solve using the following heuristics,

$$x_j^* = x_j \frac{\frac{\partial f(\mathbf{x})}{\partial x_j}}{\sum_{l=1}^{n} x_l \frac{\partial f(\mathbf{x})}{\partial x_l}} \tag{7}$$

Using this heuristics for solving (5), we obtain,

$$x_j(t+1) = x_j(t) \frac{d_j}{u(x(t)^{[k]})} \quad \forall j = 1, \ldots n \tag{8}$$

where $d_j = u(e^j, x(t)^{[k-1]})$ and $u(x(t)^{[k]}) = \sum_l x_l d_l$

Frank-Wolfe Algorithm

- Use ϵ-bounded simplex set Δ_ϵ s.t. $x \in [0, \epsilon]^N$.
- Initialize $\mathbf{x}(0) \in \Delta_\epsilon$, $t \leftarrow 0$.
- Iterate
    1. Compute $\mathbf{d}$.
    2. $\mathbf{y}^* \leftarrow \arg\max \mathbf{d}^T \mathbf{y} \quad s.t. \quad \mathbf{y} \in \Delta_\epsilon$.
    3. If $\mathbf{d}^T(\mathbf{y}^* - \mathbf{x}(t)) = 0$, return $\mathbf{x}(\mathrm{t})$.
    4. $\delta^* \leftarrow \arg\max u(w_\delta^{[k]}) \quad s.t. \quad w_\delta = (1-\delta)\mathbf{x}(t) + \delta\mathbf{y}^*$.
    5. $\mathbf{x}(t+1) \leftarrow w_{\delta^*}$

The overall complexity of each iteration of all the algorithm is $\mathcal{O}(N^k)$. Frank-Wolfe algorithm converges the fastest with an average of 10 iterations.

This page is intentionally left blank.

Reference

[1] Samuel Rota Bulo and Marcello Pelillo. A game-theoretic approach to hypergraph clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1312–1327, June 2013.

[2] Swapnil Dhamal, Satyanath Bhat, K. R. Anoop, and Varun R. Embar. Pattern clustering using cooperative game theory. *CoRR*, abs/1201.0461, 2012.

[3] Michael Donoser. Replicator graph clustering. In *Proceedings of British Conference on Computer Vision (BMVC)*, 2013.

[4] Vikas K. Garg, Y. Narahari, and M. Narasimha Murty. Novel biobjective clustering (bigc) based on cooperative game theory. *Knowledge and Data Engineering, IEEE Transactions on*, 25(5):1070–1082, May 2013.

[5] Hairong Liu, Longin Jan Latecki, and Shuicheng Yan. Revealing cluster structure of graph by path following replicator dynamic. *CoRR*, abs/1303.2643, 2013.