生命科学中的随机动力学模型 II

姚 远

2011.03.01

几种聚类算法比较

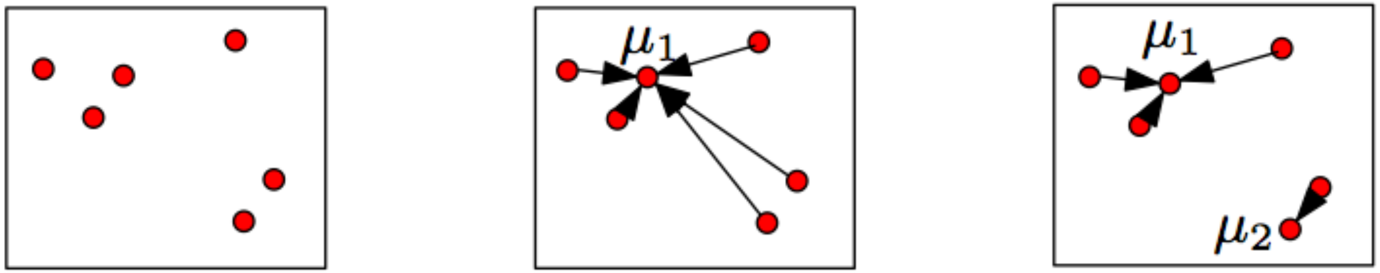
类别	复杂性	近似算法	在线算法	Hierarchical	统计一致性
K-means	NP	50-app	✗	✗	✔[Pollard81]
K-center	NP	2-app. O(kn)	✔ （8-app）	✔ （8-app）	✗ (metric net)
Average-linkage	Close to k-means	?	?	✔	?
Complete-linkage	Close to k-center	a(k)-app k<a(k)<k^log(3)	?	✔	?
Single-linkage	Minimal spanning tree	…	✔ （Persistent Homology）	✔	✔ [Hartigen81, Stuetzle03]

Recall K-center clustering

- input: conformations in a metric space (RMSD) and a number k
- goal: obtain a partition of the points into clusters $C_1, \cdots, C_k$ with centers $\mu_1, \cdots, \mu_k$.
  - condition: minimize the maximum cluster radius:

$$\max_{i} \max_{x \in C_i} d(x, \mu_i)$$

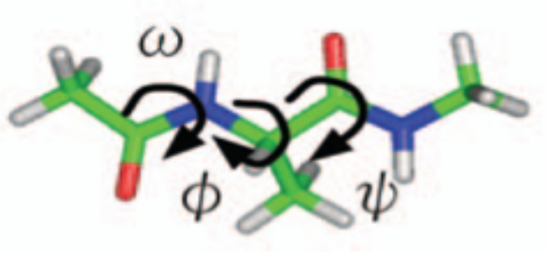
- NP-hard problem
- 2-approximation algorithm (greedy k-center algorithm)



K-center 几何性质

- Farthest-first-traversal算法形成了样本空间的一个度量R-net
  - Any two points in C are R-distance away
  - Points in C form a R-cover of sample space
- K-center is NP-hard, but the 2-approx. algorithm is O(kn), much faster than K-means etc.
- 只依赖于度量结构
- K-center在ISOMAP(TdL’2000, Science)中被采用，称为Landmark技术
- Molecular dynamics application [Sun, Y, Huang, et al. JPC, 09]
- 缺点：
  - 对样本空间边缘的outlier和noise比较敏感 (Good or bad?)
  - 没有statistical consistency theory

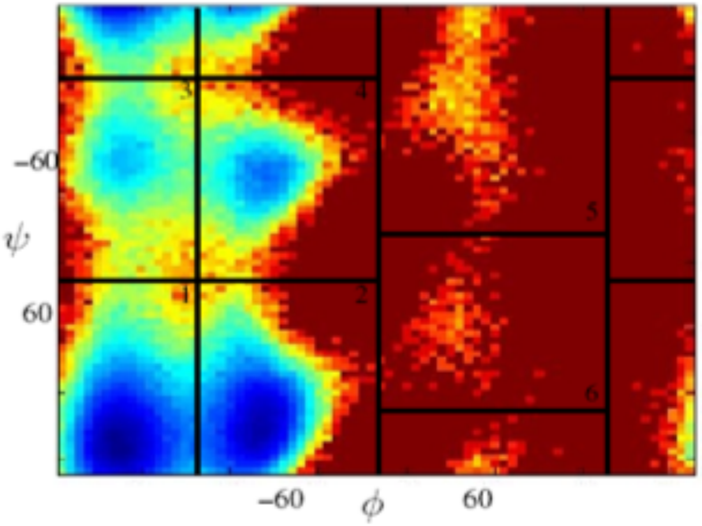
Application I: Alanine-dipeptide



[Chodera et al. 2007]

975 trajectories

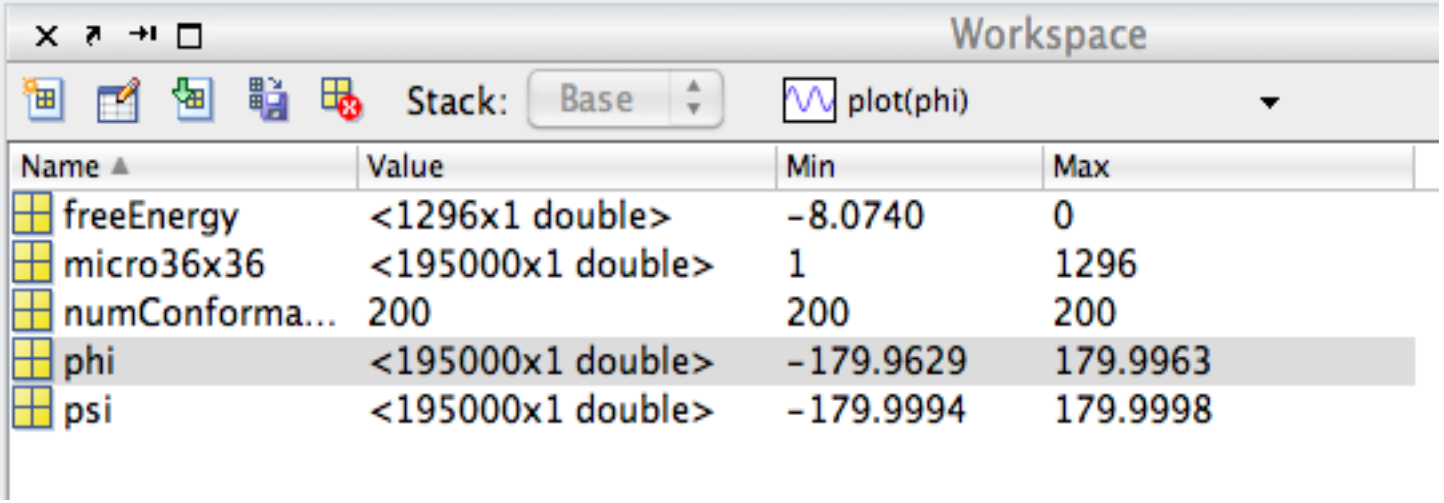
200 conformations per trajectory



density on $\phi - \psi$ plane

Phi-Psi Matlab 数据

```
>> load ../data/alanine_dipeptide_phi-psi.mat
% phi, psi: reaction coordinates of 195000 points
% micro36x36: a map to 36x36 torus cell index
% freeEnergy: 1296 (=36x36) vector, free energy estimation for each cell
```

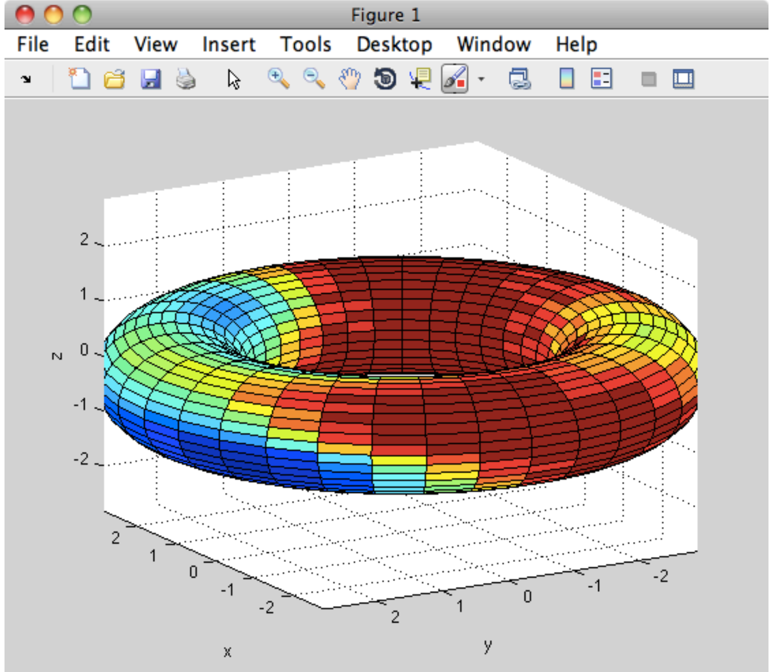


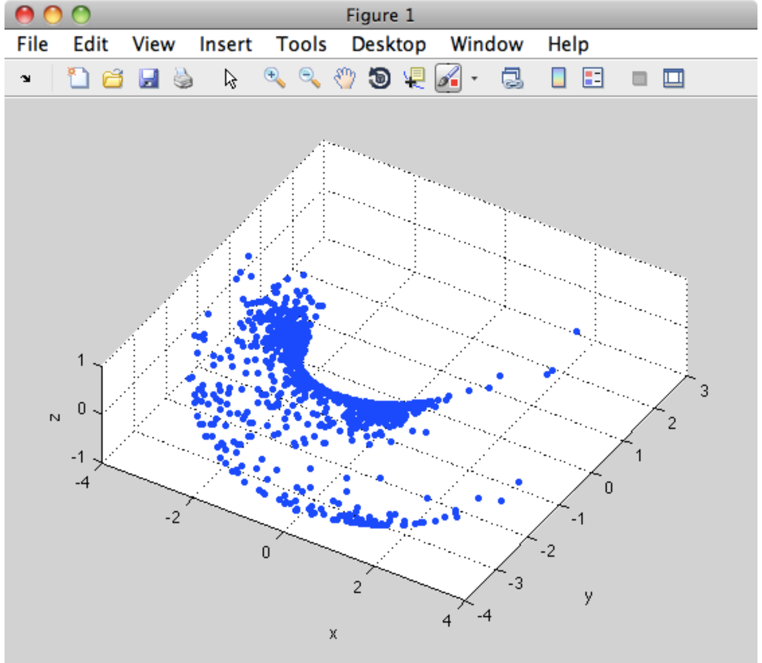
Stack: Base

Name	Value	Min	Max
freeEnergy	<1296x1 double>	-8.0740	0
micro36x36	<195000x1 double>	1	1296
numConforma...	200	200	200
phi	<195000x1 double>	-179.9629	179.9963
psi	<195000x1 double>	-179.9994	179.9998

Torus Embedding

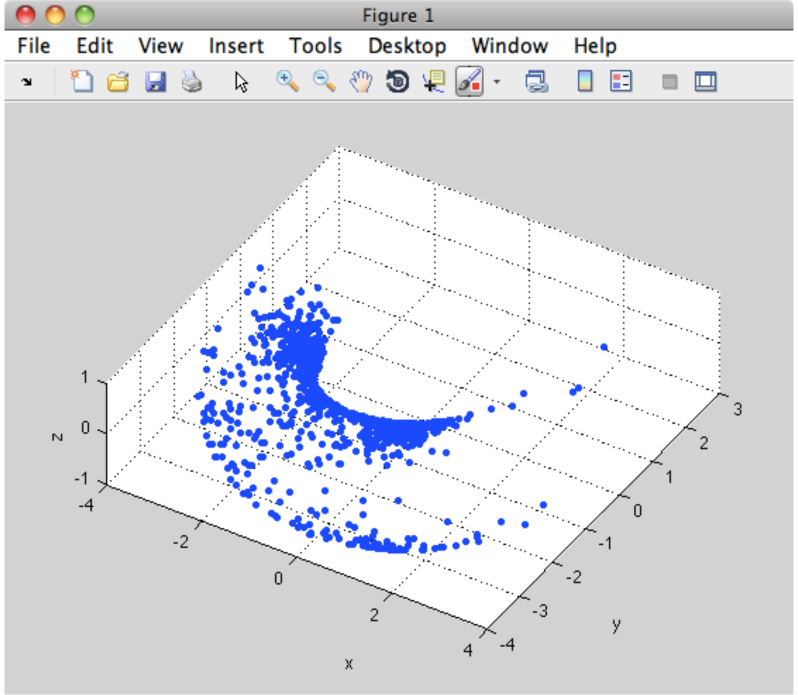
```
>> [x,y,z]=embedTorus(3,1,phi,psi);
>> freeEnergyTorus;
>> idx=randperm(length(phi));
>> scatter3(x(idx(1:1000)),y(idx(1:1000)),z(idx(1:1000)),'.')
```

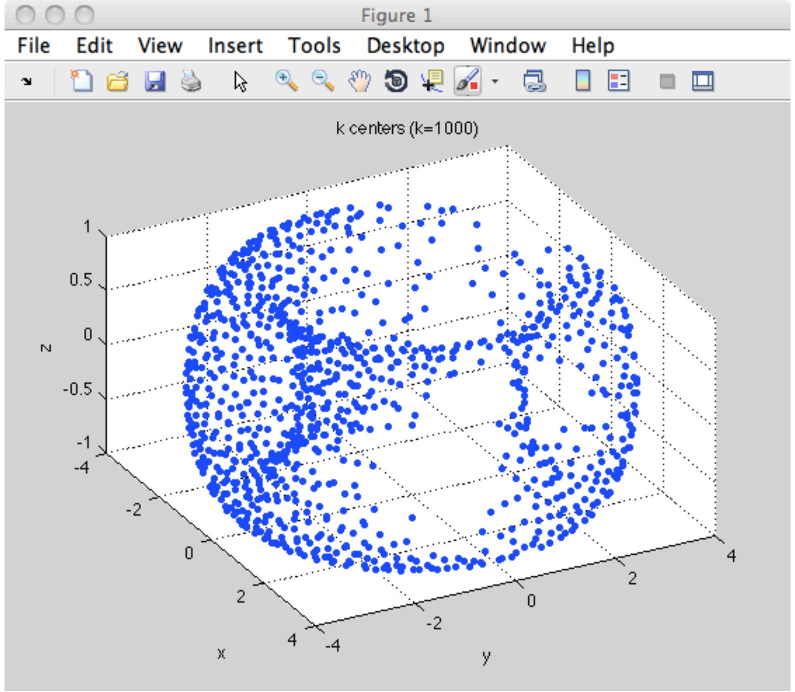




Random vs. Kcenter

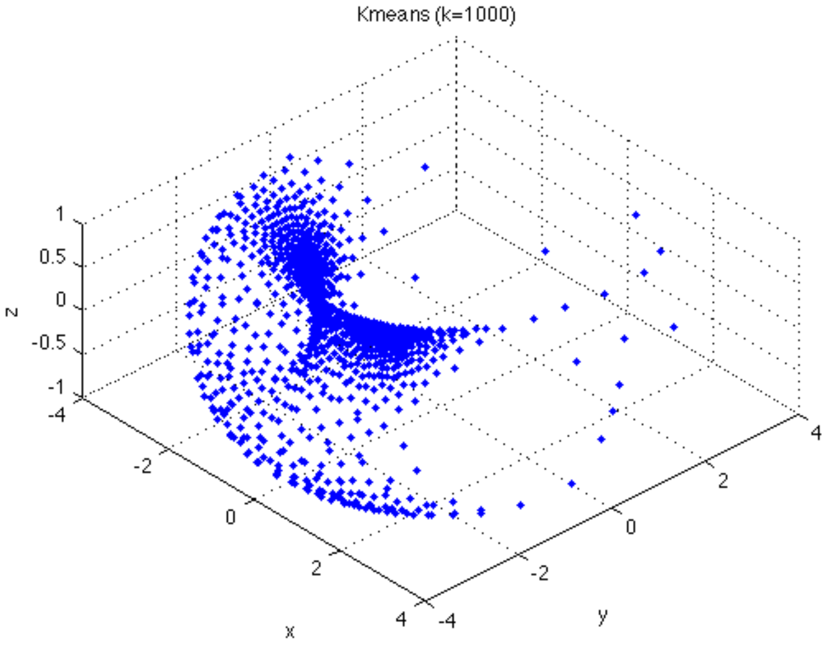
```
>> idx=randperm(length(phi)); % 随机采样
>> scatter3(x(idx(1:1000)),y(idx(1:1000)),z(idx(1:1000),'.')
>> L=kcenter([x,y,z],1000); % 笔记本上需要几分钟…
>> scatter3(x(L),y(L),z(L),'.')
```

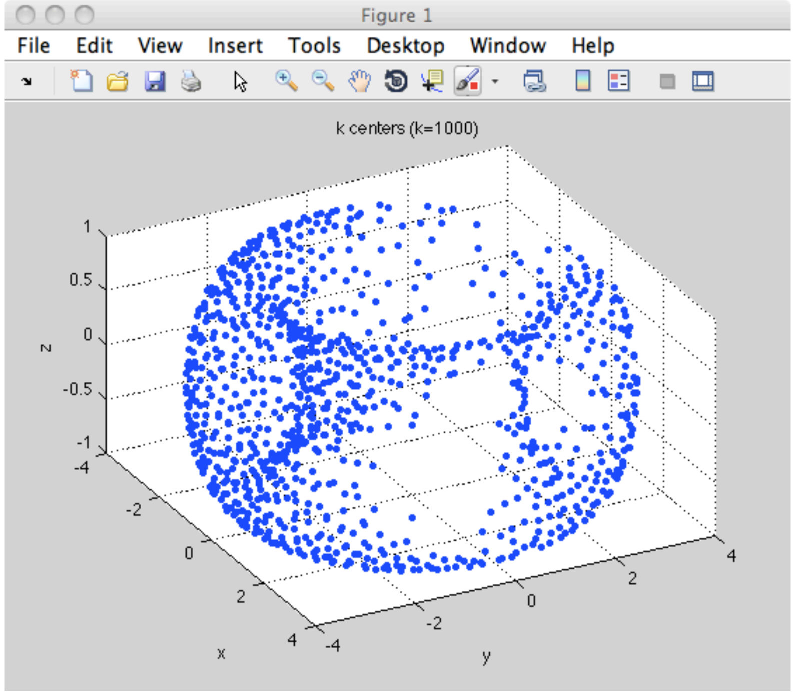




Kmeans vs. Kcenter

```
>> [idx,C]=kmeans([x,y,z],1000); % Kmeans (k=1000) about 10
   times running time of kcenter
Warning: Failed to converge in 100 iterations.
>> scatter3(C(:,1),C(:,2),C(:,3),'.')
>> L=kcenter([x,y,z],1000);
>> scatter3(x(L),y(L),z(L),'.')
```





Demo Kcenter

```
>> demo_ala_kcenter

…
•   % initial choice of L, DL – distance from data to landmarks
•   L = seed;
•   DL = zeros(n, k);
•   DL(:,1:length(L)) = dist2(X,X(L,:));     % Euclidean distance
•
•   % Farthest-First-Traversal, or maximin search
•   DLmin = min(DL(:,(1:length(L))), [], 2);
•   r = zeros(k,1);
•   for a = (length(seed)+1: k),
•     [r(a-1), newL] = max(DLmin, [], 1);
•     L = [L; newL];
•     DL(:,a) = dist2(X, X(newL,:));
•     DLmin = min(DLmin, DL(:,a));
•   end
```

Kcenter.m

```
•   function [L, R, IDX, C, DL]=kcenter(X,k,L0,EorD)
•   % Farthest-First Traversal Algorithm as a 2-approximation for  kcenter clustering
•   %   [L,R,IDX,C,DL] = KCENTER(X,k,L0,EorD)
•   %
•   % INPUT:
•   %   X - see description for input EorD.
•   %   k - the number of centers to be chosen.
•   %   L0 - the first centroid index.
•   %   EorD - character EorD determines how input matrix is interpreted. If
•   %       EorD is 'e', then the N x p input matrix X is interpreted as N
•   %       points in R^p. If EorD is 'd', then the N x N input matrix X is
•   %       interpreted as the distance matrix for N points in an arbitrary
•   %       metric space.
•   %
•   % OUTPUT:
•   %   L - an p-by-1 vector containing indices of each landmark.
•   %   R - covering radius, i.e. the smallest number such that every data
•   %       point lies within distance R of a landmark point.
•   %   IDX - an N-by-1 vector containing the cluster indices of each point.
•   %   C - a k-by-p matrix for the k cluster centroid locations.
•   %   DL - an N-by-k matrix of distances from each point to every centroid.
•   %
```

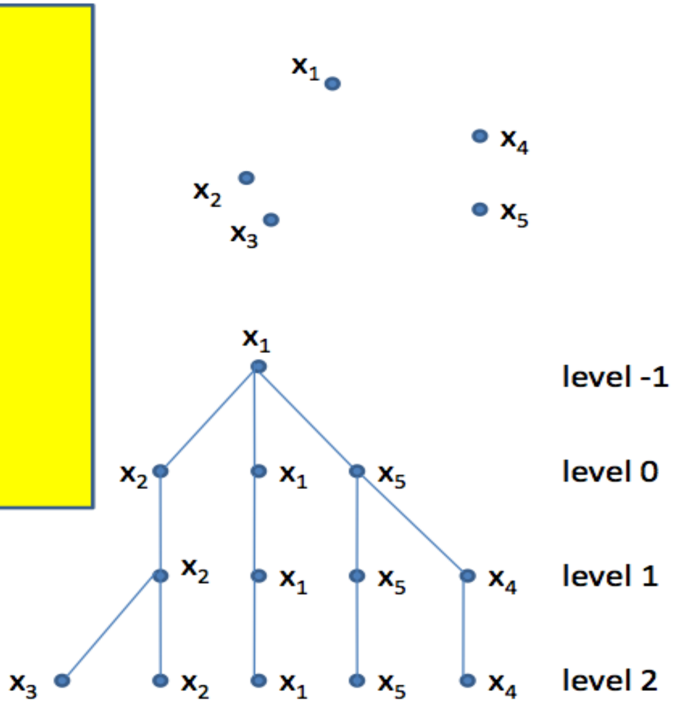
Homework

- EASY: 自己生成一个 dataset (比如圆附近的随机点，混合高斯分布点), 比较 kcenter、kmeans、linkage (single, complete, average).

- CHALLENGE: 用自己喜欢的计算机语言实现一个 online kcenter 算法。(optional)

Build online! When new point x arrives:

1. Find largest j such that x is within dist $1/2^j$ of some node p at level j
2. Make x a child of p



Full-Atomic Coordinates

```
>> load ../data/alanine_dipeptide_traj_coords.mat
% natom is 22, number of atoms
% nconf is 195000, number of conformations
% confs is 22x585000 double [x1,y1,z1,x2,y2,z2,
    …]
```

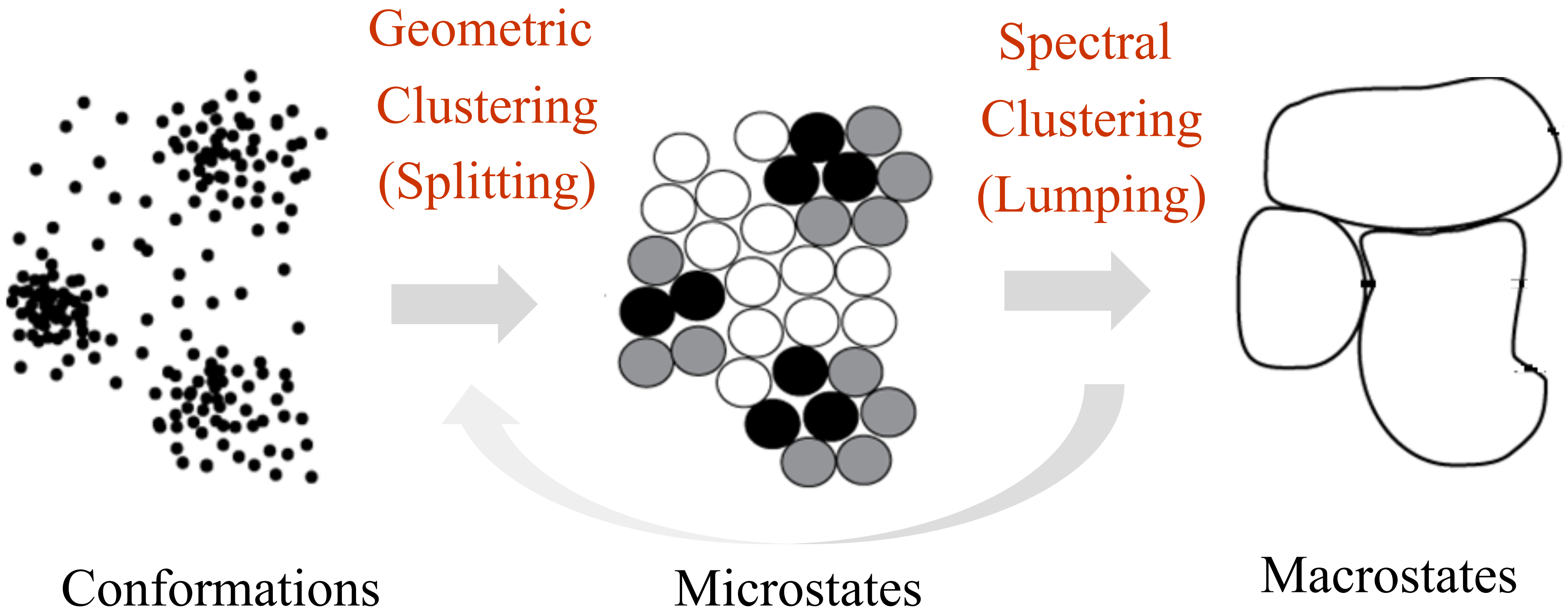
欧氏距离需要RMSD距离

特殊package：bio-basics (Jian SUN), MSMBuilder (Greg Bowman)

K-center in Molecular Dynamics

- Simple
- Fast
  - Generate thousands of clusters from millions of conformations within several hours from a single machine
  - 20-60 times faster than K-means
- Online and hierarchical algorithms (cover-tree)
- Clusters have approximately equal radii, whence cluster population provides a density estimation in systems of intrinsic low dimension
  - Note: accurate density estimation in high dimensional space (>10) is extremely difficult (Open: optional Polya Tree may work, Wong-Ma 2010, tell me if you wanna try this!)

Clustering in Biomolecular Dynamics



Geometric Clustering (Splitting)

Spectral Clustering (Lumping)

Conformations

Microstates

Macrostates

K-center Clustering with RMSD metric:

Form an epsilon-net to cover the sampled space

Spectral Clustering with Transition Counts:

Find non-spherical metastable states

谱聚类分析
Spectral Clustering

When we should not use K-means

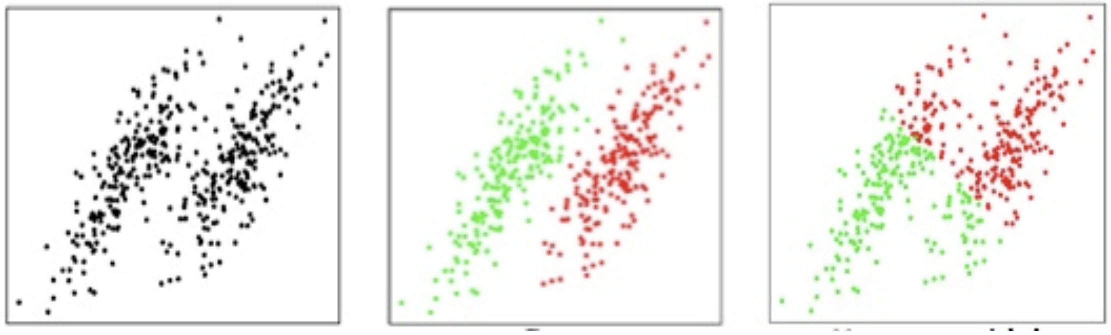
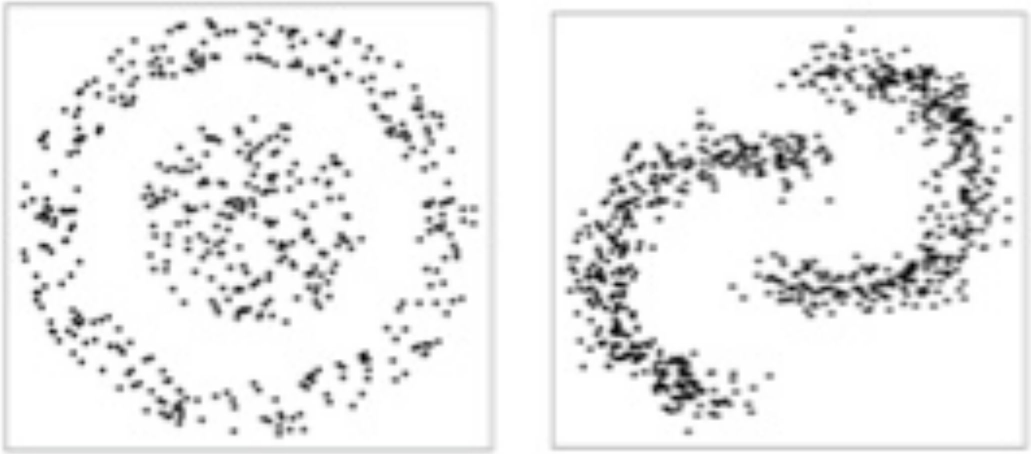


Figure: (a) data, (b) 2 clusters, (c) K-means with k=2

K-means requires any two points within the cluster close to each other.

K-means does NOT work for non-Gaussian (non-spherical) shape clusters.

Single-Lingkage & Spectral Clustering

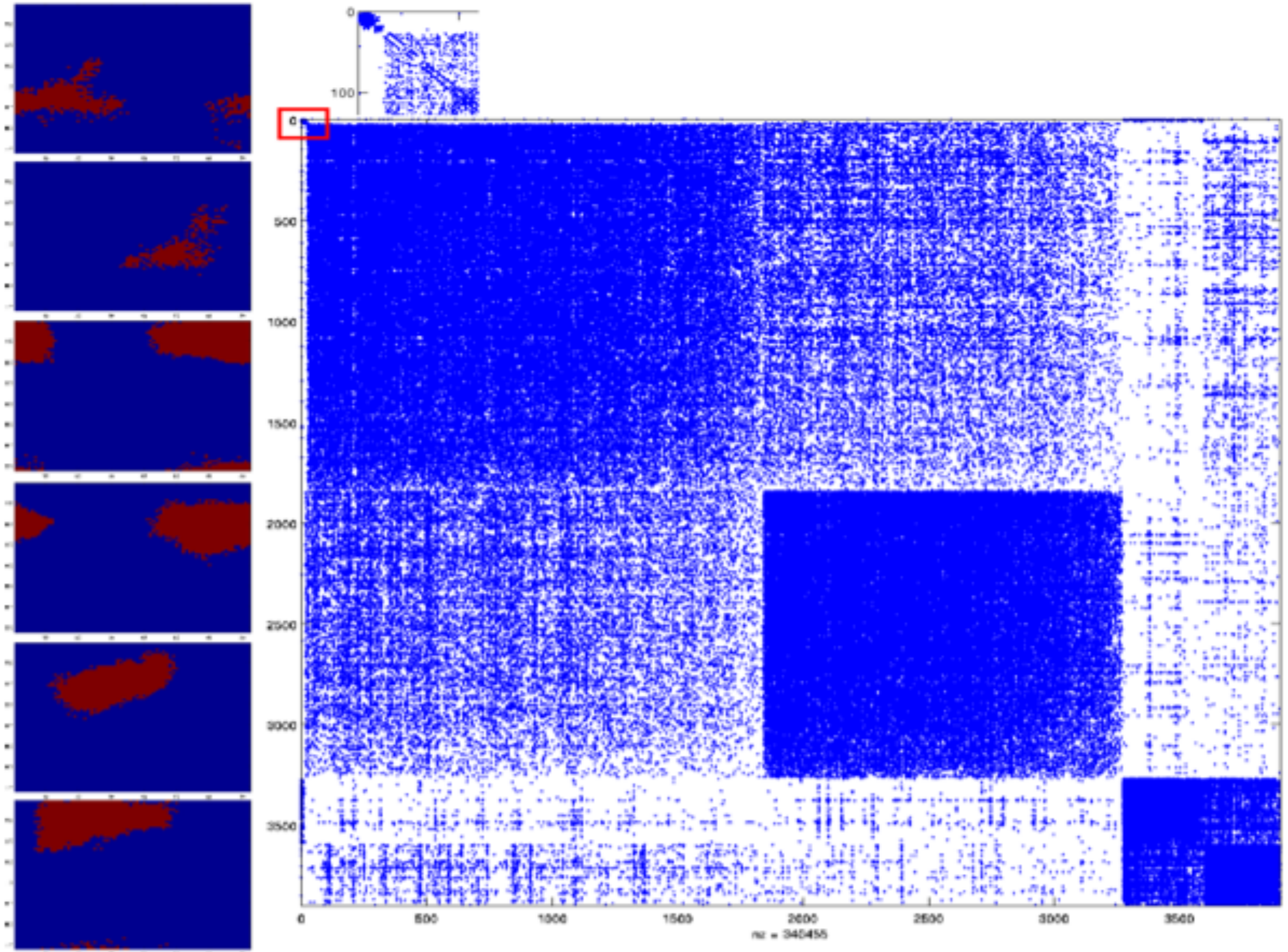


For non-Gaussian (non-spherical) shape clusters, two points within the same cluster are connected by a densely sampled path, but not necessarily close to each other

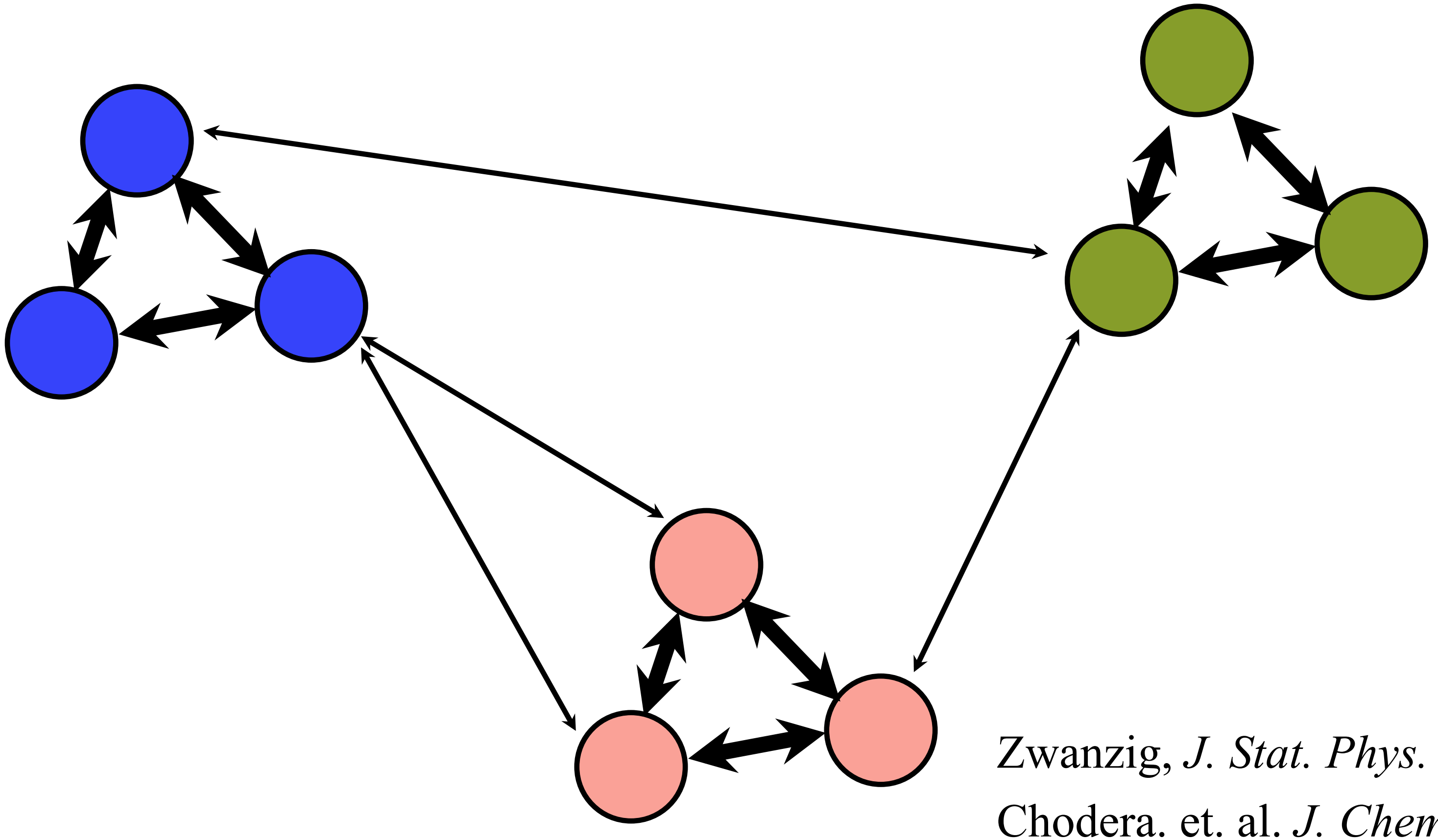
Cluster are connected components in some neighborhood graph

Single-linkage or spectral clustering are suitable to capture them

Block Structure of Transition Matrix



Conformational Dynamics: Nearly Uncoupled Markov Chains



Zwanzig, *J. Stat. Phys.* 1983

Chodera. et. al. *J. Chem. Phys.* 2007

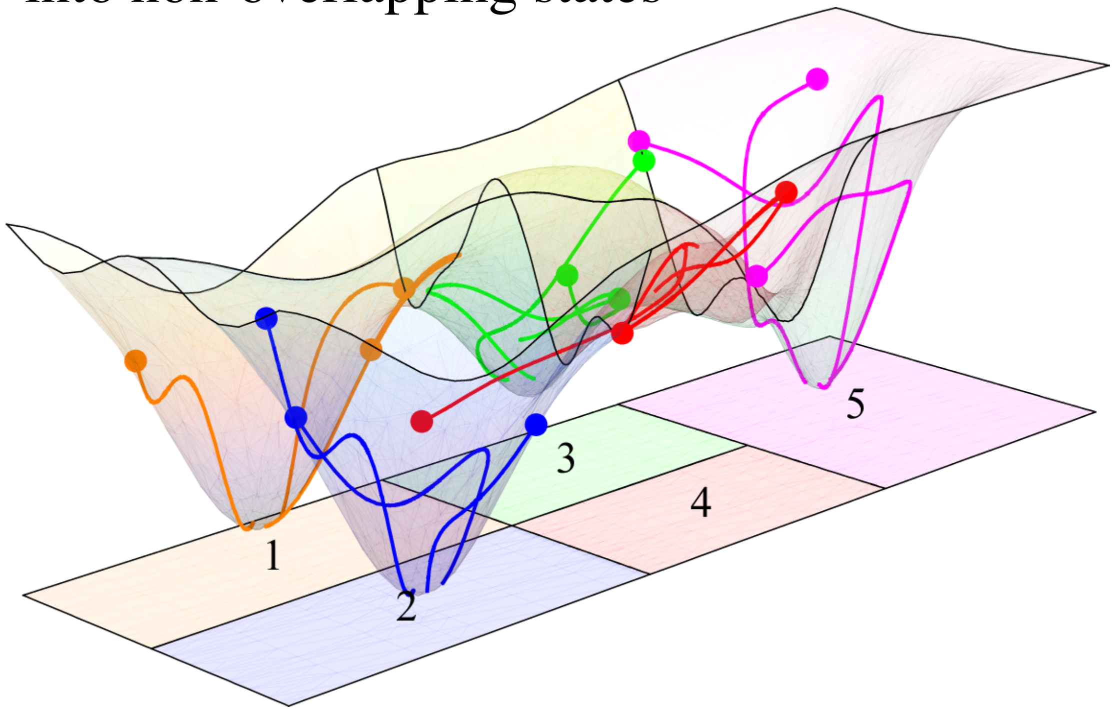
Noé. et.al. *J. Chem. Phys.* 2007

Huang et.al. 2009, Hummer, Shuttle....

Figure Courtesy John Chodera

Markov State Models (MSMs)

The configuration space is decomposed into non-overlapping states



Define transition probabilities between states

$$T(\tau) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{15} \\ p_{21} & p_{22} & & \\ \vdots & & \ddots & \\ p_{51} & & & p_{55} \end{bmatrix}$$

We can extract long time dynamics from MSMs built from short simulations

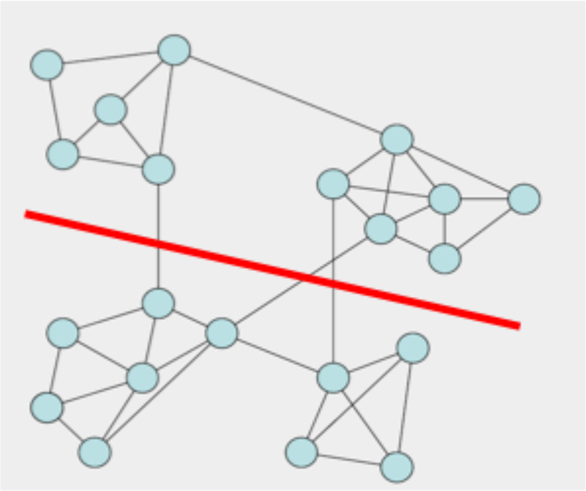
$$P(n\tau) = [T(\tau)]^n P(0)$$

The time is coarse-grained in τ

Graph Partition Problem

- goal: find a cut with the smallest Cheeger ratio (conductance)
  - For $S \subset V$, volume of S: $vol(S) = \sum_{v \in S} d_v$
  - $\partial S = \{(u,v) \in E : u \in S \& v \in S\}$
  - Cheeger ratio of S, $h(S) = \frac{|\partial S|}{\min\{vol(S), vol(G) - vol(S)\}}$

- applications
  - clustering
  - segmentation
  - task partitioning for parallel processing
  - a preprocessing step to divide-and-conquer algorithms



Two Theories on Spectral Clustering

- Lumpable Markov Chains
- Graph Minimal Cut

Lumpability of Markov Chains

- Let T be the transition matrix of a Markov chain defined on n states S={1,…,n}.
- P={S_1,…,S_k} is a partition of S into k macrostates.
- Sequences {x_0,…,x_t,…} generated by T, i.e.

$$\mathrm{Prob}(x_t=j\ ;\ x_{t-1}=i)= T_{ij}$$

- Induced dynamics: relabel x_t by y_t from corresponding states in partition P
- [Kemeny-Snell'76] T is called *lumpable* if

$$\mathrm{Prob}(y_t=k_0;\ y_{t-1}=k_1,\ \ldots,y_{t-m}=k_m) = \mathrm{Prob}(y_t=k_0;\ y_{t-1}=k_1)$$

i.e. the induced dynamics is markovian.

Lumpability of Markov Chains

- [Kemeny-Snell'76] T is *lumpable* w.r.t. partition P= {S_1,…,S_k} iff for any s, t chosen from P, and for any i, j lying in S_s, the following holds

$$T_{it} = T_{jt}$$

where $T_{it} = \mathrm{sum}_{k\ \varepsilon\ St}\ T_{ik}$.

Spectral Theory of Lumpability

- [Meila-Shi 2001] T is *lumpable w.r.t. P* iff T has k independent piece-wise constant right eigenvectors in the span of characteristic functions of $P=\{S_1,\ldots,S_k\}$.
- Special case: If T is block diagonal, i.e. uncoupled Markov chain, then T is lumpable with piece-wise constant right eigenvectors associated with multiple eigenvalue 1.
- [Belkin-Shi-Yu 2009] If T is close to being block diagonal, then there are top (k) eigenvectors which fix signs within the block.
- [E-Li-Vanden_Eijnden 2007] Let T be an n-dim reversible Markov chain, then the best approximation of T from k-dim lumpable chains solves the following optimization

  $$\mathrm{Min}_Q \ \mathrm{norm}(T-Q,\text{`Hilbert-Schmidt'})$$

  where the Hilbert-Schmidt norm of a reversible chain $T = D^{-1}W$, is defined to be sqrt((DT)'(DT))=sqrt(W'W).

Spectral Clustering Algorithm

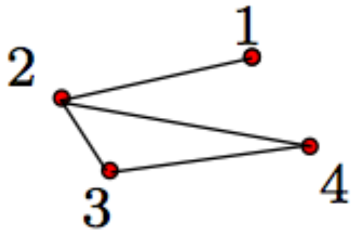
- Typical spectral algorithm to find lumpable states in nearly uncoupled systems [Ng-Jordan-Weiss'02]:
  - Find top k right eigenvectors of T where a large spectral gap occurs, $v_1,\ldots,v_k$
  - Embed the data into R^k by those eigenvectors
  - Use k-means (or alternatives) to find k clusters in R^k
- In biomolecular dynamics, this type algorithm is named after Perron, or PCCA [Weber'04].

Note there are issues when using with k-center here!

Graph Laplacian Operator

- given an undirected graph G=(V, E),
  - Adjacency matrix A:

$$A(u,v) = \left\{ \begin{array}{ll} 1 & \text{if } u \sim v \\ 0 & \text{o.w.} \end{array} \right\}$$



  - Diagonal degree matrix $D = diag(d_{v_1}, \cdots, d_{v_n})$
  - Graph Laplace Operator $L = D^{-1}(D - A)$
  - Tranistion probability matrix $W = D^{-1}A = I - L$,
  - $Wv = \lambda v$ implies $Lv = (1 - \lambda)v$
  - 1 is the largest eigenvalue for W; 0 is the smallest eigenvalue for L.

Graph Partition Problem

- Rayleigh quotient $R(f) = \frac{\sum_{u \sim v} (f(u) - f(v))^2}{\sum_u f^2(u) d_u}$ for $f \neq 0$
  - find a boolean function f minimizing $R(f)$ $\Leftarrow$ NP-complete
  - RELAXATION: find a real valued function f minimizing $R(f)$
  - $R(f) = \frac{< f, (D-A) f >}{< f, D f >}$
  - $\lambda_1 = \inf_f R(f) \Rightarrow \lambda_1$ and f are the first nonzero eigenvalue and eigenvector of L.

How good is this relaxation? Cheeger inequality

Cheeger Inequality

$$2h_G \geq \lambda_1 \geq \frac{h_f^2}{2} \geq \frac{h_G^2}{2}.$$

- f is the eigenvector of L corresponding to λ_1
- h_G is the smallest conductance (Cheeger ratio) of graph G
- h_f: the minimum Cheeger ratio determinded by a sweep of f
  - order the vertices: $f(v_1) \geq f(v_2) \geq \cdots \geq f(v_n)$.
  - $S_i = \{v_1, \cdots, v_i\}$
  - $h_f = \min_i h_{S_i}$
- find a partition whose conductance is within $2\sqrt{h_G}$

Local Cheeger Inequality

- Heat kernel $H_t = e^{-tL} = I - tL + \cdots + (-1)^k \frac{t^k}{k!} L^k + \cdots$
  - induce a random walk on the graph
  - a version of local Cheeger inequality for local graph partition
  - see the following references for details:
    - "A local graph partitioning algorithm using heat kernel pagerank," WAW 2009, LNCS 5427, (2009), 62-75
    - "The heat kernel as the pagerank of a graph," PNAS, 105 (50), (2007), 19735–19740.

Spectra of Graph Laplacians

- Graph min-cut is NP-hard
- However one can find a polynomial approximation via second eigenvector of normalized graph Laplacian
- Graph Laplacian is symmetric diagonal dominant (SDD)
- [Spielman-Teng 2009, Koutis-Miller-Peng 2010] SDD has fast solver with preconditioners

Reference

- Shi, Belkin, and Yu, Data spectroscopy: Eigenspaces of convolution operators and clustering. Annals of Statistics, 37 (6B): 3960-3984. 2008.
- Chodera, J. D., Singhal, N., Pande V. S., Dill, K. A., and Swope W. C. (2007) *J. Chem. Phys., **126, 155101-.***
- **E, Li, and** Vanden_Eijnden. Optimal partition and effective dynamics of complex networks. PNAS, 105 (23): 7907–7912, 2008.
- T.F. Gonzalez. Clustering to minimize the maximum intercluster distance. Theoretical Computer Science, 38:293-306, 1985.
- J.A. Hartigan. Consistency of single linkage for high-density clusters. Journal of the American Statistical Association, 76:388-394, 1981.
- Kemeny and Snell 1976. Finite Markov Chains. Springer-Verlag.
- Koutis, Miller, and Peng. Approaching Optimality For Solving SDD Linear Systems, 2010.
- Meila and Shi, A random walk view of spectral segmentation, AISTATS 2001.
- D. Pollard. Strong consistency of k-means clustering. Annals of Statistics, 9(1):135-140, 1981
- W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. Journal of Classification, 20(5):25-47, 2003.
- Spielman and Teng. Nearly-Linear Time Algorithms for Preconditioning and Solving Symmetric, Diagonally Dominant Linear Systems, **arXiv:cs/0607105v4, 2009**.