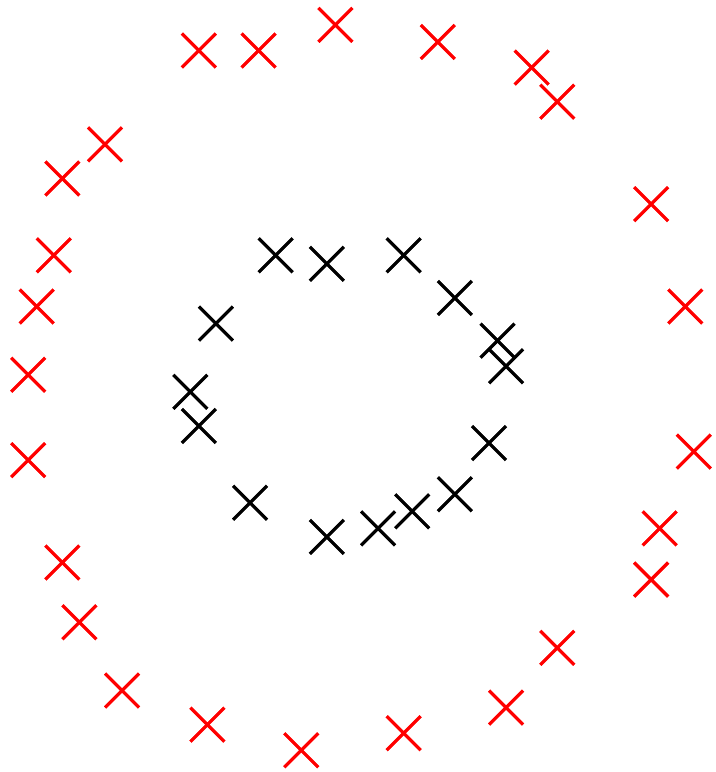
Spectral Clustering

Virginia de Sa
desa at cogsci

Spectral Clustering

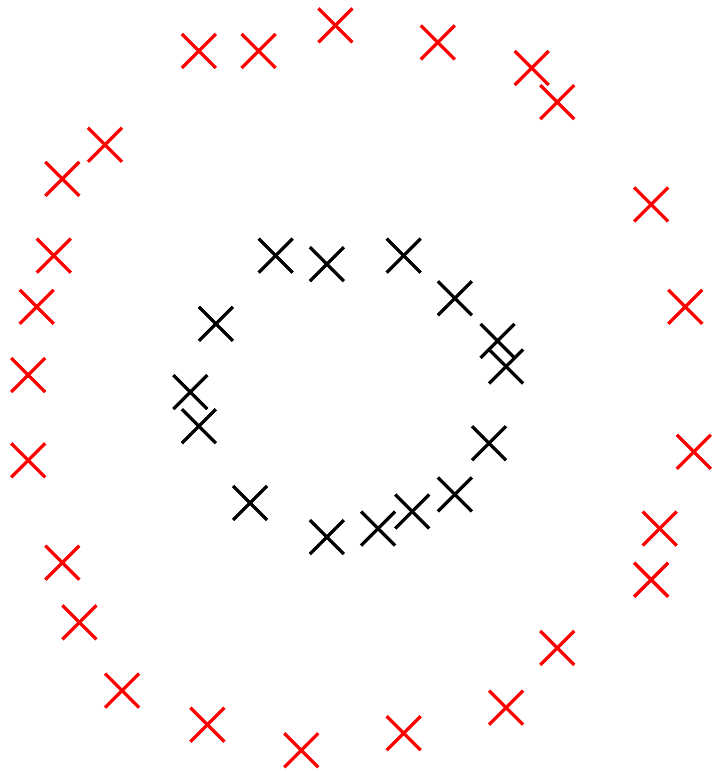
How do you deal with clustering this kind of data?



Spectral clustering clusters based on following data density based on graph theory ideas

Spectral Clustering

- Spectral clustering clusters based on pairwise proximity/similarity/affinity
- Clusters do not have to be Gaussian or compact



Spectral Clustering

We represent the data as a full set of pairwise similarities. (This set can be written as a matrix).

We usually get the similarity matrix using the Gaussian kernel that you saw earlier.
$W(i,j) = exp(-||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2/2\sigma^2)$

This matrix can be represented as a graph where the weight of the edge represents the similarity between the two nodes.

Graph cuts

Cut the graph in to n pieces while minimizing the number of edges (weight of edges) cut.

Min-Cut - find set of edges of minimal weight to disconnect the graph (tends to do lopsided cuts– e.g. cut off a vertex)

Normalized Cut [Shi & Malik 1997]

$$\text{Normalized Cut}(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(A, V)} + \frac{\text{cut}(A, B)}{\text{asso}(B, V)}$$

asso(A,V) $=$ sum of all weights from nodes in A

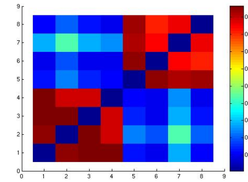
cut(A,B) $=$ sum of all weights between nodes in A and nodes in B

Ng, Jordan, Weiss Spectral Clustering Algorithm

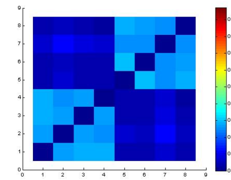
- Form the similarity/affinity matrix $W(i,j) = exp(-||x_i - x_j||^2/2\sigma^2)$
- Set the diagonal entries $W(i,i) = 0$
- Compute the normalized graph Laplacian as $L = D^{-.5}WD^{-.5}$ where D is a diagonal matrix with $D(i,i) = \sum_j W(i,j)$
- Compute top k eigenvectors of L and place as columns in a matrix X
- Form Y from X by normalizing the rows of X
- Run kmeans to cluster the row vectors of Y
- pattern x_i is assigned to cluster α iff row i of Y is assigned to cluster α

Ng, Jordan, Weiss Spectral Clustering Algorithm

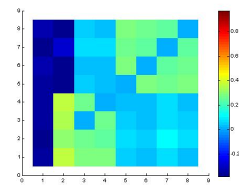
- Form the similarity/affinity matrix $W(i,j) = exp(-||x_i - x_j||^2/2\sigma^2)$



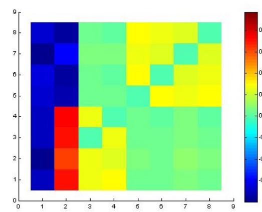
- Set the diagonal entries $W(i,i) = 0$
- Compute the normalized graph Laplacian as $L = D^{-.5}WD^{-.5}$ where D is a diagonal matrix with $D(i,i) = \sum_j W(i,j)$



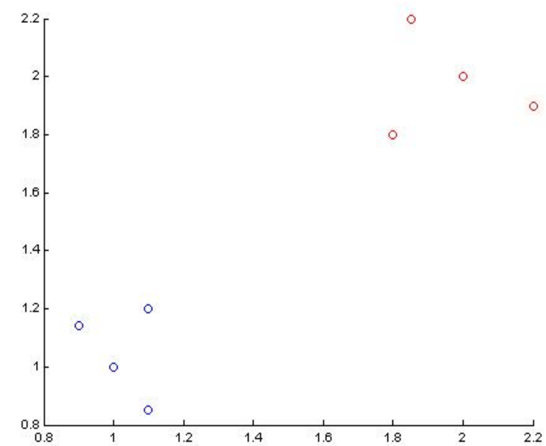
- Compute top k eigenvectors of L and place as columns in a matrix X



- Form Y from X by normalizing the rows of X

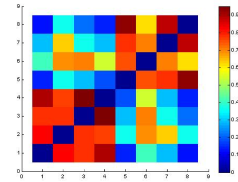


- Run kmeans to cluster the row vectors of Y
- pattern x_i is assigned to cluster α iff row i of Y is assigned to cluster α

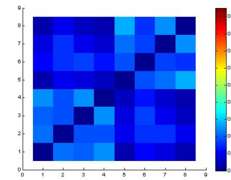


Ng, Jordan, Weiss Spectral Clustering Algorithm

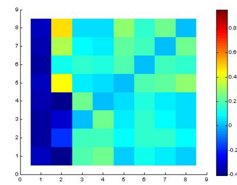
- Form the similarity/affinity matrix $W(i,j) = exp(-||x_i - x_j||^2/2\sigma^2)$



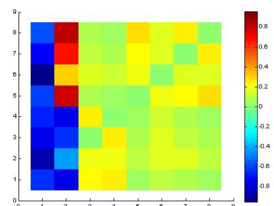
- Set the diagonal entries $W(i,i) = 0$
- Compute the normalized graph Laplacian as $L = D^{-.5}WD^{-.5}$ where D is a diagonal matrix with $D(i,i) = \sum_j W(i,j)$



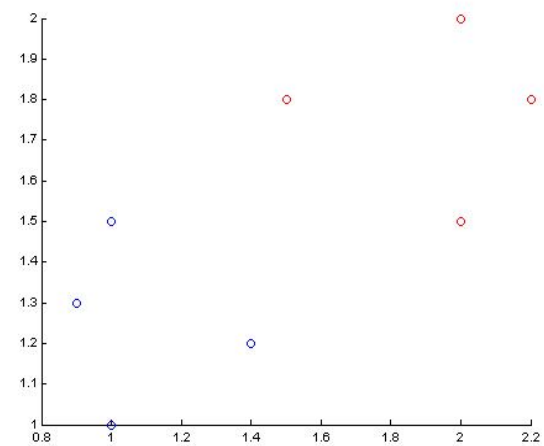
- Compute top k eigenvectors of L and place as columns in a matrix X



- Form Y from X by normalizing the rows of X



- Run kmeans to cluster the row vectors of Y
- pattern x_i is assigned to cluster α iff row i of Y is assigned to cluster α



Spectral Clustering

Can be shown to be a relaxation of the Normalized Cut on the similarity graph

$$\text{Normalized Cut}(A, B) = \frac{\text{cut}(A, B)}{\text{asso}(A, V)} + \frac{\text{cut}(A, B)}{\text{asso}(B, V)}$$

asso(A,V) $=$ sum of all weights from nodes in A

cut(A,B) $=$ sum of all weights between nodes in A and nodes in B

Spectral Clustering

Can be shown to be a relaxation of the Normalized Cut on the similarity graph

$$\text{Normalized Cut}(A,B) = \frac{\text{cut}(A,B)}{\text{asso}(A,V)} + \frac{\text{cut}(A,B)}{\text{asso}(B,V)}$$

asso(A,V) $= \sum_{u \in A, t \in V} w(u,t)$

cut(A,B) $= \sum_{u \in A, v \in B} w(u,v)$

Normalized Cut Method (after Michael Jordan's notes)

$$\text{Normalized Cut}(A,B) = \frac{\text{cut}(A,B)}{\text{asso}(A,V)} + \frac{\text{cut}(A,B)}{\text{asso}(B,V)}$$

asso(A,V) $= \sum_{u\in A, t\in V} w(u,t)$

cut(A,B) $= \sum_{u\in A, v\in B} w(u,v)$

e_A indicator variable for A ($e_A = 1$ if ith datapoint in A)

$$D(i,i) = \sum_j W(i,j)$$

Normalized Cut Method (after Michael Jordan's notes)

e_A indicator variable for A ($e_A = 1$ if ith datapoint in A)

$$D(i,i) = \sum_j W(i,j)$$

$$e'_A D e_A = \sum_{u \in A, t \in V} W_{ut}$$

$$e'_A W e_A = \sum_{u \in A, w \in A} W_{uw}$$

$$e_A (D - W) e_A = \sum_{u \in A, v \in B} W_{uv}$$

$$NC = \frac{e'_A (D - W) e_A}{e'_A D e_A} + \frac{e'_B (D - W) e_B}{e'_B D e_B}$$

Normalized Cut Method (after Michael Jordan's notes)

Computing Normalized Cut exactly is NP hard.

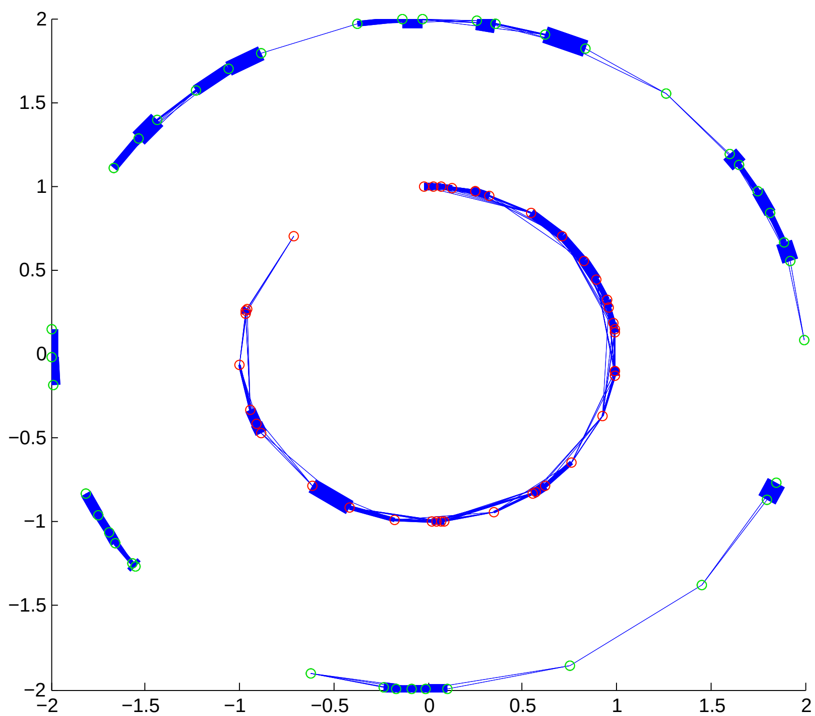
The spectral solution is to relax the constraint on 0,1 membership.

$$NC = \frac{e'_A(D-W)e_A}{e'_A D e_A} + \frac{e'_B(D-W)e_B}{e'_B D e_B}$$

Smallest generalized eigenvector of $(D-W)v = \lambda D v$ which is simply related to the largest eigenvector of $D^{-.5} W D^{-.5} = L$

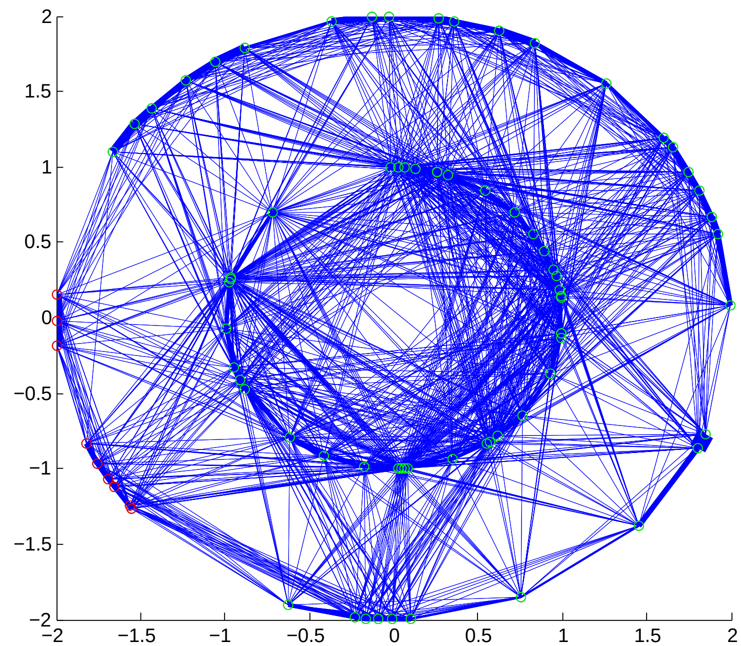
For small graphs, you can compute the normalized cut exactly. Michael Jordan says that "Surprisingly using the relaxed solution performs $better$ than using the exact solution. No one really knows why."

Picking σ appropriately is critical



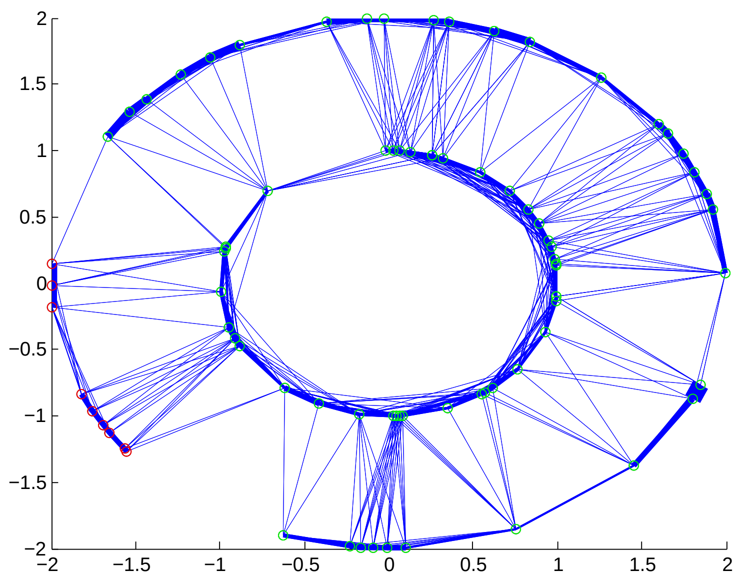
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

Picking σ appropriately is critical



2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

Spectral Clustering



2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2

Spectral Clustering

```
%% refs
%%[1] Shi, J., and J. Malik (1997) "Normalized Cuts and Image Segmenta
%% in Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition,
%% Puerto Rico.

%%[2] Kannan, R., S. Vempala, and A. Vetta (2000) "On Clusterings - Go
and Spectral", Tech. Report, CS Dept., Yale University.

%% This code is from ref [3]
%%[3] Ng, A. Y., M. I. Jordan, and Y. Weiss (2001) "On Spectral Cluste
%%  Analysis and an algorithm", in Advances in Neural Information Proc
Systems 14.


%%[4] Weiss, Y. (1999) "Segmentation using eigenvectors: a unifying vi
%% Tech. Rep., CS. Dept., UC Berkeley.
```

```
%% make sample data -- change as you wish
numsamps=20
cluster1=mvnrnd([0 0], [3 .4 ; .4 .1], numsamps)
cluster2=mvnrnd([0 2], [.5 0;0 .5],numsamps)
scatter(cluster2(:,1),cluster2(:,2),'r')
hold on
scatter(cluster1(:,1),cluster1(:,2),'b')

allpts=[cluster1;cluster2]
goto=length(allpts);
%%

%% compute A (step 1)
sigsq=1;
Aisq=allpts(:,1).^2+allpts(:,2).^2;
Dotprod=allpts*allpts'
distmat=-repmat(Aisq',goto,1)-repmat(Aisq,1,goto)+2*Dotprod;
Afast=exp(distmat/(2*sigsq));
A = Afast-diag(diag(Afast));
```

```
%
%  slow but safe way
%for i=1:goto
%  D(i,i)=0;
%  for j=1:goto
%    distmatlong(i,j)=(-norm(allpts(i,:)-allpts(j,:))^2);
%    A(i,j)=exp((-norm(allpts(i,:)-allpts(j,:))^2)/(2*sigsq));
%    A(i,i)=0;
%    D(i,i)=D(i,i)+A(i,j);
%  end
%end

%% step 2
D = diag(sum(A'))
L=D^(-.5)*A*D^(-.5)

%% step 3
[X,di]=eig(L)
[Xsort,Dsort]=eigsort(X,di)
Xuse=Xsort(:,1:2)
%% normalize X to get Y (step 4)
Xsq = Xuse.*Xuse;
```

```
divmat=repmat(sqrt(sum(Xsq')'),1,2)
Y=Xuse./divmat
%% step 5/6
addpath /home/lab
[c,Dsum,z] = kmeans(Y(:,2)',2)




%for comparison
[c2,Dsum2,z2] = kmeans(allpts',2)



%comparison
[X2,d2] = eig(A)
[X2sort,d2sort]=eigsort(X2,d2)
imagesc(X2sort(:,39:40))
kmeans(X2sort(:,39)',2)



[X2,d2] = eig(L)
[X2sort,d2sort]=eigsort(X2,d2)
imagesc(X2sort(:,1:2))
```

```
kmeans(X2sort(:,2)',2)
```

Multi-class Schemes

Some algorithms use multiple eigenvectors (k for a k-way partition) to divide in to k classes. Others perform repeated binary cuts.

In general results are not as nice with the multi-class algorithms.

Example Results (Ng,Jordan & Weiss 2001)

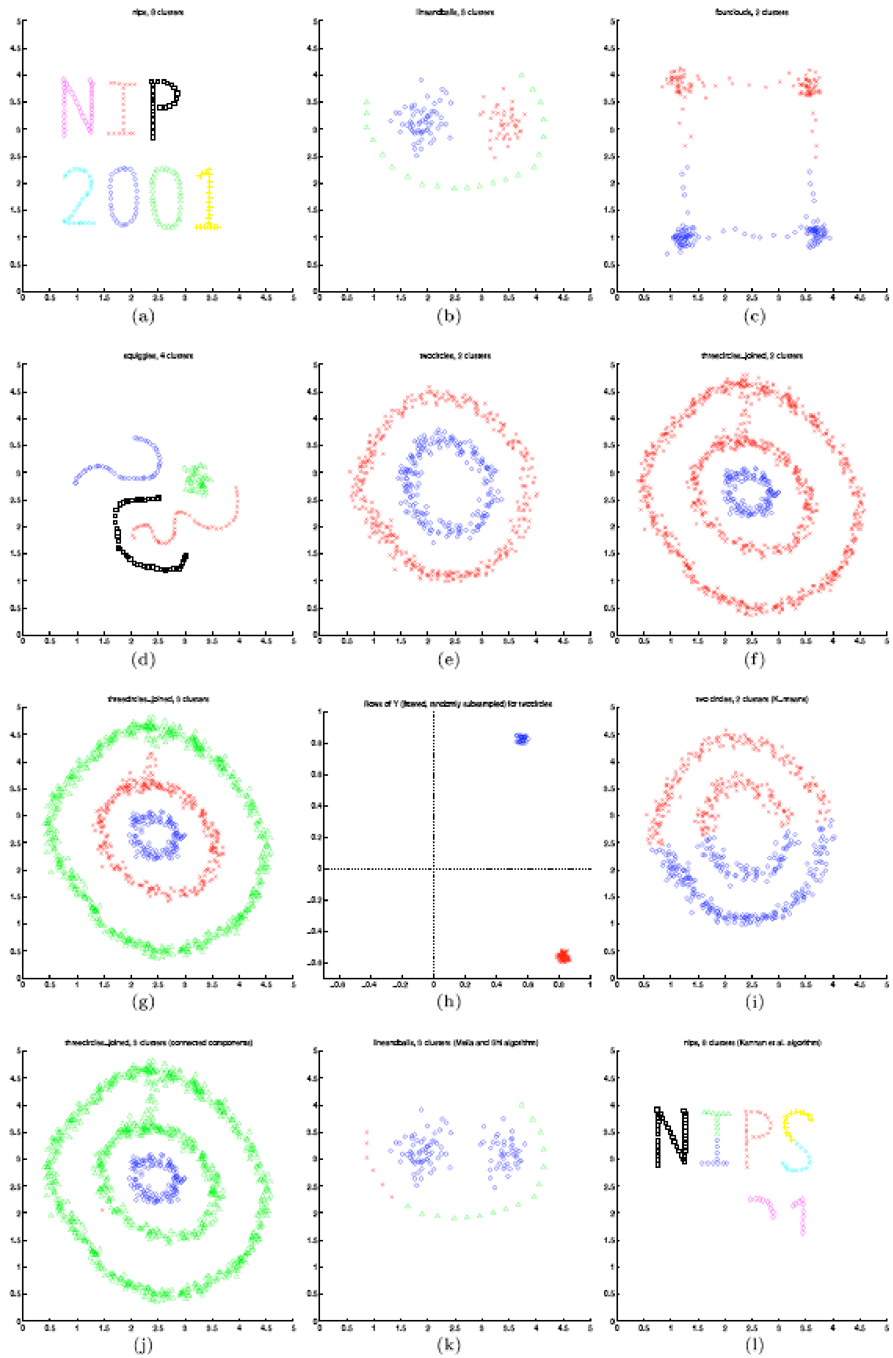


Figure 1: Clustering examples, with clusters indicated by different symbols (and colors, where available). (a-g) Results from our algorithm, where the only parameter varied across runs was k. (h) Rows of Y (jittered, subsampled) for `twocircles` dataset. (i) K-means. (j) A "connected components" algorithm. (k) Meila and Shi algorithm. (l) Kannan et al. Spectral Algorithm I. (See text.)

Image segmentation results (Fowlkes,Martin,Malik)

http://www.cs.berkeley.edu/ fowlkes/BSE/cvpr-segs/

(work by Fowlkes, Martin, Malik (UC Berkeley) – uses sophisticated learning of pairwise similarities)