Úvod do shlukování v matlabu

R. Kessl

CS CAS, 2. March 2009

Outline

K-means – Jak funguje ?

Mějme body $x_i \in R^n$ a číslo k určující počet shluků.

1. Vytvoř počáteční centroidy c_i.
2. Pro každý bod x_i spočítej *příslušnost* bodu ke clusteru c_i.
3. přepočítej polohu každého centroidu c_i.

$\Rightarrow$ probíhá iterativně: postupně se upravují centroidy.

K-means

1. v matlabu fce `[IDX, C]=kmeans(data,počet shluků)` (další parametry viz. help)
2. podstatné parametry: data, počet shluků, měřítko vzdálenosti.

Zkusme si udělat shluky na datech `fisheriris`.

```
load fisheriris;
[measIDs, measC]=kmeans(meas, 2);
gplotmatrix(meas, meas, measIDs);
silhouette(meas, measIDs);
```

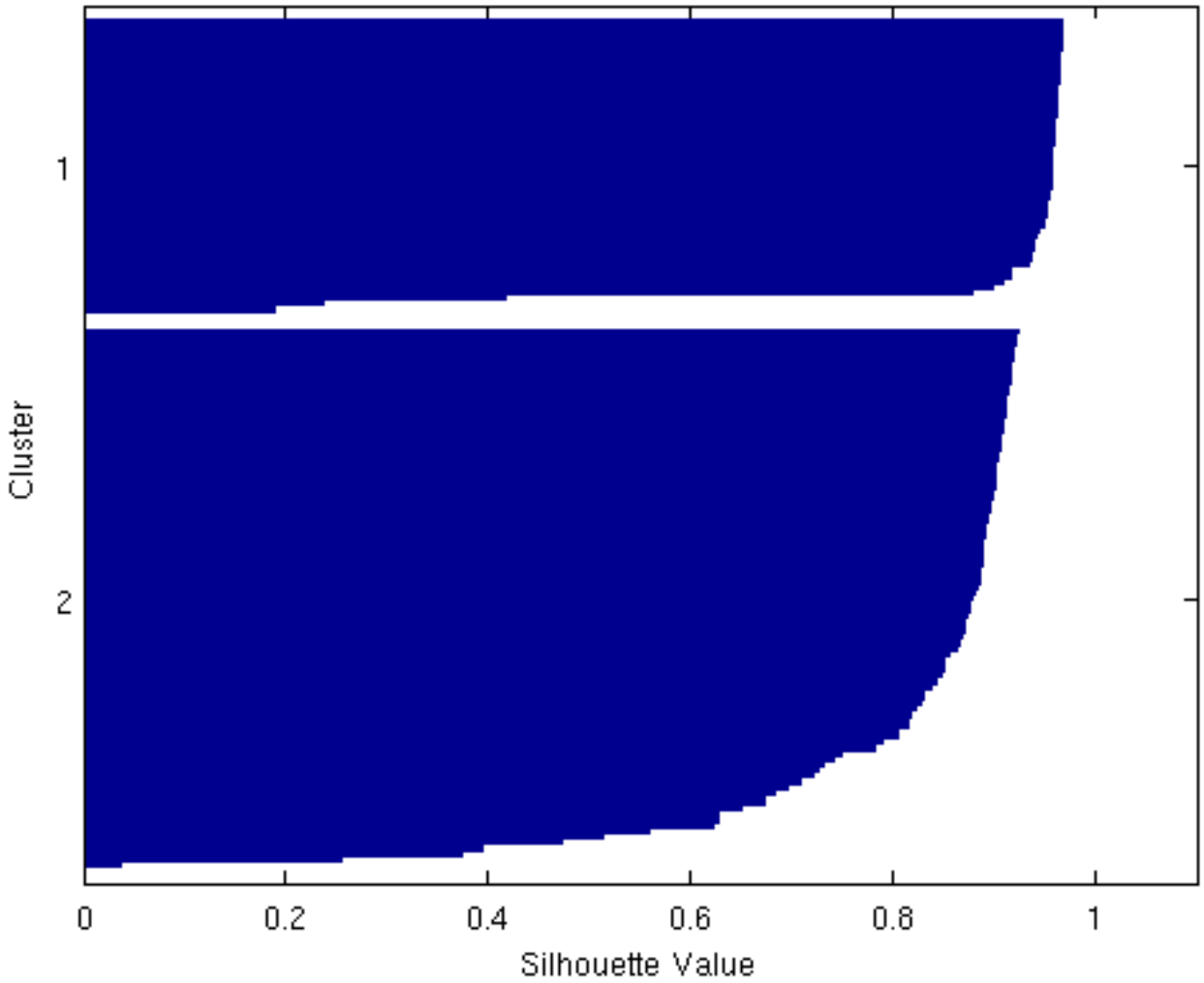
K-means - funkce silhouette

- pro bod x_i udává c_i číslo shluku do kterého x_i patří
- z matematického pohledu je funkce *Silhouette(x)* měřítko příslušnosti bodu x do clusteru
- počítá se z příslušnosti x_i do **všech** clusterů

K-means - funkce silhouette

- funkce `silhouette` v *matlabu* ukáže jak moc patří jednotlivé body do shluků.
- `silhouette(data, shluky)`
- zobrazuje se každý bod v datech a jeho *silhouette* hodnota
- čím větší hodnota tím víc patří bod ke shluku.
- body jsou seřazeny podle shluků a potom podle hodnoty funkce *silhouette*
- zkuste změnit vykreslit silhouette pro různý počet shluků.

K-means - funcke silhouette



K-means - zobrazení dat ve 3D

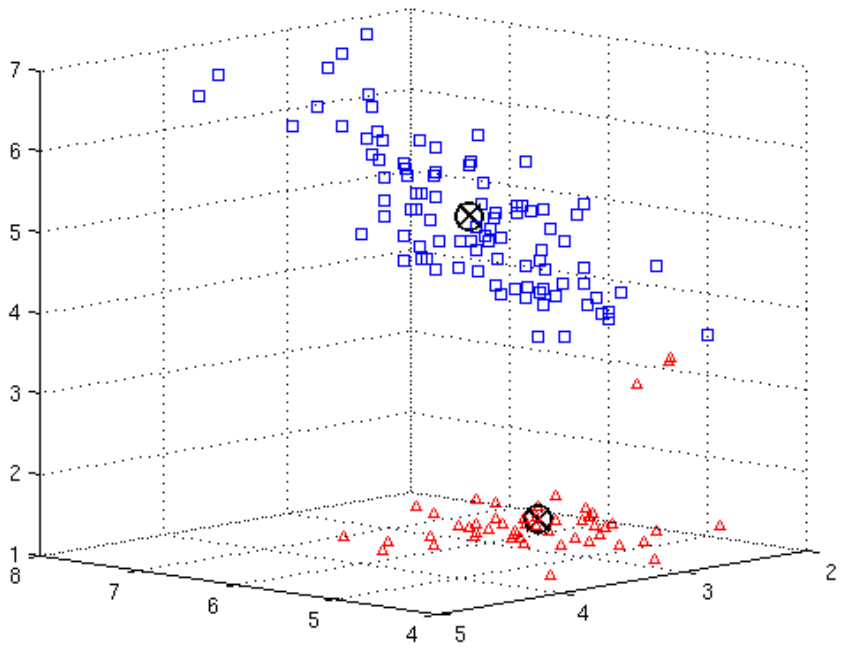
- zkusme vizualizovat data, rozdělená na dva shluky

```
ptsymb = {'bs','r^','md','go','c+'};
for i = 1:2
    clust = find(measIDs==i);
    plot3(meas(clust,1),meas(clust,2),meas(clust,3),ptsymb{i});
    hold on
end

plot3(measC(:,1),measC(:,2),measC(:,3),'ko', 'MarkerSize', 14, '↩
    LineWidth',2);
plot3(measC(:,1),measC(:,2),measC(:,3),'kx', 'MarkerSize', 14, '↩
    LineWidth',2);

view(-137,10);
grid on
```

K-means - zobrazení dat ve 3D

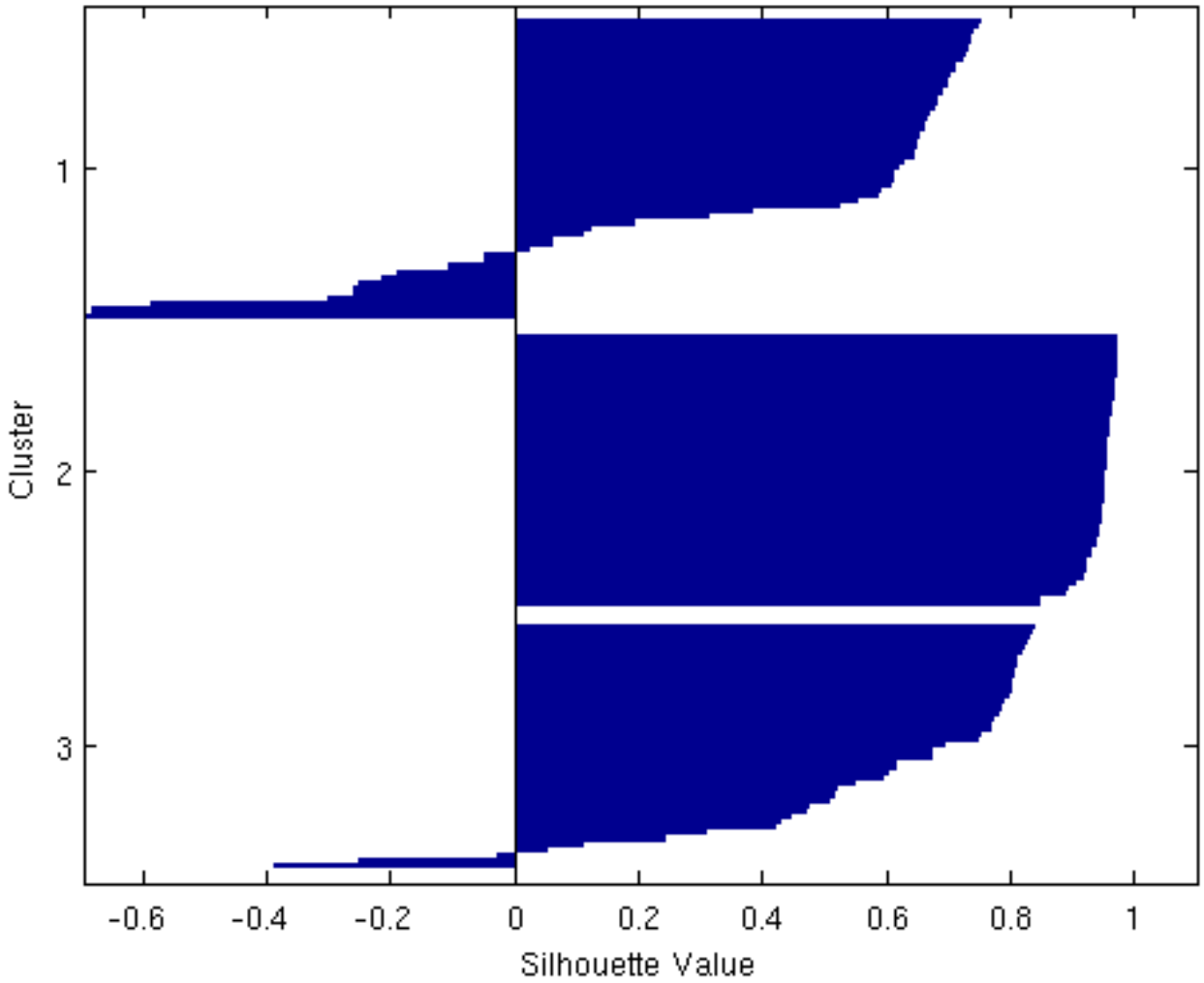


K-means - další parametry shlukování

- parametr 'display' specifikuje co se má během výpočtu vypisovat, zkuste zadat hodnotu iter
- Volba počtu iterací, tedy kolikráte se spustí kmeans s různými počátečními nastaveními centroidů: 'replicates', 5, 'display', 'final'. Ze všech pokusů se vybere ten nejlepší. (zkuste si)
- Volba počátečních centroidů: 'start' (zkuste si doma)
- změna měřítka vzdálenosti: 'dist', zkuste třeba 'cos'

```
[measIDs, measC]=kmeans(meas, 2, 'display', 'final', '↩
    replicates', 5);
[measIDs, measC]=kmeans(meas, 3, 'display', 'final', '↩
    replicates', 5);
[measIDs, measC]=kmeans(meas, 3, 'display', 'final', '↩
    replicates', 5, 'dist', 'cos');
silhouette(meas, measIDs);
```

K-means - další parametry shlukování



K-means – srovnání hodnot silhouette

```
[measIDs,measC] = kmeans(meas,2,'dist','sqeuclidean');
[silh1] = silhouette(meas,measIDs,'sqeuclidean');


[measIDs, measC]=kmeans(meas, 3, 'replicates', 5, 'dist','↩
    sqeuclidean');
[silh2] = silhouette(meas,measIDs,'sqeuclidean');

[measIDs, measC]=kmeans(meas, 3, 'dist','sqeuclidean');
[silh3] = silhouette(meas,measIDs,'sqeuclidean');


[measIDs, measC]=kmeans(meas, 3, 'dist', 'cos');
[silh4]=silhouette(meas, measIDs, 'cos');


[mean(silh1) mean(silh2) mean(silh3) mean(silh4)]
```

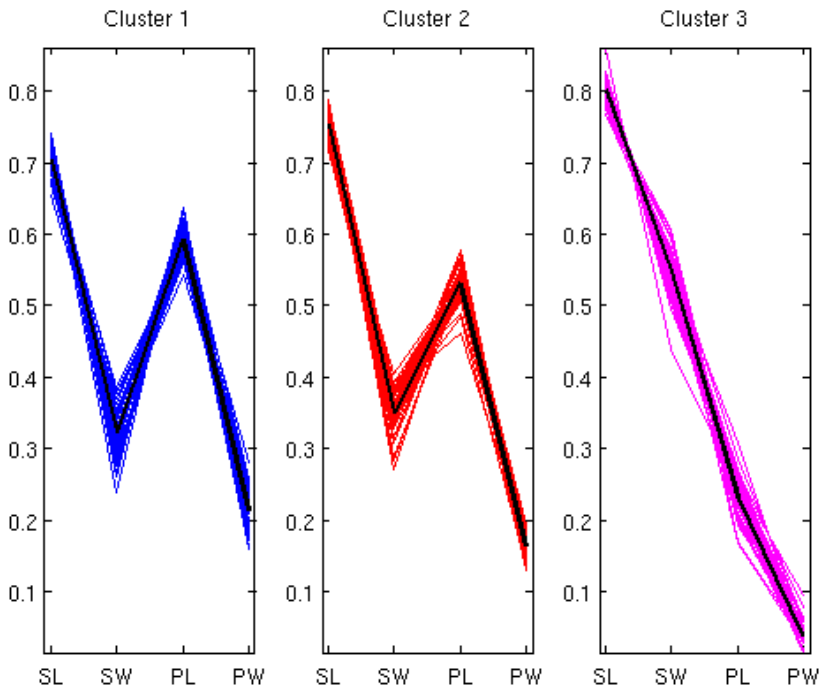
Jaký je rozdíl ve vytvořených shluknutí ?

silh1=0.8504 silh2=0.7357 silh3=0.7357 silh4=0.7491

Rozdíl mezi velikostí listů

```
[measIDs, measC]=kmeans(meas, 3, 'dist', 'cos');
lnsymb = {'b-','r-','m-'};
names = {'SL','SW','PL','PW'};
meas0 = meas ./ repmat(sqrt(sum(meas.^2,2)),1,4);
ymin = min(min(meas0));
ymax = max(max(meas0));
for i = 1:3
    subplot(1,3,i); plot(meas0(measIDs==i,:)',lnsymb{i});
    hold on; plot(measC(i,:)','k-','LineWidth',2); hold off;
    title(sprintf('Cluster %d',i));
    set(gca,'Xlim',[.9 4.1],'XTick',1:4,'XTickLabel',names,'YLim'↩
        ,[ymin ymax])
end
```

Rozdíl mezi velikostí listů



Hierarchické shlukování

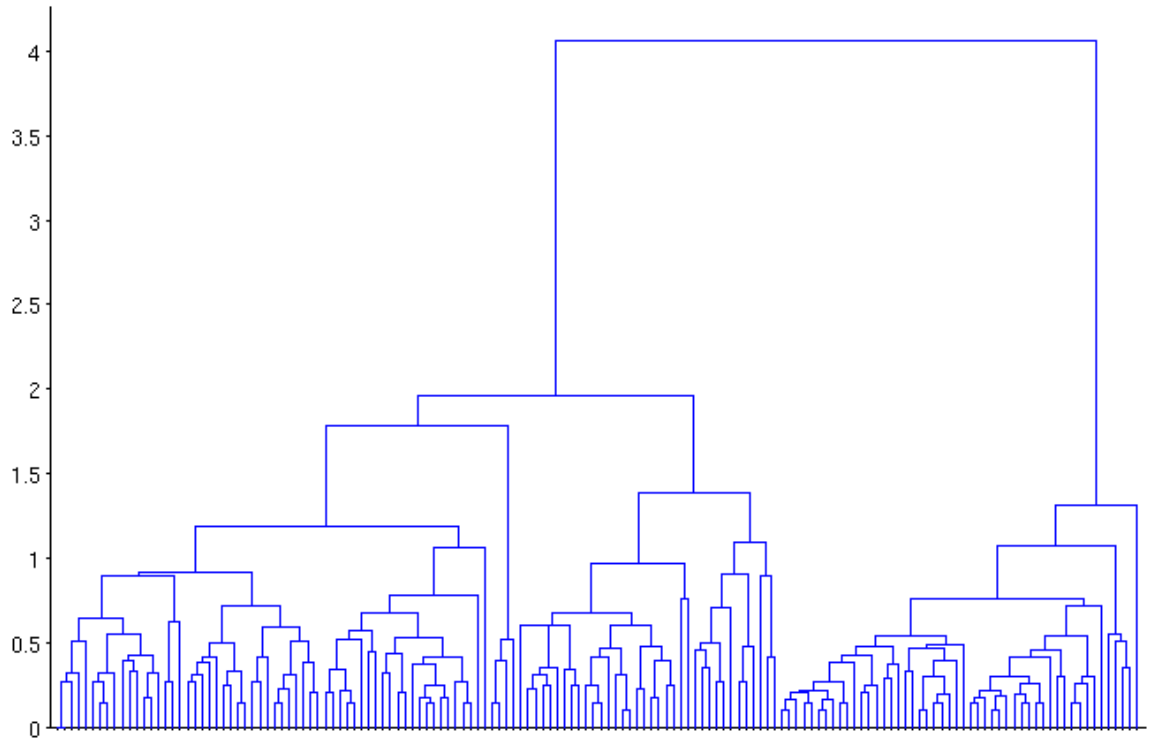
Jak probíhá shlukování ?

1. Pro hierarchické shlukování potřebuji znát jejich *podobnost/vzdálenost*.
2. Pro spočítání matice vzdáleností (každý-s-každým) $\Rightarrow$ funkce `pdist`. $O(n^2)$.
3. Shluknutí objektů do binárního stromu $\Rightarrow$ funkce `linkage`
4. cophenetická korelace: určuje jak moc je dendrogram shodný se skutečnými vzdálenostmi. Čím větší číslo, tím lépe.
5. Visualizace: dendrogram $\Rightarrow$ funkce `dendrogram`

```
measPdist = pdist(meas,'euclidean');
measTree = linkage(measPdist,'average');
[h,nodes] = dendrogram(measTree,0);
set(gca,'TickDir','out','TickLength',[.002 0],'XTickLabel',[]);

cophenet(measTree,measPdist)
```

Hierarchické shlukování



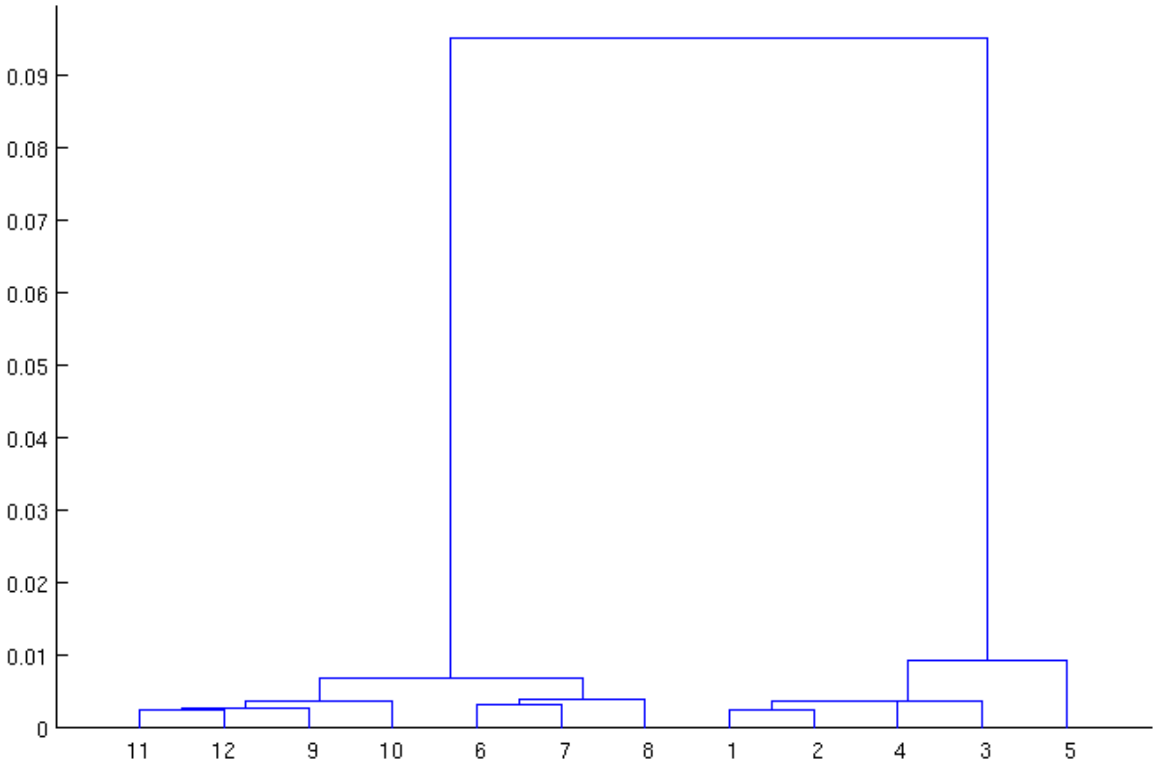
Hierarchické shlukování – pomocí cos vzdálenosti

```
measPdistCos = pdist(meas,'cosine');
measTreeCos = linkage(measPdistCos,'average');
cophenet(measTreeCos,measPdistCos)


[h,nodes] = dendrogram(measTreeCos,0);
set(gca,'TickDir','out','TickLength',[.002 0],'XTickLabel',[]);


[h,nodes] = dendrogram(clustTreeCos,12);
[sum(ismember(nodes,[11 12 9 10])) sum(ismember(nodes,[6 7 8])) ↩
    sum(ismember(nodes,[1 2 4 3])) sum(nodes==5)]
```

Hierarchické shlukování – pomocí cos vzdálenosti



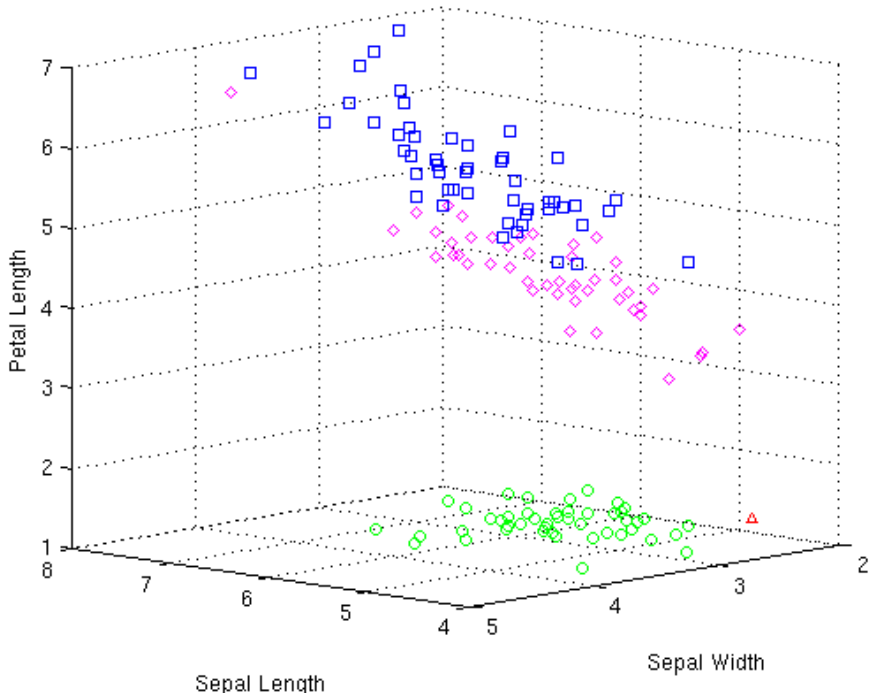
Cophenetický koeficient korelace: 0.9360

Hierarchické shlukování – zobrazení shluků

- modrý a růžový shluk leží blízko u sebe
- zelený shluk je dobře separovný od ostatních shluků

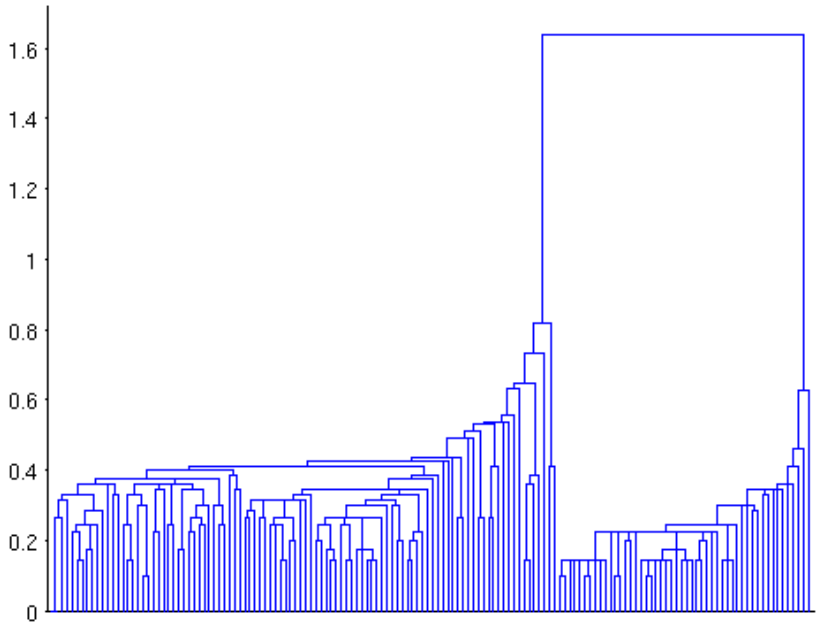
```
hidx = cluster(measTreeCos,'criterion','distance','cutoff',.006);
for i = 1:5
    clust = find(hidx==i);
    plot3(meas(clust,1),meas(clust,2),meas(clust,3),ptsymb{i});
    hold on
end
hold off
xlabel('Sepal Length'); ylabel('Sepal Width'); zlabel('Petal ↩
    Length');
view(-137,10);
grid on
```

Hierarchické shlukování – zobrazení shluků



Hierarchické shlukování – single linkage

```
eucD = pdist(meas,'euclidean');
clustTreeSng = linkage(eucD,'single');
[h,nodes] = dendrogram(clustTreeSng,0);
set(gca,'TickDir','out','TickLength',[.002 0],'XTickLabel',[]);
```



Cophenetický koeficient korelace: 0.8639