

# Psychology 594

## Final Take Home Exercise

November 19, 2010

All of the areas below are to be done with your chosen proximity matrix (from Michael Lee's web site).

[http://cda.psych.uiuc.edu/multivariate\\_class\\_final\\_2010.zip](http://cda.psych.uiuc.edu/multivariate_class_final_2010.zip)

I. (a) From routines available in the Matlab Statistical Toolbox, carry out a complete-link hierarchical clustering. Use at least the following three M-files:

`squareform.m`  
`linkage.m`  
`dendrogram.m`

(b) From the Cluster Analysis Toolbox M-files:

(i) use `order.m`; `ultrafit.m` (with a complete-link target); `ultrafnd.m` (with `randperm` several times).

(ii) use `ultrafnd_confit.m` (with the order found in (i)), and `ultrafnd_confnd.m`.

(iii) use `partitionfnd_averages.m`; `partitionfnd_diameters.m`; and then `partitionfit.m` after each.

(iv) use `cent_ultrafnd_confit.m` (with the order found in (i)), and `cent_ultrafnd_confnd.m`.

(v) use `atreefit.m` (with a complete-link target); `atreefnd.m` (with `randperm` several times); `atreedec.m`, and `ultraorder.m`.

(vi) use `consec_subsetfit.m` and `consec_subsetfit_alter.m`.

II. Carry out multiple restart Monte Carlo on your data using an M-script parallel to `ms_script_yourdata_mds.m`.

```
load decathlon.dat
```

```
decathlon_dissimilarities = 1 - decathlon;
```

```
decathlon_dissimilarities
```

```
n = 10;
```

```
tic;
```

```
opts = statset('Maxiter',1000);
```

```
best_vaf = 0.0;
```

```
store_vaf = zeros(100,1);
```

```
for k = 1:100
```

```
    [coords, stress] = ...  
    mdscale(decathlon_dissimilarities, 2, 'Criterion', ...  
    'metricsstress', 'Start', 'random', ...  
    'Replicates', 1, 'Options', opts);
```

```
n = size(coords,1);
```

```

distance_matrix = zeros(n,n);

for i = 1:n
    for j = 1:n

        distance_matrix(i,j) = ...
            sqrt(((coords(i,1) - coords(j,1))^2) + ...
                ((coords(i,2) - coords(j,2))^2));
    end
end

decathlon_vec = squareform(decathlon_dissimilarities);

distance_vec = squareform(distance_matrix);

r = corrcoef(decathlon_vec', distance_vec');

vaf = r(1,2)^2;

store_vaf(k) = vaf;

if(vaf > best_vaf)

    best_vaf = vaf;
    best_coords = coords;
    best_distance_vec = distance_vec;

end
end

```

```

sorted_vafs = sort(store_vaf');

sorted_vafs
best_vaf
best_coords
best_distance_vec

figure(1)

axis equal

plot(best_coords(:,1),best_coords(:,2),'ko')

hold on

for i = 1:n

    objectlabels{i,1} = int2str(i);

end

text(best_coords(:,1),best_coords(:,2),objectlabels,...
'fontsize',10,'verticalalignment','bottom')

toc;

euclidean_coordinates = [best_coords(:,1),best_coords(:,2)];

figure(2)

axis equal

```

```

plot(decathlon_vec,best_distance_vec,'bo')

hold on

xlabel('Dissimilarities')
ylabel('Distances')

tic;

best_vaf = 0.0;

store_vaf = zeros(100,1);

best_disparities = zeros(n,n);

for k = 1:100

[coords,stress,disparities] = ...
mdscale(decathlon_dissimilarities,2,'Criterion',...
'sstress','Start',...
'random','Replicates',1,'Options',opts);

n = size(coords,1);

distance_matrix = zeros(n,n);

for i = 1:n
    for j = 1:n

        distance_matrix(i,j) = ...
            sqrt(((coords(i,1) - coords(j,1))^2) + ...

```

```

                ((coords(i,2) - coords(j,2))^2));
            end
        end

        decathlon_vec = squareform(decathlon_dissimilarities);

        distance_vec = squareform(distance_matrix);

        r = corrcoef(decathlon_vec', distance_vec');

        vaf = r(1,2)^2;

        store_vaf(k) = vaf;

        if(vaf > best_vaf)

            best_vaf = vaf;
            best_coords = coords;
            best_disparities = disparities;
            best_distance_vec = distance_vec;

        end end

        store_vaf;

        sorted_vafs = sort(store_vaf');

        sorted_vafs
        best_vaf
        best_coords

```

```

figure(3)

axis equal

plot(best_coords(:,1),best_coords(:,2),'ko')

hold on

for i = 1:n

    objectlabels{i,1} = int2str(i);

end

text(best_coords(:,1),best_coords(:,2),objectlabels,...
'fontsize',10,'verticalalignment','bottom')

toc;

euclidean_coordinates_nonmetric = ...
[best_coords(:,1),best_coords(:,2)];

best_disparities_vec = squareform(best_disparities);

best_distance_vec

best_disparities_vec

figure(4)

axis equal

```

```

[dum,ord] = sortrows([best_disparities_vec(:) decathlon_vec(:)]);

plot(decathlon_vec,best_distance_vec,'bo',...
decathlon_vec(ord),best_disparities_vec(ord),'r.-')

hold on

xlabel('Dissimilarities')
ylabel('Distance/Disparities')

legend({'Distances' 'Disparities'}, 'Location', 'NW')

[d,z,transform] = ...
procrustes(euclidean_coordinates_nonmetric,euclidean_coordinates);

figure(5)

axis equal

plot(euclidean_coordinates_nonmetric(:,1),...
euclidean_coordinates_nonmetric(:,2),'rx',...
euclidean_coordinates(:,1),...
euclidean_coordinates(:,2),'b.',...
z(:,1),z(:,2),'ko')

hold on

text(euclidean_coordinates_nonmetric(:,1),...
euclidean_coordinates_nonmetric(:,2),objectlabels,...

```



```
'fontsize',8,'verticalalignment','bottom')
```

```
text(z(:,1),z(:,2),objectlabels,'fontsize',8,...  
'verticalalignment','bottom')
```

```
transform(1).b
```

```
transform(1).T
```

```
transform(1).c
```