Aprendizaje Automático - Trabajo Práctico 4

Gonzalo Castiglione - 49138

June 1, 2012

Objetivo: Aprender a agrupar datos mediante análisis de clusters

1 Determinación de clases. Clustering.

1. El comando *princomp* de MATLAB c calcula las componentes principales para una matriz de datos. También se pueden calcular los valores y vectores propios de la matriz de correlación usando los comandos *corrcoef* y *eigs* porque la componente principal es el autovector asociado al autovalor dominante de dicha matriz.

   Una consideracion a tener antes del cálculo de las componentes principales es que las unidades de medida de las variables X_i son muy distintas y no hay que olvidar que los cambios de unidades afectan a la varianza de la variable en el sentido que:

   $$\xi = aX_i \Longrightarrow var(\xi) = a^2 var(X_i)$$

   y como consecuencia, esto afecta a las componentes principales.

   Las elevadas varianzas de las mediciones de "Número de teléfonos fijos registrados" y el "Número de vehículos de motor matriculados" hacen prever que un análisis de componentes principales realizado a partor de la matroz de covariansa S dará como resultado una primera y segunda componentes principales que coincidan basicamente con estas dos variables observadas.

   Para poder obtener uasn componentes principales que no dependan de las unidades en que han sido medidas las variables originales, se debe estandarizar a media vero y varianza unidad las variables originales. Lo que equivale a realizar el análisis de componentes a partir de la matriz de correlaciones.

   (a) [r,p] = corrcoef(x) % compute sample correlation and p-values

   (b) [i,j] = find(p<0.05); % find significant correlations

2. Lirios de Fisher

   (a) Clasificación del las mediciónes según el algoritmo *kmeans*

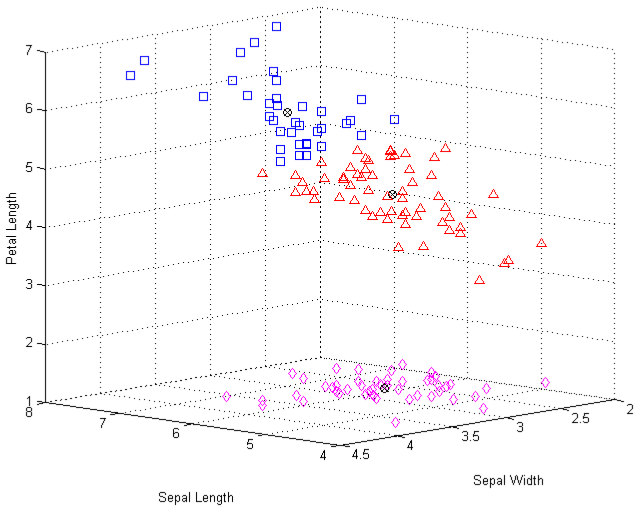

Figure 1: Agrupamiento de los lirios de Fisher

Matriz de confusión

	Clase Predicha			
Clase Real		Setosa	Versicolor	Virginica
	Setosa	50	0	0
	Versicolor	0	42	8
	Virginica	0	14	36

$$error = \frac{8+14}{150} \backsimeq 0.15$$

(b) Calsificación segun algoritmo de clusters jerárquicos aglomerativos

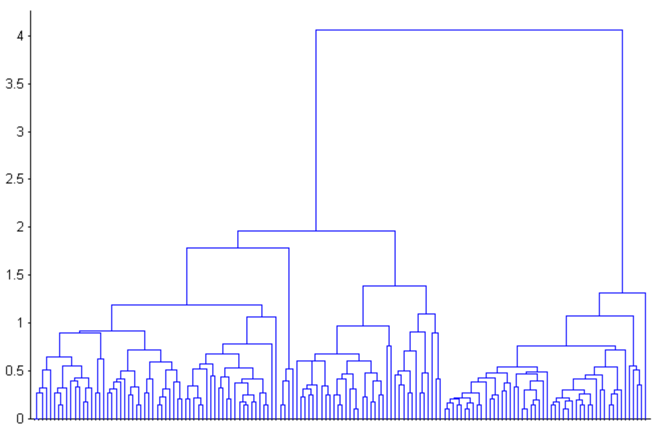

Figure 2: Agrupamiento de los lirios de Fisher

Matriz de confusión

	Clase Predicha			
Clase Real		Setosa	Versicolor	Virginica
	Setosa	50	0	0
	Versicolor	0	49	1
	Virginica	0	15	35

$$error = \frac{1+15}{150} \backsimeq 0.11$$

(c) Luego de comprados los resultados obtenidos por cada algoritmo en los puntos (a) y (b) se pudo concluir que para este problema, el algoritmo que mejor clasifica a los los lirios recogidos es el de clasificación jerarquica ya que obtuvo un error menor al momento de realizar las clasificaciones.

Matriz obtenida en el ejercicio 5 del tp 2:

	Clase Predicha			
Clase Real		Setosa	Versicolor	Virginica
	Setosa	50	0	0
	Versicolor	0	47	3
	Virginica	0	3	47

$$error = \frac{3+3}{150} \backsimeq 0.04$$

Como se pudo obervar, este algoritmo clasificó aun mejor que los anteriores a los lirios, en especial para los de la clase $Virginica$, que es donde mayor cantidad de errores tuvieron los dos anteriores.

3. (TO DO)

4. Congresistas

   (a) Luego de realizadas pruebas de clasificación variando el tipo de distancia entre *euclidean* y *cityblock*. Y el tipo de metodo de creación del árbol jerárquico entre *single*, *complete* y *average*. Se pudo observar que todos pudieron calsificar correctamente a todos los congrsistas de acuerdo a si es Demócrata o Republicando (excepto por uno que siempre es clasificado mal) a excepción del método *single*, el cual terminó por clasificar a todos como Democratas excepto por el número 2, que fue clasificado correctamente con Republicano.

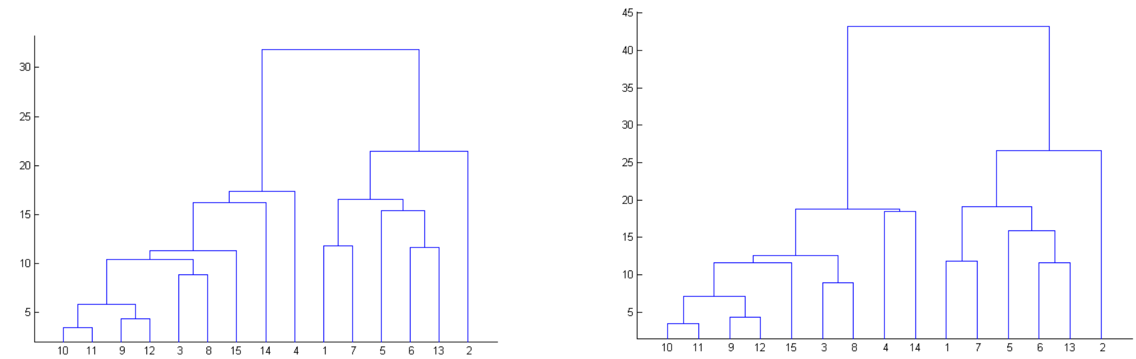

Figure 3: Árbol de jerarqua creados utilizando al distancia *euclidea* con el método *average*(izquierda) y el método *complete*(derecha)

Para los métodos aplicados en los gráficos anteriores, se puede observar que si bien se obtuvieron distintas categorías para varios conjuntos de congresistas, tomando clusters de 2 (por los grupos democrata o republicano) ambos algoritmos clasificaron correctamente a todos los congresistas a exepción del numero 12 que en realidad pertenece al partido republicano.

A continuación se muestra la matriz de confusión para ambos árboles jerarquicos (notar que ambos presentan la misma matriz):

		Clase Predicha	
		Demócrata	Republicano
Clase Real	Demócrata	8	1
	Republicano	0	6

$$error = \frac{1}{15} \backsimeq 0.07$$

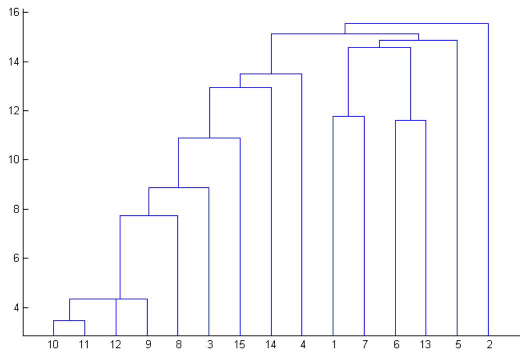

Figure 4: Árbol de jerarqua creado utilizando al distancia *euclidea* con el método *single*

Como se puede ver en la figura 4, no se obtuvieron buenos resultados al tratar de utilizar el metodo *single* (minimas distancias) para agrupar los datos ya que fueron clasificados todos dentro de la misma categoria, a excepcion del numero 2.

2 Código

1. (TO DO)
2. Lirios de Fisher

```
// a)
ptsymb = {'bs','r^','md','go','c+'};
load fisheriris;
k = 3;
[cidx,cmeans,sumd] = kmeans(meas,k);
for i = 1:k
    clust = find(cidx==i);
    plot3(meas(clust,1), meas(clust,2), meas(clust,3), ptsymb{i});
    hold on
end
plot3(cmeans(:,1), cmeans(:,2), cmeans(:,3),'ko');
plot3(cmeans(:,1), cmeans(:,2), cmeans(:,3),'kx');
hold off
xlabel('Sepal Length');
ylabel('Sepal Width');
zlabel('Petal Length');
view(-137,10);
grid on
types = [ones(1,50)*2 ones(1,50) ones(1,50)*3];
cMat = confusionmat(types,cidx)
// b)
dist = pdist(meas, 'euclidean');
clustTree = linkage(dist, 'average');
[h, nodes] = dendrogram(clustTree);
set(gca, 'TickDir', 'out', 'TickLength', [.002 0], 'XTickLabel', []);
cluster_labels = cluster(clustTree, 'maxclust', 3 );
confusion_matrix = crosstab( cluster_labels, class )
```

3. (TO DO)
4. Congresistas

```
Y = pdist(X, 'cityblock');
Z = linkage(Y, 'average');
H = dendrogram(Z);
```