Árboles Aleatorios en Frijoles Secos

Alberto Rodríguez

Universidad de Santiago de Chile. https://www.usach.cl/

Abstract. La base de datos "Dry Beans" corresponde a una serie de mediciones obtenidas a distintos frijoles no certificadas como de una única variedad. Se desarrolla un sistema de visión por computadora para distinguir siete variedades diferentes de frijoles secos. Se plantea utilizar la técnica de árboles aleatorios a través del paquete *randomForest* del software R. Los resultados se comparan con lo encontrado en la literatura, comprobando la importancia de la mayoría de las variables para poder identificar a que clase pertenece. Esta investigación concluye con la obtención de la mejor cantidad de árboles con la cantidad de nodos que mejor clasifique las clases conforme al OOB (*Out-of-Bag*) generado.

Keywords: Dry Beans · randomForest · OOB

1 Introducción

El frijol seco (*Phaseolus vulgaris L.*) es la legumbre más importante y producida en el mundo, por lo que desempeña un papel importante en la agricultura de Turquía [1]. La determinación de la mejor semilla es el principal problema para los productores de esta, por lo que es necesario determinar las características físicas como tamaño, color y diversidad. La clasificación manual de estas semillas de frijol es un proceso difícil y que consume mucho tiempo, por lo que se requiere métodos automáticos para clasificar y calificar.

Este artículo tiene la intención de usar el algoritmo de clasificación de bosques aleatorios, a través del uso de la función *randomForest*, el cual permite inferir más información al respecto de los datos contenido dentro del dataset "Dry Beans". Todo esto bajo el objetivo de predecir a cual de las 7 clases pertenece cada frijol, según las variables estudiadas dentro del dataset. Para realizar un contraste y aumentar la precisión del estudio, se elabora un pre-procesamiento de los datos con correlaciones entre los atributos para descartar variables innecesarias y en un caso un equilibrio en cuanto a la cantidad muestral clasificadora de cada clase. Se finaliza esta investigación con una discusión en base a los distintos modelos generados, realizando cambios en sus distintos parámetros como en el número de nodos y cantidad de árboles a generar, basándose en el método de clasificación de árboles aleatorios, conforme al menor error OOB generado.

2 Datos de la investigación

Esta conjunto de datos es obtenido por la Facultad de Tecnología de la Universidad de Selcuk. El objetivo del conjunto de datos es predecir según las carac-

terísticas de cada frijol seco si este pertenece a una de las siete tipos de frijoles secos. Se tomaron imágenes de 13.611 granos de las siete variedades, los conjuntos de datos describen características físicas de los frijoles y se tiene una variable resultado, la cual es una variable categórica la cual indica que tipo de frijol se tiene. Estas clases son:

- **Cali:** Es de color blanco, sus semillas son un poco gruesas y un poco más grandes que los frijoles secos y en forma de riñón.
- **Horoz:** Los frijoles secos de este tipo son largos, cilíndricos, de color blanco y generalmente de tamaño mediano.
- **Dermason:** Este tipo de frijoles secos, que son más planos, son de color blanco y un extremo es redondo y los otros extremos son redondos.
- **Seker:** Semillas grandes, de color blanco, la forma física es redonda.
- **Bombay:** Es de color blanco, sus semillas son muy grandes y su estructura física es ovalada y abultada.
- **Barbunya:** Fondo de color beige con rayas rojas o abigarrado, color moteado, sus semillas son grandes, la forma física es ovalada cerca de la redonda.
- **Sira:** Sus semillas son pequeñas, de color blanco, la estructura física es plana.

De los 13.611 frijoles, hay 1.322 (9,7% del total) que pertenecen a la clase Barbunya, 522 (3,8%) a Bombay, 1630 (11,9%) a Cali, 3546 (26%) a Dermason, 1928 (14%) a Horoz, 2027 (14,99%) a Seker y 2636 (19,3%) a Sira.

A continuación se enumeran y se describen los atributos del dataset, los cuales son todos numéricos. Se obtuvieron 12 características dimensionales y 4 de forma (todas medidas en milímetros).

- **Características dimensionales:**
  - **Área:** Zona de frijoles y el número de píxeles dentro de sus límites.
  - **Perímetro:** La circunferencia del frijol o longitud del borde.
  - **Longitud del eje mayor (L):** Distancia entre los extremos de la línea más larga.
  - **Longitud del eje menor (l):** La línea más larga que se puede trazar desde el frijol estando perpendicular al eje principal.
  - **Relación de aspecto:** Defina relación entre L y l.
  - **Excentricidad:** Excentricidad de la elipse que tiene los mismos momentos de la región.
  - **Área convexa:** N° de píxeles en el polígono convexo más pequeño que puede contener el área de la semilla de un frijol.
  - **Diámetro equivalente:** El diámetro de un círculo que tiene la misma área que el área de una semilla de frijol.
  - **Extensión:** La relación entre los píxeles del cuadro delimitador y el área del frijol.
  - **Solidez, Redondez, Compacidad:** Otras medidas del frijol
- **Características de forma:**
  - Shape Factor 1
  - Shape Factor 2
  - Shape Factor 3
  - Shape Factor 4

3 Pre-procesamiento

Tomando en cuenta los valores entregados por el dataset y con el objetivo de mejorar la precisión de la experiencia, se ha realizado un estudio de las variables. Primero se elimina la variable clase para no tener problemas posteriores en el agrupamiento (Este atributo lo utilizaremos después).

Segundo se verifica si el dataset se encuentra completo, es decir, que no hayas datos "perdidos" o algunas variables hayan sido rellenadas con un 0 o un "null". Al usar la función de R *apply*, se pudo verificar que a ninguna variable le faltan datos. Para complementar lo anterior también se busco que no existiera valores extremos (outliers) para esto se gráfico los datos de cada variable (ver **Figura 2**) y ver si existían outliers. Se puedo verificar que no tiene valores extremos.

3.1 Eliminación de variables

Eliminación del Área y Perímetro: A pesar de que la mayoría de los atributos son el resultados fórmulas que contienen como entrada otros atributos, es decir, atributos son dependientes de otros, estas son variables aleatorias y es importante ver si existe algún tipo de relación entre estas. Para esto evaluarlos utilizaremos una matriz de correlación.

Como se aprecia en la **Figura 3** existen muchas variables con varias relaciones entre ellas (Los colores azul y rojo mas oscuro indican que existe mayor relación), el coef. de correlación utilizado es Pearson ya que se trata solo de variables numéricas. Se puede observar que existe una relación muy grande entre el área y el perímetro, además estas tiene una relación muy grande con otras variables, así que se decide eliminar ambas

Eliminación de 4 variables más: Al realizar nuevamente una matriz de correlación (Ver **Figura 4)** Se puede apreciar que las variables "Compactness" y "Shape Factor3" tiene una fuerte relación cal igual que ConvexArea" y "Equiv-Diameter", por lo que se realiza un test de hipótesis para ambos, en donde:

- **H0:** No hay relación entre las variables
- **H1:** Hay relación entre las variables

Para realizar el test, se realiza un estudio del coef. de correlación de Pearson con 95% de confianza, el cual se resume en la Tabla 1:

Table 1. Test de correlación entre Compacidad y Shape Factor 3.

Atributos	**Correlación**	**P-valor**	**Intervalo de confianza**
Compacidad / ShapeFactor3	0.9986	$< 2.2e\text{-}16$	[0.99864 0.99872]
ConvexArea / EquivDiameter	0.9852	$< 2.2e\text{-}16$	[0.98472 0.98571]

Con los resultados se cumple H1 y las cuatro variables esta relacionadas por lo que se eliminan.

Eliminación de Longuitud del eje menor y ShapeFactor1: Se realiza nuevamente una matriz de correlación (Ver **Figura 5**) y se encuentra relación entre los atributos "MinorAxisLength" y "ShapeFactor1" y se realiza el mismo test de hipótesis anterior obteniendo resultados similares, donde se cumple H1 y se eliminan ambas variables.

4 Método Utilizado

El método utilizado para realizar la clasificación de los datos existentes dentro del dataset corresponde a la clasificación por bosques aleatorios. Este modelo utiliza una parte del conjunto de datos para crear una gran cantidad de árboles de clasificación, los cuales son utilizados en conjunto para generar un clasificador robusto. Además, para probar el clasificador se utilizan los datos que no fueron seleccionados en un principio, el cual se denomina Out of Bag (OOB). Para conseguir una mejor clasificación el algoritmo modifica los árboles generados, en base a la importancia de las variables, mientras compara el error producido por cada una de las combinaciones generadas.

5 Resultados de la investigación

Se utiliza el método de bosques aleatorios, variando el dataset generado, la cantidad de árboles a generar y la cantidad máxima de nodos a utilizar como parámetros de entrada de la función r*randomForest*. El número de árboles utilizados son de 400, 500, 600, 700 y 800; mientras el número de nodos depende de la cantidad de variables de dataset utilizado. A continuación se en listan los distintos dataset generados.

- Caso 1: Se utiliza el dataset generado después del pre-procesamiento.
- Caso 2: Se realiza la extracción de dos variables que tenían correlación.
- Caso 3:Se utiliza el dataset generado después del pre-procesamiento, pero se le extraen filas para balacear las clases.

Se muestra a continuación un resumen con los mejores resultados(menor error OOB) de cada caso:

5.1 Resultados caso 1: Dataset después del pre-procesamiento

Se generan 800 árboles con 1 nodo en cada división, con un error OOB del 7.39%, la tabla 3 muestra la matriz de confusión.

5.2 Resultados caso 2: Eliminando variables Longitud del eje Mayor y Shapfactor2

Al realizar una nueva matriz de correlación se observa que puede existir relación entre "ShapeFactor2" con 3 variables mas "Longitud del eje mayor", por lo que se eliminan y se realiza el algoritmo nuevamente.

Se generan 800 árboles con 1 nodo en cada división, con un error OOB del 22.38%, la tabla 4 muestra la matriz de confusión.

5.3 Resultados caso 3 (Mejor caso): Balanceando clases

Se utiliza el dataset del pre-procesamiento, pero se balancea las clases, ya que algunas clases (ej: Dermason o Sira) tenían muchos datos, se eliminaron filas del dataset, hasta que cada clase tuviera aproximadamente solo 1000 datos (a excepción de Bombay que solo eran 522).

Se generan 800 árboles con 1 nodo en cada división, con un error OOB del 3.55%, siendo el mejor caso. En la tabla 2 se muestra la matriz de confusión.

Table 2. Matriz de confusión Caso 3

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira	% Error
Barbunya	986	0	25	0	10	1	0	3.5%
Bombay	0	521	1	0	0	0	0	0.1%
Cali	16	0	1003	0	11	0	0	2.6%
Dermason	0	0	0	996	1	38	11	4.7%
Horoz	8	0	6	0	1003	0	11	2.4%
Seker	0	0	0	43	0	956	28	6.9%
Sira	3	0	0	0	15	10	1008	2.7%

Además, se presenta los errores por nodo luego de aplicar la función **randomForest** para 800 árboles (ver **Figura 1**). También,es posible obtener la importancia de las variables al aplicar el método de bosques aleatorios, para lo cual se ha obtenido el resultado presentado en la **Figura 6**.

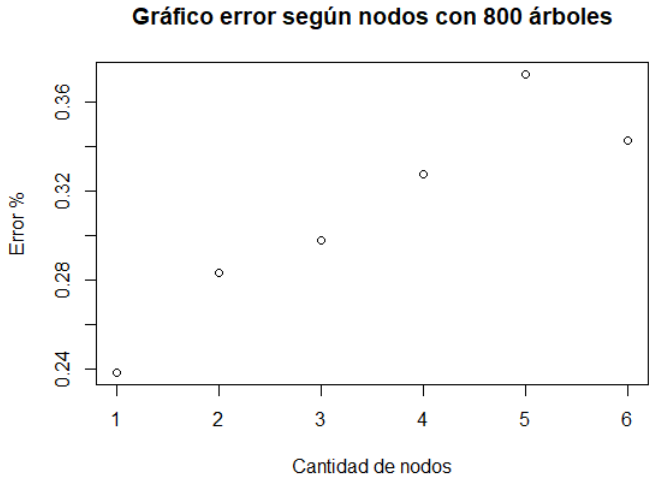

Fig. 1. Error OOB para 800 árboles y 1 nodo.

6 Discusión

En el caso 1 se puede observar que el error OOB es de 7.39%, lo cual es un buen indicador de que la clasificación ha sido correcta, también al observar la matriz de confusión (Tabla 3) los errores de las clases son bastantes bajas, solo la clase Sira presenta un error sobre el 10%. En el caso 2 el error OOB es de 22.38% aumentando considerablemente, al igual que lo errores de clasificación en las clases (Tabla 4), esto se debe a que al extraer el atributo de longitud del eje mayor, la cual es una de las variables de mas importancia (según Figura), esto afecto considerablemente en la clasificación.

En el caso 3, se obtuvo un error OOB de 3.55%, siendo mejor que en el caso 1, lo que nos demuestra que el balanceo benefició los resultados del algoritmo clasificador, esto se puede deber, porque en el caso al existir mucho datos de una clase, esto puede llevar al sobreajuste. La lectura indica que los frijoles que mas les costaba diferenciar eran entre la clase Dermason y Sira, en el caso 1 se puede observar en la matriz que son lo que mas error de clasificación tienen, mientras que en el caso 3 esto disminuyó, dando a entender que tal vez se necesitaba un balanceo de los datos anteriormente.

Ahora, en cuanto al gráfico de Coordenadas Paralelas del caso 3 (ver **Figura 7**), no queda muy claro la seperación entre la clase, solo se puede apreciar que la clase Horoz tiende mas a la variables "Extent", pero en si el gráfico no es muy claro, porque hay clase una sobre otras, tal vez la eliminación de otras variables ayudaría a verlo con mejor claridad.

7 Conclusiones

Con el objetivo de encontrar el modelo que mejor clasifique las clases, se realizó un pre-procesamiento, para después tener 3 casos distintos de dataset. Tanto el primer como segundo caso, se denotan con mayor error de clasificación, uno por sobrecarga de clases y el otro por extraer una variable importante. A partir, de esto se da a entender que extraer variables importantes alteran totalmente la clasificación, por lo que es importante notar cuales son para no sufrir este error. Y lo importante que es balancear las clases que puede ayudar a no sufrir sobreajuste en el proceso de clasificación.

En base a los casos anteriores, se pudo comprender que para poder implementar correctamente el método de Bosques Aleatorios haciendo uso de la función *randomForest*,se debe equilibrar el dataset en cuanto a su cantidad de observaciones por clase. Es de esto, donde nace un tercer caso, el cual mantuvo disminuyó el error OOB y permitió generar un mejor modelo de clasificación en cuanto al error entre clases. A partir de este caso, se pudo realizar un análisis más profundo del método implementado con sus correspondientes resultados, donde se pudo notar que la mejor cantidad de nodos a utilizar es de 1, mientras que la cantidad mínima de árboles a implementar para obtener un buen modelo es de 800. Aunque la diferencia de error entre los árboles eran mínima no variaba en mas de un 2% mas de error.

Finalmente, se puede decir que se cumplió el objetivo del laboratorio, donde se comprendió la teoría de Random Forest a partir de una aplicación práctica, demostrando el algoritmo de árboles aleatorios es mucho más exacto en cuanto a la clasificación de clases que el método de agrupamiento basado en modelos.

References

1. Koklu, M., Ali Ozkan, I. (2020, mayo). Multiclass classification of dry beans using computer vision and machine learning techniques. https://www.sciencedirect.com/science/article/pii/S0168169919311573?via

8 Anexos

A A Análisis Estadístico

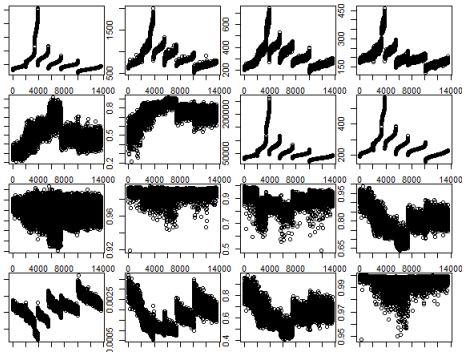

Fig. 2. Busqueda de Outliers en los datos de las variables

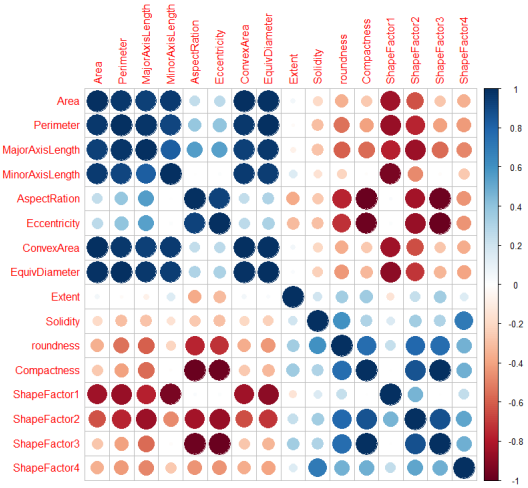

Fig. 3. Matriz de correlación con todas las variables

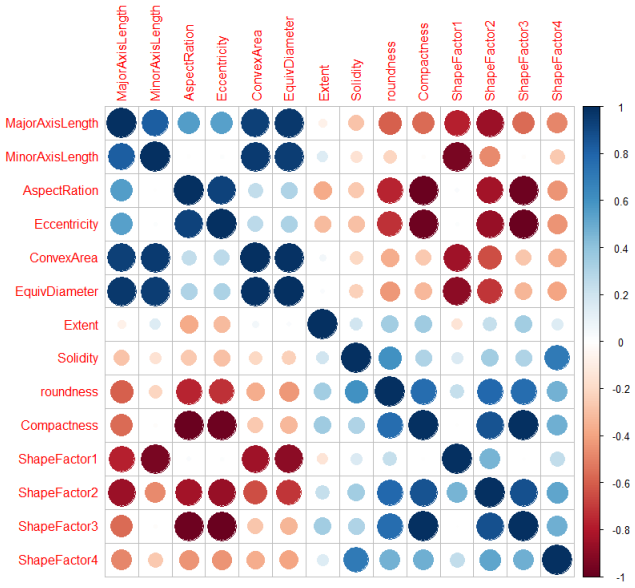

Fig. 4. Matriz de correlación con todas las variables

B B Aplicación Bosques Aletorios

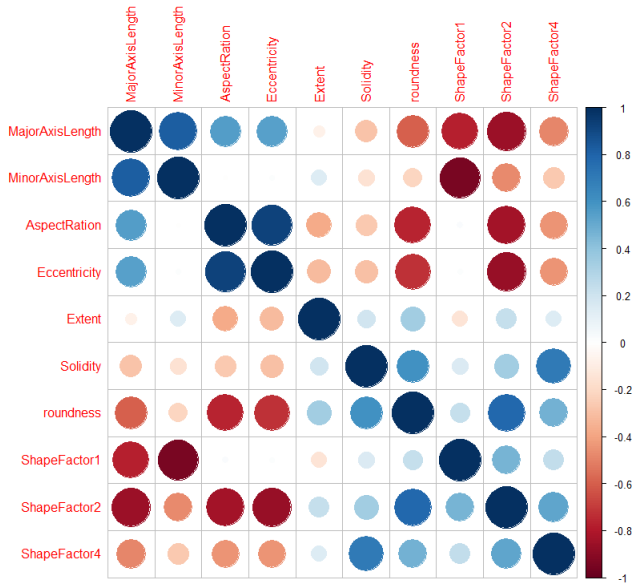

Fig. 5. Matriz de correlación con todas las variables

Table 3. Matriz de confusión Caso 1

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira	% Error
Barbunya	1210	0	55	0	12	13	32	8.5%
Bombay	1	518	2	0	1	0	0	0.8%
Cali	36	0	1541	0	32	3	18	5.5%
Dermason	0	0	0	3274	5	59	208	7.7%
Horoz	4	0	34	11	1829	0	50	5.1%
Seker	4	0	0	48	0	1915	60	5.5%
Sira	7	0	9	242	35	25	2318	12.1%

Table 4. Matriz de confusión Caso 2

Clases	Barbunya	Bombay	Cali	Dermason	Horoz	Seker	Sira	% Error
Barbunya	1132	8	22	31	18	17	94	14.4%
Bombay	20	137	171	31	9	10	144	73.8%
Cali	16	42	1246	7	105	2	212	23.5%
Dermason	85	8	26	2777	6	95	549	21.7%
Horoz	5	3	112	8	1777	0	23	7.8%
Seker	25	1	0	164	0	1825	12	9.9%
Sira	53	18	196	676	16	6	1671	36.6%

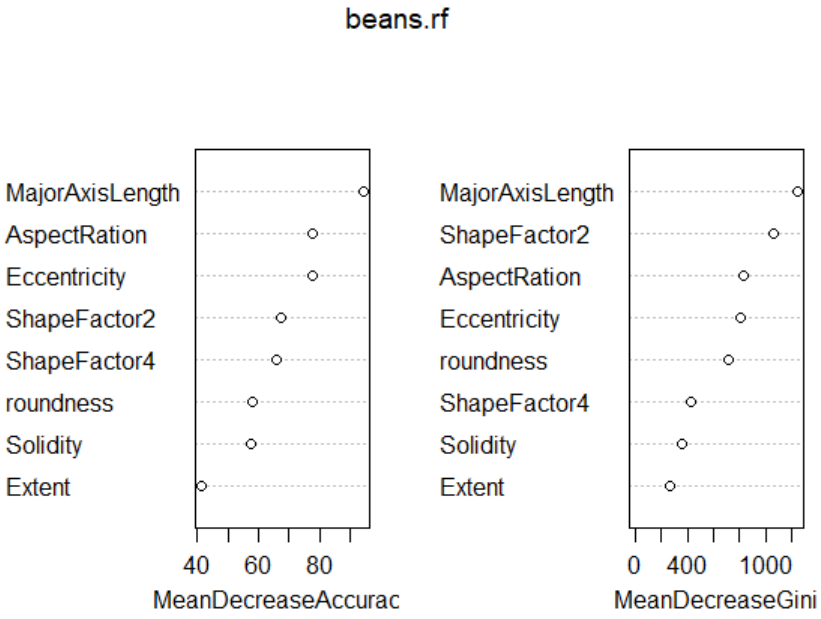

Fig. 6. Importancia de las variables para 800 árboles

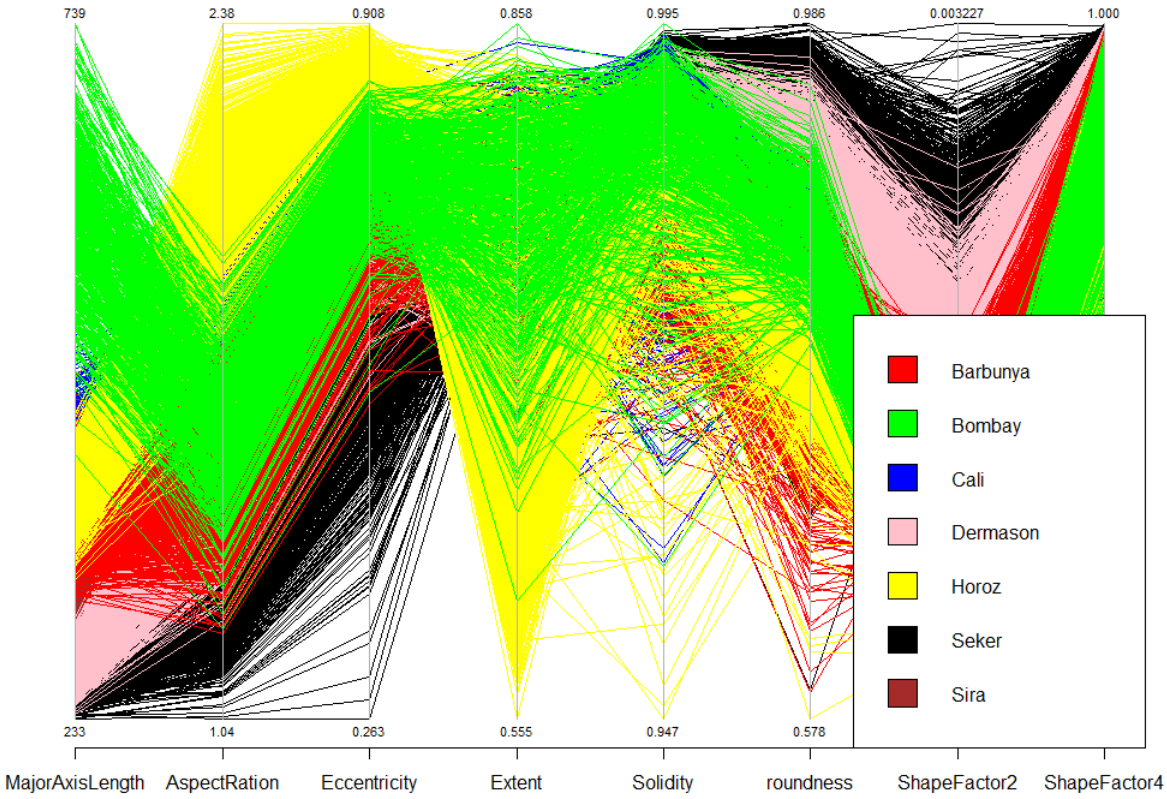

Fig. 7. Coordenadas Paralelas