Laboratorio 6 - ciencia de datos

Manuel Felipe Pineda L 1093223607

December 11, 2016

1 Base de Datos "segment"

Se utilizó la base de datos "segment" con las versiones provistas para entrenamiento y clasificación.

En esta base de datos se entregan diferentes características de un conjunto de instancias (regiones de 3x3 píxeles de una imagen) y se busca identificar a que segmento corresponde cada una de estas instancias.

1.1 Técnicas a utilizar

En esta sección se evalúan diferentes clasificadores para evaluar su desempeño con respecto a la base de datos.

Para cada clasificador se registran los tiempos de construcción y evaluación del modelo. También se registraran la cantidad de instancias clasificadas correctamente.

1.1.1 Clasificación - Naive Bayes Updateable

```
Time taken to build model: 0.05 seconds
Time taken to test model : 0.11 seconds
Correctly Classified Instances         624               77.037  %
Incorrectly Classified Instances       186               22.963  %
```

1.1.2 Clasificación - Logistic

```
Time taken to build model: 6.81 seconds
Time taken to test model : 0.04 seconds
Correctly Classified Instances         758               93.5802 %
Incorrectly Classified Instances        52                6.4198 %
```

1.1.3 Clasificación - Multilayer Perceptron

```
Time taken to build model: 5.32 seconds
Time taken to test model : 0.03 seconds
Correctly Classified Instances         769               94.9383 %
Incorrectly Classified Instances        41                5.0617 %
```

1.1.4 Clasificación - KStar

```
Time taken to build model: 0 seconds
Time taken to test model : 27.21 seconds
Correctly Classified Instances         781               96.4198 %
Incorrectly Classified Instances        29                3.5802 %
```

1.1.5 Clasificación - K-neighbors classifier

```
Time taken to build model: 0 seconds
Time taken to test model : 0.12 seconds
Correctly Classified Instances         776               95.8025 %
Incorrectly Classified Instances        34                4.1975 %
```

Observación

En este caso se puede notar que la clasificación basada en las muestras obtiene un mejor desempeño. Esto es debido a la naturaleza del problema, es muy probable que instancias que correspondan al mismo segmento estén ubicados en lugares cercanos en la imagen.

1.2 Clasificación basada en muestras

En esta sección se modifican algunos parámetros de los métodos que buscan similitud entre las muestras. También se compara con algunos métodos de clustering.

1.2.1 Kstar

Usando global bend = 50.

```
Time taken to build model: 0 seconds
Time taken to test model : 27.8 seconds
Correctly Classified Instances         787               97.1605 %
Incorrectly Classified Instances        23                2.8395 %
```

1.2.2 KNeighbors - Ball Tree

```
Time taken to build model: 0.08 seconds
Time taken to test model : 0.13 seconds
Correctly Classified Instances         776               95.8025 %
Incorrectly Classified Instances        34                4.1975 %
```

1.2.3 KNeighbors - Cover Tree

```
Time taken to build model: 0.07 seconds
Time taken to test model : 0.18 seconds
Correctly Classified Instances         773               95.4321 %
Incorrectly Classified Instances        37                4.5679 %
```

1.2.4 KNeighbors KD Tree

```
Time taken to build model: 0.02 seconds
Time taken to test model : 0.03 seconds
Correctly Classified Instances         767               94.6914 %
Incorrectly Classified Instances        43                5.3086 %
```

Observación

En este caso se puede observar que el clasificador K* tiene un mejor desempeño que KNeighbors. Ambos métodos se enfocan en encontrar medidas de distancia entre muestras pero con la diferencia de que K* utiliza una medida de distancia basada en la entropía y KNeighbors utiliza una medida de distancia basada en las características.

Sin embargo, es importante notar la diferencia de tiempos entre ambos métodos, según los experimentos con K* se puede lograr un mejor desempeño (1.4% mas) que con KNeighbors, pero en este caso es al rededor de 200 veces mas lento que KNeighbors.

1.2.5 Clustering - EM

```
Time taken to build model (full training data) : 0.42 seconds

Cluster Instances  Correct Value
0       214 ( 14%) 125
1       207 ( 14%) 110
2       132 (  9%) 122
3       186 ( 12%) 110
4       178 ( 12%) 126
5       217 ( 14%) 94
6       366 ( 24%) 123


Log likelihood: -25.55882
```

1.2.6 Hierarchical clustering

```
Time taken to build model (full training data) : 0.04 seconds

Cluster Instances  Correct Value
0      205 ( 25%) 125
1      123 ( 15%) 110
2      110 ( 14%) 122
3       44 (  5%) 110
4       50 (  6%) 126
5      154 ( 19%) 94
6      124 ( 15%) 123
```

1.3 Base de datos “glass”

En esta sección se comparan diferentes algoritmos de clustering y clasificación en la base de datos “glass” de weka.

1.4 Identificación automática de clases

En esta sección se utilizan algunos algoritmos para tratar de descubrir automáticamente la cantidad de clases en las cuales se deben separar los datos.

1.4.1 cluster - canopy

```
Time taken to build model (full training data) : 0.01 seconds
Classes 4
Incorrectly clustered instances : 120.0  56.0748 %
```

1.4.2 cluster - coweb

```
Time taken to build model (full training data) : 0.06 seconds
Classes 6
Incorrectly clustered instances : 115.0  53.7383 %
```

1.4.3 cluster - EM

```
Time taken to build model (full training data) : 0.64 seconds
Classes 2
Incorrectly clustered instances : 120.0  56.0748 %
```

1.5 Utilizando un numero fijo de clases

Utilizando 7 clases.

1.5.1 cluster - EM

```
Time taken to build model (full training data) : 0.04 seconds
Incorrectly clustered instances : 120.0  56.0748 %
```

1.5.2 cluster - Hierarchical

```
Time taken to build model (full training data) : 0.05 seconds
Incorrectly clustered instances : 136.0  63.5514 %
```

1.5.3 cluster - Kmeans

```
Time taken to build model (full training data) : 0.01 seconds
Incorrectly clustered instances : 118.0  55.1402 %
```

1.5.4 Clasificación - logistic

```
Time taken to build model: 0.39 seconds
Correctly Classified Instances         137               64.0187 %
Incorrectly Classified Instances        77               35.9813 %
```

1.5.5 Clasificación - multilayer perceptron

```
Time taken to build model: 0.55 seconds
Correctly Classified Instances         145               67.757  %
Incorrectly Classified Instances        69               32.243  %
```

1.5.6 Clasificación - KStar

Global Bend = 40

```
Time taken to build model: 0 seconds
Correctly Classified Instances         165               77.1028 %
Incorrectly Classified Instances        49               22.8972 %
```

1.5.7 Clasificación - Random Tree

```
Time taken to test model on training split: 0 seconds
Correctly Classified Instances          49               67.1233 %
Incorrectly Classified Instances        24               32.8767 %
```

Observación

Vale la pena notar que en este caso, proveer el numero original de clases no necesariamente mejora el desempeño de nuestros algoritmos de clustering.

También se puede notar que los algoritmos de clasificación generan mejores resultados para esta base de datos.