Procesamiento de texto manuscrito usando conjuntos de clasificadores

Emilio Samuel Aced Fuentes
Roberto Alcober Couso
Arturo Blázquez Pérez
Nicolás Trejo Moya

30 de diciembre de 2018

1. Introducción

En este proyecto vamos a clasificar imágenes de letras manuscritas intentando predecir de forma correcta el símbolo que representan. Utilizaremos varios algoritmos de clasificación y los compararemos para encontrar los mejores resultados posibles, viendo diferencias de tiempos y tasa de acierto.

2. Análisis de los datos

En el dataset provisto, están representadaslas 10 primeras letras del alfabeto(A-J), por tanto 10 clases las cuales hemos etiquetado en nuestra base de datos etiquetadas de 0 a 10 respectivamente. Las imagenes nos vienen en un tamaño de 206×150 en escala de grises.

Como podemos observar las imagenes contienen ruido, el cual podría dificultar la tarea de clasificación gravemente.Este problema lo hemos solucionado sometiendo a la imagen a diferentes tratamientos.

2.1. Tratamientos

Para todos los experimentos de este proyecto hemos umbralizado la imágen mediante otsu y posteriormente hemos realizado un filtro de mediana con un kernel cuadrado de tamaño 3. Esto nos ha dado el siguiente resultado:

A continuación explicaremos los tratamientos que le hemos aplicado a las imágenes para reducir la complejidad del problema

2.2. Atributos

Teniendo en cuenta que las imágenes son matrices 206×150 lo cual nos da un total de 30900 atributos, lo cual es una cantidad de atributos desorbitada, por ese motivo hemos decidido ver como afecta reducir la dimensión. Las principales modificaciones que realizaremos a las imagenes para reducir su dimensionalidad son las siguientes:

Figura 1: Ejemplo de una A

Figura 2: Ejemplo de una A tratada

Eliminar filas y columnas poco importantes Esto lo realizaremos mediante la selección de un umbral, con el cual consideraremos que las filas y columnas que tengan una cantidad menor de zona pintada que el umbral especificado no son de relevancia y por tanto las despreciaremos del problema

Figura 3: Ejemplo de una A recortada

Redimensionar la imágen interpolando Con el fin de evitar el aliasing (meter bordes que no existian al reducir una imágen) hemos redimensionado la imágen siempre aplicando un kernel gausiano previamente.

A continuacion mostraremos distintos valores de los tamaños de las imágenes que compararemos en nuestro estudio:

3. Modelos

Usaremos los siguientes clasificadores en este trabajo:

1. Random Forest
2. SVC
3. KNN

en los cuales variaremos los tamaños de las imágenes y compararemos tiempos y tasa de aciertos para poder decidirnos por uno.

Para todos ellos destinaremos la mitad de los datos para train y la otra mitad para test.

3.1. Random Forest

Con random forest hemos encontrado que funciona muy bien incluso reduciendo mucho la dimension de los datos encontrando que la mejor tasa de acierto

Figura 4: Varios tamaños de una A que usaremos

comparandola con el tiempo es reducir las imagenes a cuadrados 9×4 con 500 arboles de profundidad máxima 10 consiguiendo los siguientes resultados:

1. **Tasa de acierto**: 92.57 %
2. **Matriz de confusion**:

$$M = \begin{bmatrix} 69 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 48 & 0 & 5 & 2 & 3 & 0 & 0 & 4 & 0 \\ 0 & 0 & 73 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 59 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 54 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 61 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 66 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 67 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 58 & 6 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 56 \end{bmatrix}$$

3. **Ejemplos de clasificacion**

3.2. Clasificador SVC

Es el algoritmo que más tarda de los 3 con los que hemos experimentado, pero a su vez es el que consigue la mejor tasa de acierto. Para el ánalisis de resultados de este clasificador con estos datos decidimos reducir las imagenes a 100×50 con una gamma de 0.001 obteniendo los siguientes resultados:

1. **Tasa de acierto**: 96.21 %
2. **Matriz de confusion**:

Figura 5: Ejemplos de clasificacion con random forest

$$M = \begin{bmatrix} 69 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 57 & 0 & 3 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 74 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 62 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 57 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 66 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 70 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 61 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 57 \end{bmatrix}$$

3. **Ejemplos de clasificacion**

Figura 6: Ejemplos de clasificacion con random forest