Clasificador Bayesiano

Pablo Antonio Stack Sánchez
Programación y Algoritmos

23 de octubre de 2019

1. Introducción

El proceso de clasificación es algo natural para los seres humanos, la capacidad de abstraer información es esencial para la supervivencia, ayuda a la mejora en la toma de decisiones y permite diferenciar correctamente entre comida saludable y venenosa, o entre una presa y un depredador. Desde el punto de vista matematico la clasificación consiste en asignar una clase c, de un conjunto de clases C, a cierto objeto representado por un vector de características o rasgos $x = x_1, x_2, ..., x_m$. Existen dos tipos báasicos de clasificadores:
Supervisado:Las clases se conocen a priori, y el problema consiste en encontrar una funcion que asigne a cada objeto su clase correspondiente.
No supervisado:Se crean agrupamientos espectrales o clusters y se debe intentar asociar una clase a cada uno de dichos grupos, de forma que a objetos similares se les asigna la misma clase.
En esta tarea nos enfocaremos en los clasificadores supervisados, se utilizará la teoría de decisión Bayesiana para encontrar una función que realice un mapeo de los rasgos del objeto a su clase correspondiente, este método basado en probabilidad cuantifica los costos de las diferentes acciones asociados a las decisiones de clasificación. Para poder aplicar este método se debe asumir que todas las probabilidades en juego son conocidas.

2. Metodología

Se crearon dos archivos de texto:

- spam.txt
- correo.txt

spam.txt contiene una recopilación de correos spam y correo.txt contiene una recopilación de correos normales. Se utilizarón estos archivos para entrenar al clasificador bayesiano.
Se probó el clasificador con dos correos:

- correoaclasificar.txt: Correo de la bandeja de entrada el cual no es spam.
- spamaclasificar.txt: Correo de la bandeja de spam.

Correoaclasificar.txt

hola chicos les envio algunas ultimas correcciones por favor agreguenlas a su ultimo documento que corrigieron agreguenlo a dropbox en word y pdf por favor si tienen alguna duda con algun comentario o correccion haganmelo saber por este medio saludos

Spamaclasificar.txt

tu opinion es muy importante para nosotros y nos permitira detectar aquellos aspectos que no debemos descuidar y aquellos en los que debemos de seguir mejorando te agradeceremos nos regales unos minutos de tu tiempo para contestar la siguiente encuesta

A continuación se muestra el algoritmo utilizado:
it: Iterador del mapa Spam; it2: Iterador del mapa Correo; itpr: Iterador del mapa Prueba;

```
ifstream abrir "spam.txt";
ifstream abrir2 çorreo.txt";
ifstream abrir3 çoreoaclasificar.txt";
map <string, int>Spam, Correo, Prueba;
while do
    Correo[abrir]++;
    Spam[abrir2]++;
    Prueba[abrir3]++;
end
//calcular intersección entre las palabras de Correo y Spam;
for do
    if it- > first == it2- > first then
        Resultado[it- > first] = make_pair(it2- > second, it- > second);
        it + +;
        it2 + +;
    end
    else if it- > first < it2- > first then
        it + +;
    else
    it2 + +;
end
for itpr = prueba.begin(); itpr! = prueba.end(); itpr + + do
    pnospam* = (double)Resultado[itpr- > first].first/total_correo;
    pspam* = (double)Resultado[itpr- > first].second/total_spam;
end
cout << "ProbabilidaddeSpam : " << pspam << "ProbabilidaddenoSpam : " <<
 pnospam << endl;
```

Algorithm 1: Clasificador Bayesiano

3. Resultados

En la figura 1 se muestra el resultado obtenido al probar con el archivo de texto correoaclasificar.txt, se observa que el algoritmo funciona de manera adecuada ya que fue clasificado como correo no spam.

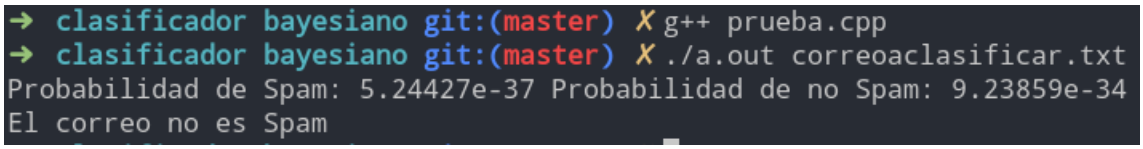

Figura 1: Resultado de probar con correoaclasificartxt

En la figura 2 se presenta el resultado al probar con spamaclasificar.txt

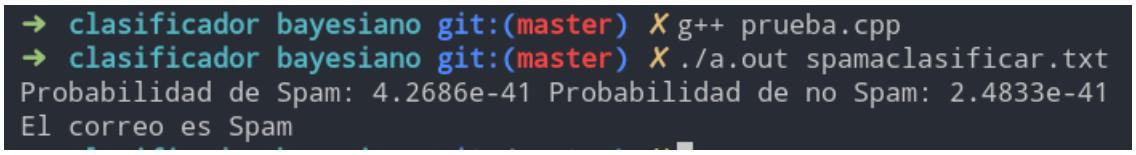

Figura 2: Resultado de probar con spamaclasificartxt

4. Conclusión

El clasificador Bayesiano ingenuo fue efectivo en la tarea de clasificar correos spam y no spam. Algunas mejoras que se podrían hacer son:

- Aumentar la base de datos
- Estimar las probabilidades apriori de que un correo pertenezca a determinada clase

Componentes Conectados

Pablo Antonio Stack Sánchez
Programación y Algoritmos

23 de octubre de 2019

1. Introducción

Un componente fuertemente conectado en la teoría de grafos es llamadado así si para cada par de vértices u y v existe un camino de u hacia v y un camino de v hacia u. En el caso de imagenes se dice que un componente es conectado si el pixel está de alguna forma relacionado con sus 8 vecinos.
Para esta tarea se implementó un DFS para encontrar todos los componentes conectados de una imagen en formato pgm.

2. Metodología

En primer lugar se busca un pixel que sea diferente de negro y se le aplica un DFS, recorremos la imagen como si fuera un grafo buscando los componentes conectados. En este caso los nodos del grafo serían los pixeles de la imagen y las aristas serían valores de intensidad diferentes de 0 que se comparten en una 8-vecindad.

Algoritmo:

1)Buscar un pixel diferente de 0.
2)Si no ha sido visitado antes:
2.1)Aplicar DFS.
2.2)Visitar cada uno de sus 8 vecinos y si no han sido visitados:
2.3)Aplicar DFS.
3)Regresar a 1) hasta recorrer toda la imagen.

3. Resultados

Se probó el algoritmo con 5 imágenes distintas que se adjuntan en el zip de la tarea, a continuación se muestra un ejemplo:

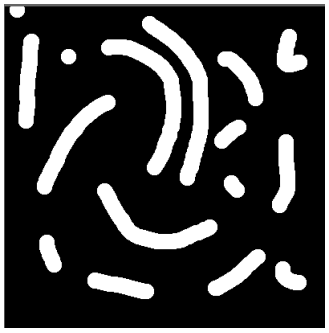

Figura 1: Primera imagen

```
Componente: 0 Tamaño: 230
Componente: 1 Tamaño: 4587
Componente: 2 Tamaño: 1321
Componente: 3 Tamaño: 2221
Componente: 4 Tamaño: 4322
Componente: 5 Tamaño: 293
Componente: 6 Tamaño: 1582
Componente: 7 Tamaño: 2929
Componente: 8 Tamaño: 823
Componente: 9 Tamaño: 1862
Componente: 10 Tamaño: 517
Componente: 11 Tamaño: 3518
Componente: 12 Tamaño: 874
Componente: 13 Tamaño: 1586
Componente: 14 Tamaño: 881
Componente: 15 Tamaño: 1587

maximo: 4587 i:12 j:177
minimo: 230 i:0 j:7
```

Figura 2: Número de componentes conectados

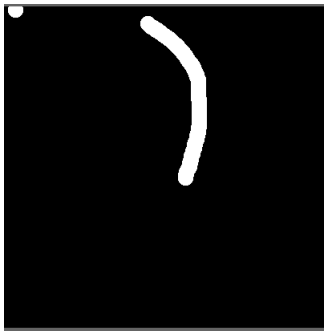

Figura 3: Primera imagen

4. Conclusiones

El algoritmo funciona bien para la mayoría de las imágenes probadas. Se tuvo un problema con imágenes que tuvieran más columnas que filas, el programa tiene un segmentation fault en estos casos. Se intentó solucionar pero no pude encontrar el error.