MONOGRAFÍA: COMPUTACIÓN EVOLUTIVA

Clasificación y selección de características mediante algoritmos GA-P

Luciano Sánchez[1], Luis Junco[1], Fernando Cano[1], Inés Couso[2]

Departamento de Informática (1)
Departamento y Estadística e I.O. y D.M (2)
Universidad de Oviedo
e-mail: luciano@lsi.uniovi.es

Resumen

La fase más costosa de algunos problemas de clasificación es aquella en la que se asocia el objeto que se desea clasificar con un vector de atributos numéricos. Un buen algoritmo de selección de características es importante en estos casos porque, además de mejorar la capacidad de generalización del clasificador, puede reducir notablemente el tiempo de proceso.

La mayoria de los sistemas de clasificación están diseñados para optimizar su error y no tienen en cuenta el coste computacional de calcular cada atributo. Por esta razón, operan con un vector de características de tamaño fijo. No obstante, es frecuente que algunas clases puedan discriminarse basándose solamente en un subconjunto de los atributos. En este trabajo se propone un sistema de clasificación que calcula los atributos secuencialmente y decide la clase a la que pertenece el objeto tan pronto como tiene información suficiente para hacerlo, lo que sirve para ahorrar cálculos innecesarios. La metodología seguida en este diseño se basa en aprendizaje automático mediante computación evolutiva (algoritmos GA-P). Se comprobará experimentalmente que el sistema que se propone, además de ser más eficiente en tiempo de cálculo, no es menos potente que los clasificadores estadísticos y neuronales más frecuentes.

Palabras Clave: Algoritmos GA-P. Programación Genetica. Clasificación.

1. Introducción

De cara a un sistema de clasificación, un objeto se suele caracterizar mediante un vector de atributos. Este vector suele contener información redundante para el clasificador, por lo que, mediante un proceso conocido como *extracción de características*, algunas de sus coordenadas se eliminan o se cambian por combinaciones no lineales de las demás. El objetivo del proceso es doble: lograr un vector cuyas componentes sean estadísticamente independientes y que las superficies de separación entre las distintas clases de objetos sean representables por el clasificador. Por ejemplo, si empleamos un clasificador basado en funciones discriminantes lineales, es necesario que el proceso de extracción de características transforme el problema original en otro separable linealmente. Si, por el contrario, empleamos un clasificador neuronal, esto último no sería necesario.

Hasta donde nosotros conocemos, la extracción de características está orientada a mejorar la potencia de clasificación del sistema (redu-

ciendo la dimension del vector de entrada se aumenta la densidad de ejemplos). Sin embargo, existen problemas en los que disponemos de información suficiente como para discriminar bien las clases, pero en los que el cálculo de los atributos es muy costoso. El caso más característico se da en la visión artificial. Por ejemplo, en la clasificación de texturas podemos disponer de varias decenas de atributos que caractericen a un tipo de textura, pero el tiempo necesario para evaluarlos todos en cada muestra es desmesurado. Es necesario llegar a un compromiso entre la potencia de clasificación y los atributos empleados.

En este papel se propone una forma de lograr este compromiso. Mediante programación genética (algoritmos GA-P) se ha realizado un sistema que determina automáticamente qué operaciones deben llevarse a cabo con los atributos para llegar a un vector de características que permita realizar una clasificación eficaz y consiguiendo a la vez que

1. No todos los atributos intervengan en las características
2. No todas las características intervengan en la clasificación de todas las muestras.

El sistema que se expone tiene doble utilidad. Por una parte es un clasificador optimizado para ahorrar tiempo de proceso en el cálculo de los atributos. Por otra parte, mostraremos que también sirve para identificar qué atributos son necesarios para separar las clases y por tanto es un extractor de características que puede usarse en combinación con otros clasificadores de cualquier tipo.

2. Planteamiento

Supongamos que se dispone de una población C que consta de N subpoblaciones C_i, y que deseamos conocer a cuál de estas subpoblaciones pertenece un objeto $\omega \in C$. Supongamos también que, de cara a nuestro sistema de clasificación, cada objeto se caracterizará mediante un vector de atributos numéricos $\mathbf{a}(\omega) = (a_1, \ldots, a_n)$.

Habitualmente este vector a no se emplea directamente, porque es redundante. Los atributos que no aportan información se eliminan sin más del vector o bien se compone un nuevo vector cuyas componentes sean funciones de los atributos útiles, lo que suele ser más eficaz. Llamaremos *vector de características* al

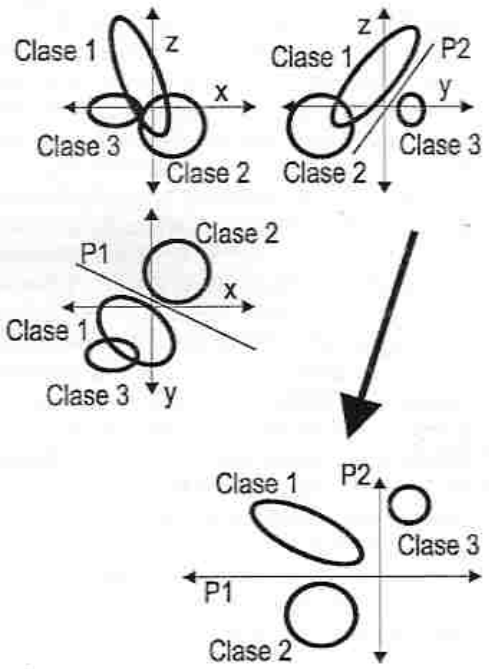

Figura 1: Estructuración de un problema de clasificación

vector transformado y lo denotaremos por $\mathbf{f} = (f_1, \ldots, f_p)$, con $p <= n$ y $f_i = f_i(\mathbf{a})$. La transformación de a en f se denomina *extracción de características*. Idealmente, el conjunto de características usado en un sistema de clasificación debería ser mínimo y por tanto estadísticamente independiente, en el sentido de que ninguna de las características debería poder ser determinada mediante una función de las demás. Pero la elección de un conjunto de funciones f_i con estas propiedades es un proceso complejo. Puede automatizarse mediante procedimientos de análisis de correlación (potencia de discriminación) o de ponderación de características, pero los métodos estadísticos de selección de características son intrínsecamente inadecuados para clases separables no linealmente (ver [2], pág 339).

La selección de las mejores características puede ser entendida como un cambio de variable seguido de una proyección. El objetivo que pretendemos se muestra gráficamente en la figura 1. En la parte superior de dicha figura se muestran las tres vistas de un problema de clasificacion de vectores de tres dimensiones. En la parte inferior de la figura se muestra como un cambio de variable seguido de una proyeccion

transforma el problema en otro de dos dimensiones. En general no se ahorra tiempo con este cambio, ya que para en el cálculo de estas dos características necesariamente han de intervenir los tres atributos, según se aprecia en la figura. Nosotros proponemos un clasificador que evalue primero la característica $P1$, que es función de x y de y. Con $P1$ podemos decidir si el objeto es de la clase 3 o no. Si lo es, no es necesario evaluar $P2$, que depende de y y de z y por tanto no hemos evaluado el atributo z. Por supuesto, si el objeto no es de la clase 3 debemos calcular los tres atributos y por tanto la ganancia de tiempo será tanto mayor cuanto más probable sea la aparición de un objeto de la clase 3 y cuanto más costosa sea la evaluación de z.

Con esta idea en mente, a continuación se expone una metodología, basada en los algoritmos GA-P –un método híbrido entre programación genética y algoritmos genéticos– con la que se tratará de estructurar cualquier problema de clasificación en la forma anterior.

3. Definición del clasificador

Para estructurar el problema de clasificación tal y como se ha mostrado en la figura 1 debemos encontrar, en primer lugar, una función f_1 tal que $f_1(\mathrm{a}(\omega)) > 0$ si $\omega = (x, y, z)$ pertenece a la clase 3 y menor o igual que cero en caso contrario. A continuación eliminamos los puntos de la clase 3 y buscamos una función f_2 positiva en los puntos de la clase 2, con lo que el problema queda resuelto. La extensión a un número mayor de clases es inmediata.

Obviamente, si existe f_1 también podemos hablar de una superficie de separación $f_1(\mathrm{a}) = 0$ entre la clase 3 y las demás, con lo que estamos admitiendo que para realizar esta estructuración será necesario que las clases sean separables. Nos es conveniente extender el problema a clases no separables, y para ello adaptaremos la formulación bayesiana más sencilla [3] y para clasificar objetos de N clases $C_1, \ldots, C_N$ buscaremos N funciones f_i que minimicen las tasas de error ϵ_i definidas como $\epsilon_i = P\{\omega \in C_i \mid f_i(\mathrm{a}(\omega)) < 0\} + P\{\omega \in C - \cup_{j=1\ldots i} C_j \mid f_i(\mathrm{a}(\omega)) \geq 0\}$. Hemos supuesto que la selección del objeto que se desea clasificar es un experimento aleatorio, con lo que tiene sentido definir la probabilidad $P(\omega \in C_i)$. Como deseamos optimizar el tiempo de cálculo, buscaremos funciones f_i que no dependan de todos los atributos a_i. Obsérvese que los vectores de características

```
si f_1(a)>0 entonces
   a pertenece a la clase 1
si no
   si f_2(a)>0 entonces
     a pertenece a la clase 2
   si no
      ...
      si f_N(a)>0 entonces
        a pertenece a la clase N
      si no
        punto no clasificable
      fin si
   fin si
fin si
```

Figura 2: Clasificador en el espacio de características estructurado

tendrán N componentes y que la definición del clasificador asociado a ellos es trivial (ver Figura 2). En este trabajo se admitirá que el cálculo de todos los atributos a_i tiene el mismo coste computacional y que la situación más favorable en promedio se obtendrá indexando las clases de forma que $P(C_1) > P(C_2) > \ldots > P(C_N)$. Esto último no es cierto más que cuando todas las f_i dependen de un número similar de atributos, lo que no necesariamente es cierto; actualmente estamos trabajando en el problema de encontrar el mejor orden de clasificación.

4. Solución numérica

Los algoritmos GA-P son un método híbrido entre los algoritmos genéticos y la programación genética, optimizado originalmente para realizar regresiones simbólicas. La descripción completa del método (que no reproducimos aquí por razones de espacio) se encuentra en [7], y algunas de sus aplicaciones a otros problemas en [4, 5].

4.1. Aplicación al problema

Los algoritmos GA-P proporcionan una metodología aplicable directamente al problema anterior, ya que realizan una tarea de búsqueda en un conjunto de programas que cumplen determinadas restricciones, y puede limitarse la búsqueda a los programas que no superen un número dado de instrucciones. La aplicación de la programación genética a la clasificación no es una idea nueva (ver [1]) aunque los algoritmos

GA-P solamente han sido aplicados a problemas de regresión. Creemos que este método es una alternativa muy conveniente porque su manejo de los valores de las constantes numéricas tiene como consecuencia que los árboles en que se codifican las soluciones tienden a ser menores que los obtenidos al emplear programación genética canónica.

Para resolver un problema de clasificación de múltiples clases buscaremos separada y sucesivamente las N funciones f_i que maximizan un valor de adecuación (*fitness*) definido como

$$\#\{k \in \{1,\ldots,m\} \mid \mathbf{a}^k \in C_i \wedge f_i(\mathbf{a}^k) \geq 0\}$$

restringiendo la búsqueda a las funciones que cumplan que $\#\{k \in \{1,\ldots,m\} \mid \mathbf{a}^k \in C - \cup_{j=1\ldots i} C_j \wedge f_i(\mathbf{a}^k) > 0\} = 0$ donde $M = \{\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^m\}$ es la muestra de objetos clasificados con la que construimos el clasificador. La expresión que se minimiza, por tanto, es $N_i - \#\{k \in \{1,\ldots,m\} \mid \mathbf{a}^k \in C_i \wedge f_i(\mathbf{a}^k) \geq 0\} + \lambda\#\{k \in 1,\ldots,m\} \mid \mathbf{a}^k \in C - \cup_{j=1\ldots i} C_j \wedge f_i(\mathbf{a}^k)) > 0\}$ donde N_i es el número de elementos de la clase i y λ es un factor de penalización (análogo a un multiplicador de Lagrange). Además, ante dos valores de adecuación iguales se prima la función f_i más simple (la que se pueda codificar en menor número de nodos). Nótese que no se ha incluido en la función de adecuación ningún término que dependa del coste computacional de cada atributo. Este término se incluirá en trabajos futuros.

4.2. Aprendizaje cooperativo

En la implementación canónica de GP se busca que un individuo aislado clasifique correctamente la totalidad de los puntos; cada clasificador tiene asociada una bondad (normalmente, la fracción de puntos mal clasificados). Recientemente [9] se ha aplicado a la clasificación un esquema de aprendizaje competitivo, en el que la puntuación de cada individuo depende de los puntos que solamente él puede clasificar, de modo que la recompensa obtenida por la clasificación de un punto se reparte entre todos los individuos que lo discriminan. De esta forma se evita que los individuos capaces de clasificar un número bajo de muestras sean eliminados si ninguna otra función es capaz de separar esos mismos puntos. Este aprendizaje es difícil de implementar en régimen permanente (*steady-state*) dado que la puntuación de un individuo concreto puede cambiar al introducir o eliminar un individuo de la población.

Nosotros vamos a extender este tipo de evolución mediante una técnica que es común en el aprendizaje de reglas borrosas mediante algoritmos genéticos, la conocida como "enfoque Michigan": el clasificador no estará compuesto por el mejor individuo, sino por toda la población en conjunto, que coopera para llegar a la mejor clasificación. Por "cooperación" no entendemos "empleo de técnicas distintas de búsqueda". Empleamos el término en el mismo sentido que en el "problema de cooperación-competición" del aprendizaje de reglas borrosas mediante algoritmos genéticos.

Más concretamente, nuestra estrategia de comparación consiste en lo siguiente: si cada individuo de la población toma valores ***positivos en un subconjunto de la clase que se desea separar, y negativos en el resto de los puntos*** (todos los de las demas clases y tal vez algunos de la clase que separamos) el ***máximo*** de las evaluaciones de todos los programas en cada punto discrimina el conjunto formado por la union de los conjuntos que discrimina cada individuo. Luego podemos hacer evolucionar subpoblaciones y definir f_i como el máximo de las evaluaciones de los mejores individuos de cada subpoblación.

De este modo, cada individuo puede tener una codificación sencilla, y el problema se puede segmentar automáticamente conforme algunos individuos van clasificando parte de las muestras. Pero este funcionamiento requiere modificar la operación de reproducción GA-P. En el momento de enviar este artículo estamos empleando un enfoque en régimen permanente y un tamaño de población variable, y se impone que si un individuo es el único que discrimina alguno de los ejemplos este individuo no se puede eliminar de la población. Al realizar un cruce eliminamos siempre un individuo que pueda extraerse sin que cambien los puntos clasificados por toda la población. Si eso no es posible, aumentamos la población en una unidad y permitimos que se inserte el mejor descendiente del cruce. Si no se hiciera así, llegaríamos frecuentemente a un estancamiento, a menos que el tamaño de población fuese bastante mayor que el número de muestras, lo que no suele ser operativo. Periódicamente, la población se purga eliminando algunos individuos innecesarios, de forma que su tamaño no aumente excesivamente.

Todo este proceso se ve acelerado notablemente por el empleo de la optimización local en la parte GA [8]. Se ha empleado el método Simplex de Nelder y Mead, también conocido como

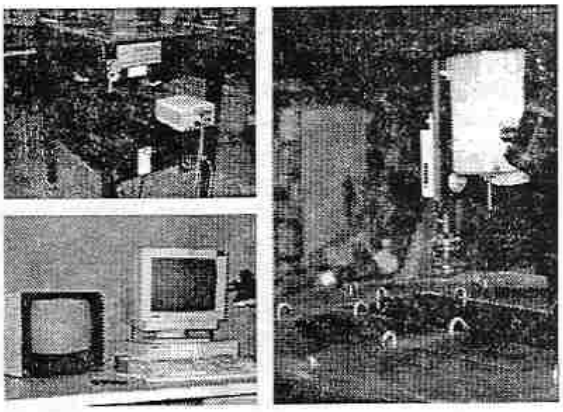

Figura 3: Sistema automático de búsqueda de defectos en vidrio

"poliedro flexible".

5. Resultados experimentales

5.1. Defectos en planchas de vidrio

En la automatización de una de las fases del control de calidad que se realiza en la industria de la fabricación del vidrio es preciso localizar y clasificar los defectos presentes en las planchas de vidrio aparecidos durante el proceso de fabricación.

El clasificador implementado actualmente en el sistema está basado en conjuntos borrosos, y se construyó originalmente a partir de una descripción imprecisa de las características de los defectos, realizada por el propio operador. Los parámetros que definen cada etiqueta lingüística fueron obtenidos automáticamente, mediante un algoritmo de optimización genético.

Cada defecto se clasifica en una de las siguientes cuatro categorías: burbujas, manchas de estaño, piedras, y polvo o elemento extraño (ver Figura 4). Se procesan un total de 6.500 imágenes para cubrir completamente la superficie total de una muestra de cristal estándar (de 600x100 mm^2). Además, la medición de la profundidad conlleva la adquisición y análisis de dos nuevas imágenes por cada defecto. Dado el elevado número de imágenes que se están analizando, la velocidad del clasificador es muy importante.

Los estadísticos empleados para realizar la discriminación son 5: contraste, brillo, grado de agrupación, profundidad y momento de inercia. Los errores de clasificación obtenidos mediante el clasificador borroso, un clasificador estadístico y la clasificación GA-P se muestran a continuación:

	C. Borroso	k-vecinos	GA-P
Entr.	5.8	-	2.8
Test	7.8	12.6	3.3

La mejora de porcentajes es significativa de por sí, pero la ventaja fundamental del clasificador radica en que, con el clasificador GA-P, el estadístico "Brillo" solamente se calcula un 20% de las muestras, y el estadístico "Profundidad", considerado como muy relevante por el experto humano, no es necesario en ningún caso.

5.2. Clasificación de Texturas

A continuación resolvemos un problema de caracterización de texturas extraídas de álbum de Brodatz[10]. Extraemos 32 atributos basados en el histograma, la dimensión fractal, CGM, GLDM y Gabor. Los resultados (porcentaje de aciertos en cada una de las 6 clases que hemos elegido) se muestran en la parte superior de la figura 5.

Con GA-P empleamos un total 15 de los 32 atributos y estimamos que se necesitan un 29.37% de las operaciones realizadas con el clasificador neuronal. A continuación hemos probado la faceta de extractor de características de nuestro método, diseñando clasificadores sobre las 15 características seleccionadas. Todos los clasificadores mejoraron, como puede verse en la parte inferior de la figura 5.

6. Conclusiones

En algunos problemas de clasificación la mayor parte del tiempo de cálculo se emplea en calcular los atributos. Mediante un proceso de extracción de características se determina cuáles de estos atributos son relevantes, lo que supone un ahorro de tiempo. Pero aun en el caso de que el vector de características sea mínimo (porque todas sus coordenadas sean funcionalmente independientes entre sí) podría ser posible reducir el tiempo medio de cálculo; algunas clases podrían discriminarse de las demás con ayuda de un subconjunto de las características. El ahorro sería función de las probabilidades de aparición de estas clases.

En este trabajo se ha expuesto una metodología que estructura un problema arbitrario de forma que puede aplicársele un clasificador con

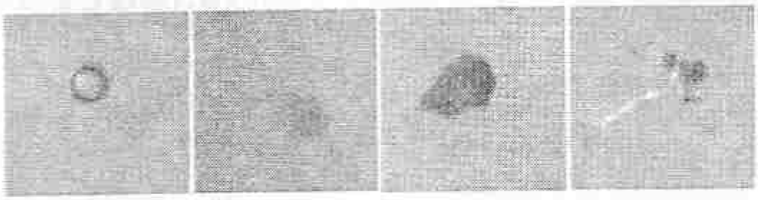

Figura 4: Defectos: Burbuja, estaño, piedra y mota de polvo

Clase	1	2	3	4	5	6	Total
R. Neur.	97.5	100	97.4	100	96.7	96.9	98.1
Distancias	90	97.4	100	87.9	96.7	100	95.2
K-Vecinos	100	100	100	93.9	96.7	90.6	97.1
GA-P	100	100	97.1	87.9	90.0	96.9	95.7

Resultados comparados del clasificador GA-P

Clase	1	2	3	4	5	6	Total
Red Neuronal	100	100	100	97	100	94	98.6
Distancias	100	97.4	100	98	96.7	100	98.6
K-Vecinos	100	100	100	97	93.3	97	98

Figura 5: Resultados obtenidos tras emplear GA-P para seleccionar atributos

la propiedad antes mencionada. Hemos comprobado experimentalmente que, en algunos problemas, este método consigue reducir el coste computacional sin que ello implique un aumento del error de clasificación.

Referencias

[1] Koza, J. R. (1992) *Genetic Programming: on the programming of computers by means of natural selection.* MIT Press.

[2] Looney, C. G. (1997) *Pattern Recognition Using Neural Networks.* Oxford University Press.

[3] Hand, D. J. (1981) *Discrimination and Classification.* Wiley.

[4] Sánchez, L., Couso I. (1997) "Modelado a partir de ejemplos imprecisos mediante algoritmos GA-P". ESTYLF 97. VII Congreso Español sobre Tecnologías y Lógica Fuzzy. Tarragona.

[5] Cordón O., Herrera F., y Sánchez L. (1997) "Evolutionary learning processes for data analysis in engineering applications". En Quagliarelli D., Periaux J., Poloni C. y Winter G. (Eds.) *Genetic Algorithms in Engineering and Computer Science.* John Wiley and Sons.

[6] Junco, L. y Sánchez, L. (1997) *Localization and fuzzy classification of manufacturing defects in sheets of glass.* Admitido en Mathware and Soft Computing.

[7] Howard, L.; D'Angelo, D. (1995) "The GA-P: A Genetic Algorithm and Genetic Programming Hybrid". IEEE Expert. June 1995. 11-15.

[8] Renders, J.M., Flasse, S. P. (1996) "Hybrid Methods Using Genetic Algorithms for Global Optimization". IEEE Transactions on SMC. Part B: Cybernetics. Vol 26, NO. 2, April 1996.

[9] Hugues, Pollack (1996) "Co-evolving Intertwined Spirals". Proceedings of the Fifth Annual Conference on Evolutionary Programming. San Diego, CA. MIT Press.

[10] Brodatz P. (1966) Textures: A Photographic Album for artists and Designes. New York, Dover Publications