QÜESTIIÓ, Vol 13, nº 1,2,3 pp. 31-46, 1989

GRAMÁTICAS DISCRIMINANTES Y FUNCIONES DISCRIMINANTES LINEALES GENERALIZADAS

ANA GARCÍA FORNES
ANTONIO RUIZ CALOMARDE
* FRANCISCO CASACUBERTA NOLLA
* ENRIQUE VIDAL RUIZ
Universidad Politécnica de Valencia

Las Gramáticas Discriminantes constituyen una aproximación para la clasificación de frases generadas por Gramáticas cuando la presencia de ruidos y distorsiones hace difícil la aplicación de las Técnicas usuales de Análisis Sintáctico. Sin embargo la formulación original (Filipski 80) presenta una restricción pues las gramáticas características deben ser las mismas para todas las clases. En este trabajo se presenta una solución al problema, mediante la aplicación de una Extensión de las Funciones Discriminantes Lineales Generalizadas, junto con una adaptación del Método de Aprendizaje denominado algoritmo "Pocket" propuesto originalmente en (Gallant 86). Se presentan experimentos con resultados para frases con distribuciones de longitud determinadas, que confirman la mejor aproximación dada por las Gramáticas Discriminantes dotadas de la extensión utilizada, respecto de la ofrecida por la aproximación clásica basada en Gramáticas Estocásticas.

Discriminant Grammars and Generalized Linear Discriminant Functions.

Keywords: Sintactic Pattern Recognition, Discriminant Grammars, Stochastic Grammars, Substring length modelling.

–Ana García Fornes - Antonio Ruiz Calomarde - Enrique Vidal Ruiz - Dept. de Sistemes Informàtics i Computació - Universitat Politècnica de València.
*Treball parcialment sostingut pel projecte TIC0448/89 de la Comissió Interministerial de Ciència i Tecnologia.
–Article rebut el setembre de 1989.

1. INTRODUCCIÓN

El reconocimiento de Formas (RF) se ha constituído en una de las herramientas más importantes para el desarrollo de los sistemas perceptivos en las máquinas. Una de las principales facetas del RF consiste en la **clasificación** de unos datos de entrada en **clases** identificables ("patrones" o "formas") según unas características o atributos determinados.

Existen básicamente dos aproximaciones al RF: la basada en la *Teoría de la Decisión (TD)* o *Reconocimiento Geométrico de Formas (RGF)*, y el *Reconocimiento Estructural o Sintáctico de Formas (RSF)* (González 78).

El RGF asume que los objetos son representables como puntos en cierto espacio vectorial (o al menos, métrico), y se basa en el uso de funciones de decisión para clasificar dichos objetos. Estas funciones reciben también el nombre de discriminantes.

El RSF se basa en conceptos de la Teoría de Lenguajes. Un aspecto básico del RSF es la descomposición de los objetos en primitivas o subpatrones. Estas primitivas se consideran como elementos terminales de una gramática. Cada clase está representada por una gramática, de manera que el lenguaje generado por ésta contiene el conjunto de patrones de la clase.

No obstante, la presencia de distorsiones y ruido en los objetos a reconocer, siempre existente en la práctica, hacen que el análisis de una frase con respecto a una Gramática, incluso si ésta es de Contexto Libre (GCL) no ambigua, no sea una tarea directamente abordable con las técnicas usuales de Análisis Sintáctico, con lo que la tarea de clasificación no resulta trivial.

Una aproximación a este problema la dan las llamadas **Gramáticas Discriminantes (GD)** (Filipski 80). Según esta aproximación, a cada regla de una gramática se le asocia un vector de tantos pesos reales como clases. La clasificación consiste en sumar los vectores de pesos de todas las reglas utilizadas en el reconocimiento de la cadena y clasificar ésta en la clase cuya componente asociada del vector suma es mayor.

Este proceso de reconocimiento se puede interpretar en términos de Reconocimiento Geométrico de Formas. Para ello se asocia a cada frase un vector en el espacio de "índices estructurales". Dicho vector tiene tantas componentes como reglas tiene la gramática y el valor de cada componente es el número de veces que se ha utilizado cada regla en el análisis sintáctico de la frase asociada. A esta representación vectorial se pueden aplicar todos los métodos clásicos de RF basados en la TD. En particular Filipski propone el uso de métodos no paramétricos basados en minimizar el error cuadrático medio (criterio LMSE, (Duda 73)) para realizar el **aprendizaje** de los vectores de pesos mencionados anteriormente.

Filipski muestra que la clasificación basada en GD con el esquema de aprendizaje LMSE, se aproxima al criterio de decisión de Bayes cuando las muestras han sido generadas por gramáticas estocásticas de contexto libre no ambiguas. También muestra que incluso pude ser mejor, si la distribución de longitudes de las (sub)cadenas es uniforme o de tipo Poisson.

Sin embargo, existe un inconveniente en las GD: todas las clases deben tener la misma gramática característica. Filipski lo reconoce, pero indica que tal problema es mínimo dada la propiedad de clausura bajo la unión de las clases de los lenguajes de contexto libre. Aunque el argumento es teóricamente correcto, esta restricción puede presentar problemas en la práctica. Si las gramáticas se construyen manualmente, basándose en el conocimiento a priori de las propiedades de las distintas clases, sería deseable poder modelizar cada una de ellas individualente. Por otra parte, si se construyen de forma automática mediante *Inferencia Gramatical (Fu 76)*, pueden aparecer problemas si se infiere una única gramática, al no poder tomar muestras de otras clases como muestras negativas de la gramática de la clase que se está infiriendo.

Este inconveniente puede ser eliminado usando la Extensión de las Funciones Discriminantes Lineales Generalizadas (EFDLG) propuesta aquí. Esta extensión hace posible la aplicación de las funciones discriminantes lineales cuando los elementos a clasificar tienen una representación distinta según la clase considerada. Es muy importante señalar que sin esta extensión, la aplicación de la TD a las GD cuando las gramáticas características son diferentes, no sería prácticamente realizable. La necesidad de esta extensión ha aparecido en problemas del Reconocimiento Automático del Habla (RAH), como el reconocimiento de palabras aisladas en ciertos diccionarios difíciles (Corell 87) (Casacuberta 88). Se puede señalar por tanto la gran importancia que la EFDLG tiene.

2. EXTENSIÓN DE LAS FUNCIONES DISCRIMINANTES LINEALES

En un sistema de RGF cada clase puede caracterizarse mediante una función del espacio de representación de los objetos en los reales, denominada *Función Discriminante (FD)*. La propiedad fundamental de una FD es que para una muestra dada, el valor de la función de la clase a la que pertenece es el máximo entre los valores de todas las FD para esa muestra.

En general se tiene el espacio de representación de los objetos, E, y C clases posibles $W_1, W_2, ..., W_c / W_i \subset E$

$$\text{con} \quad W_i \cap W_j = \phi, \ i \neq j, \qquad i, j = 1...C$$

Se define $\forall \boldsymbol{x} \varepsilon E$ las funciones discriminantes $D_i : E \rightarrow \Re$ asociadas a cada clase W_i, $i = 1...C$ de tal forma que:

$$\boldsymbol{x}\, \varepsilon W_i \longleftrightarrow D_i(\boldsymbol{x}) > D_j(\boldsymbol{x})$$
$$\text{con } i \neq j, \qquad i, j = 1..C$$

eligiéndose arbitrariamente en caso de igualdad.

Un tipo muy utilizado de FD, es la Función Discriminante Lineal Generalizada (FDLG) que tiene la expresión:

$$D(\boldsymbol{x}) = \sum_{i=1}^{N'} a_i y_i(x) = \boldsymbol{a}^t \boldsymbol{y}(\boldsymbol{x})$$

donde $\boldsymbol{y} : E_N \rightarrow E'_N$ y, $\boldsymbol{a}$ es un vector de pesos de dimensión N' (Duda 73).

Con esta formulación, la función discriminante no es lineal con $\boldsymbol{x}$ pero si con $\boldsymbol{y}$. Es importante notar que la representación vectorial $\boldsymbol{y} \longleftarrow \boldsymbol{x} \longleftarrow$ es única.

Existen diversos métodos iterativos para la obtención o aprendizaje de los vectores de pesos. El método general del *descenso por gradiente* consiste en definir alguna función escalar $J(\boldsymbol{a}_i)$ donde $i = 1...C$, que sea mínima para los vectores de peso solución $\boldsymbol{a}_i$, $i = 1...C$.

Una definición usual de $J(\boldsymbol{a}_i)$ consiste en el llamado Criterio Perceptrón, que está directamente relacionado con el error empírico de clasificación de las muestras de aprendizaje con respecto al conjunto de vectores $\boldsymbol{a}_i$ $i = 1...C$. En este caso el algoritmo de descenso por gradiente resulta, (Duda 73):

$$\boldsymbol{a}_i^1 \qquad \text{arbitrario}$$
$$\boldsymbol{a}_i^{k+1} := \boldsymbol{a}_i^k + \sigma(k) \sum_{\boldsymbol{y} \varepsilon Y_k} \boldsymbol{y} \qquad\qquad i = 1...C$$

donde Y_k es el conjunto de muestras mal clasificadas en el paso k y σ un factor de escala real que intensifica o decrementa las correcciones y que depende de k. En el trabajo presente $\sigma = 1$.

Para un conjunto de muestras linealmente separable, este algoritmo obtiene los pesos separadores en un número finito de pasos (Duda73).

Los problemas reales son normalmente no separables por la presencia de ruidos o distorsiones en el proceso de representación de las muestras. En estos casos se busca un conjunto de pesos "óptimo", es decir, pesos que clasifiquen correctamente el mayor número posible de muestras.

Existen varios métodos clásicos para casos no separables (Duda 73). Recientemente se ha introducido un nuevo método conocido como algoritmo "POCKET"

(Gallant 86) que genera secuencias de conjuntos de pesos, tanto para casos separables como no separables, con la propiedad de que cada conjunto de la secuencia es mejor que el anterior. Esto permite finalizar la iteración sin preocuparse por la posibilidad de haberse quedado con un conjunto particularmente desfavorable.

El algoritmo "POCKET" es una modificación del algoritmo de minimización del Criterio Perceptrón. Deriva su nombre de que se van guardando en el "bolsillo" los vectores de pesos que han clasificado correctamente un mayor número de muestras de aprendizaje consecutivas.

Las $\boldsymbol{y}_1, ..., \boldsymbol{y}_n$ muestras de aprendizaje se convierten en "infinitas" muestras $\boldsymbol{y}_1, ..., \boldsymbol{y}_n, \boldsymbol{y}_{n+1}, ...$ haciendo $\boldsymbol{y}_{n+1} = \boldsymbol{y}_1$, $\boldsymbol{y}_{n+2} = \boldsymbol{y}_2$ y así sucesivamente.

El algoritmo generalizado para C clases se presenta en la figura 1.

ALGORITMO POCKET

Datos, variables y resultados

V ≡ conjunto de muestras de aprendizaje.

$|V|$ ≡ número de muestras de aprendizaje.

C ≡ número de clases.

Para cada clase j se obtiene

$\boldsymbol{a}_j$ el vector de pesos

$\boldsymbol{b}_j$ el vector de pesos en el bolsillo

ca_j, cb_j contadores del número de reconocimientos correctos que han realizado sucesivamente el vector $\boldsymbol{a}_j$ y el $\boldsymbol{b}_j$ respectivamente.

Método

Para $j = 1...C$ hacer

inicializar $\boldsymbol{a}_j$; inicializar $\boldsymbol{b}_j$; $ca_j = cb_j := 0$;

fin hacer

$k := 0$;error=verdadero;quedanmuestras:=verdadero;

mientras error o quedanmuestras

$k := k + 1$; $(\boldsymbol{y}, i)$ =<muestra aprendizaje, clase >;

$d_i := \boldsymbol{a}_i^t \boldsymbol{y}$; error:=falso;

Para $j = 1...C$ hacer

si $d_i \leq \boldsymbol{a}_j^t y \quad j \neq i$ entonces

$\boldsymbol{a}_j := \boldsymbol{a}_j - \sigma\, \boldsymbol{y}$; $ca_j := 0$; error:=verdadero;

fin si

sino entonces

$ca_j := ca_j + 1$;

si $ca_j > cb_j$ entonces

$\boldsymbol{b}_j := \boldsymbol{a}_j$; $cb_j := ca_j$;

fin si

fin sino

fin hacer

si error entonces $\boldsymbol{a}_i := \boldsymbol{a}_i + \sigma\, \boldsymbol{y}$; fin si;

si k mod $|V| = 0$ entonces quedanmuestras:=falso

sino quedanmuestras:=verdadero;

fin mientras;

FIGURA 1.

Como se puede ver, si las clases no son linealmente separables, en el "bolsillo" se tiene en cada instante los vectores de pesos que mejor se han comportado hasta ese momento, de forma que se puede concluir la iteración cuando se observe que el número de errores se estabiliza y no disminuye. Si el conjunto de muestras de aprendizaje es linealmente separable, se puede asegurar la convergencia del algoritmo, es decir, en un momento determinado el número de muestras mal clasificadas será cero. Aunque el número de muestras es "cíclicamente infinito", se puede saber cuando se ha dado una vuelta a todas las muestras realmente diferentes sin haber cometido error, en cuyo caso se detendría la iteración.

Siguiendo (Vidal, 89) se introduce una *extensión* de las FDLG anteriores de la siguiente manera:

$$D_i(\boldsymbol{x}) = \sum_{j=1}^{d_i} \boldsymbol{a}_{ij}\boldsymbol{y}_{ij}(\boldsymbol{x}) \;=\; a_i^t y_i(\boldsymbol{x})$$
$$i = 1...C$$

donde ahora un objeto $\boldsymbol{x}$ perteneciente a cierto conjunto E, (no necesariamente estructurado como espacio vectorial) tendrá una representación vectorial $\boldsymbol{y}_j(\boldsymbol{x})$ que será distinta en cada clase j, es decir, $\boldsymbol{y}_j$ es una función de E en un cierto espacio vectorial E_j de dimensión d_j.

Esta extensión permite trabajar con representaciones vectoriales heterogéneas de objetos dados.

El criterio de decisión será idéntico al anteriormente citado para las FD:

"clasificar $\boldsymbol{x}$ en la clase W_i *sii*
$D_i(\boldsymbol{x}) > D_j(\boldsymbol{x})\;\; i,j = 1...C\,,\;\; i \neq j$ "

donde $D_i(\boldsymbol{x}) = \boldsymbol{a}_i^t\boldsymbol{y}_i(\boldsymbol{x}) \quad i = 1,...,C$

La extensión de los métodos de aprendizaje de pesos correspondiente a la extensión propuesta es bastante directa (Vidal 89). En particular, para el algoritmo de la figura 1, basta sustituir $\boldsymbol{y}$ por $\boldsymbol{y}_i$, donde i indica la clase en la que se representa el objeto $\boldsymbol{x}$.

Las funciones $\boldsymbol{y}_i$ tienen una expresión que depende del caso en concreto de que se trate. En este trabajo los objetos son frases de un lenguaje generado por una gramática. Se tienen varias clases, es decir varias gramáticas diferentes. Ciertas frases pueden pertenecer a más de un lenguaje. La representación vectorial de una frase en una clase determinada es un vector cuya dimensión es el número de reglas de la gramática y cuyas componentes son el número de veces que se ha utilizado cada regla en la generación de la frase. Si una frase no se puede generar por una gramática (no pertenece a esa clase) la representación es el vector nulo.

3. GRAMÁTICAS DISCRIMINANTES

Una gramática es una cuádrupla $G = (N, \Sigma, P, S)$ (Hopcroft 82) donde N y Σ son los conjuntos de símbolos no terminales y terminales respectivamente; P es un conjunto finito de reglas o producciones denotadas por $\alpha \rightarrow \beta$, donde α y β son elementos de $(N \ U \ \Sigma)^*$, y α contiene al menos un símbolo de N; y $S \ \varepsilon \ N$ es el símbolo inicial.

Adoptaremos los siguientes convenios de notación:

1.- Si Σ es un conjunto de símbolos, Σ^* representa el conjunto de todas las cadenas finitas que se pueden formar combinando los símbolos. Se incluye la cadena de longitud cero ó cadena nula λ. Se cumple que $\Sigma^+ = \Sigma^* - \{\lambda\}$.

2.- $n(x)$ *o* $|x|$ es la longitud de la cadena x, o número de símbolos en la cadena x.

3.- $\Gamma \Longrightarrow \Omega$, se dice que una cadena Γ genera directamente o deriva otra cadena Ω, si $\Gamma = \delta_1 \alpha \delta_2$, $\Omega = \delta_1 \beta \delta_2$ y $\alpha \rightarrow \beta$ es una regla de P.

4.- $\Gamma \overset{*}{\Longrightarrow} \Omega$, se dice que una cadena Ω genera o deriva otra cadena Ω, si existe una secuencia de cadenas $\sigma_1, \sigma_2, ., \sigma_n$ tal que $\Gamma = \sigma_1$, $\Omega = \sigma_n$, $\sigma_i \Longrightarrow \sigma_{i+1}$, $i = 1, 2, ..., n-1$. La secuencia de cadenas $\sigma_1, \sigma_2, ..., \sigma_n$ se denomina derivación de Ω desde Γ.

Se define el lenguaje generado por una gramática G como

$$L(G) = \{x \ / \ x \varepsilon \Sigma^*, \ S \overset{*}{\Longrightarrow} x\}$$

Una gramática G es ambigua si existe alguna cadena $x \varepsilon L(G)$ que tenga más de una derivación.

Existe una clasificación de las gramáticas en cuatro tipos según la forma de las producciones. Las gramáticas de tipo 2 o de contexto libre (GCL) tienen las producciones de la forma $A \rightarrow \beta$ donde $A \varepsilon N$ y $\beta \varepsilon (NU\Sigma)^+$.

Una Gramática Discriminante (GD) (Filiipski 80) es un par (G, f), donde G es una GCL no ambigua y f es una función $f : P \rightarrow \Re$, siendo $\Re$ el conjunto de los números reales. Dado $x \varepsilon L(G)$, denotamos por $D(x)$ la derivación canónica a izquierdas de x desde S, o sea $D(x) = (r_1, ..., r_{n(x)})$, donde $r_i \varepsilon P$ para $1 \leq i \leq n(x)$, $n(x)$ es la longitud de la derivación $D(x)$, y $D(x) : S \overset{*}{\Longrightarrow} x$.

Como G es no ambigua, D es única. Se define una función $\Phi : \Sigma^* \rightarrow \Re$ como

para todo $x \varepsilon \Sigma^*$

$$\Phi(x) = \begin{cases} \sum_{i=1}^{n(x)} f(r_i) & \text{si} \quad x \varepsilon L(G) \ y \ r_i \ \varepsilon D(x) \quad 1 \leq i \leq n(x) \\ 0 & \text{si} \quad x \varepsilon L(G) \end{cases}$$

Dado un número real Θ, $L(G)$ puede ser dividido en tres lenguajes

$$\begin{aligned} L_+(G,f,\Theta) &= \{\, x \,/\, x\varepsilon L(G) \;\; y \;\; \Phi(x) > \Theta \,\} \\ L_0(G,f,\Theta) &= \{\, x \,/\, x\varepsilon L(G) \;\; y \;\; \Phi(x) = \Theta \,\} \\ L_-(G,f,\Theta) &= \{\, x \,/\, x\varepsilon L(G) \;\; y \;\; \Phi(x) < \Theta \,\} \end{aligned}$$

Siguiendo (Filipski 80), la aplicación de las GD al RF se puede formalizar como sigue. Sean C clases y una única gramática característica G de contexto libre que sirve de modelo estructural único a todas las clases. Cada clase se caracteriza por una función f_i: $1 \leq i \leq C$ para cada gramática discriminante (G, f_i) y por un número real Θ_i: $1 \leq i \leq C$. Una muestra $x\varepsilon\Sigma^*$ se clasifica en la clase i si y sólo si

$$\Phi_i(x) - \Theta_i > \Phi_j(x) - \Theta_j \text{ para todo } i,j \; 1 \leq i,j \leq C \quad i \neq j$$

Dada G se define una función vectorial $\boldsymbol{I} : \Sigma^* \rightarrow \boldsymbol{N}^{|P|+1}$ donde $\boldsymbol{N}$ es el conjunto de números naturales y $|P|$ el número de reglas de la gramática G. Dada una muestra $x\varepsilon\Sigma^*$, se denota por $I_j(x)$ la j-ésima componente del vector $\boldsymbol{I}(x)$, que es definida como el número de veces que la regla r_j $1 \leq j \leq |P|$ se ha usado para obtener x desde S, y la componente $(|P|+1)$–ésima se hace igual a 1. $\boldsymbol{I}(x)$ se denomina **índice estructural** de x. Esta función define un cambio en el espacio de representación, que va desde Σ^* al espacio vectorial $\boldsymbol{N}^{|P|+1}$, que permite disponer de todas las herramientas del RF basadas en la TD. En particular, dada $\boldsymbol{I}(x)$ se puede escribir $D_i(x)$ como una FDL.

$$D_i(x) = \sum_{j=1}^{|P|} I_j(x) f_i(r_j) - \Theta_i = \boldsymbol{f}_i^t \boldsymbol{I}(x)$$

donde $\boldsymbol{f}_i$ es un vector de dimensión $(|P|+1)$, cuyas $|P|$ primeras componentes son los pesos definidos en (G, f_i), y la última componente es Θ_i.

Para permitir el uso de C **diferentes** gramáticas características, se genealiza lo anterior y se obtienen C gramáticas discriminantes $(G_i, \boldsymbol{f}_i)$ $1 \leq i \leq C$. Se pueden ahora definir C FDLG extendidas como:

$$D_i(x) = \sum_{j=1}^{|P_i|} I_{ij}(x) \boldsymbol{f}_i\,(r_j) - \Theta_i = f_i^t I(x)$$

donde ahora I_i: $1 \leq i \leq C$ es una función vectorial desde Σ^* a $\boldsymbol{N}^{|P_i|+1}$, cuya $j-$ ésima componente es el número de veces que la regla r_j de G_i se ha usado para reconocer x en la gramática G_i, y $\boldsymbol{f}_i$ es idéntica a la anterior.

Con esta generalización, x se clasifica en la clase i si y sólo si

$$f_i^t I_i(x) > f_j^t I_j(x) \qquad 1 \leq i,j \leq C \qquad i \neq j$$

Se puede ver claramene que los métodos descritos en el apartado 2 se pueden utilizar para aprender los C vectores f_i dadas las C gramáticas de contexto libre G_i.

4. EXPERIMENTOS

Uno de los usos más interesantes de las GD es la modelización de lenguajes co n distribuciones de longitud de las (sub)cadenas, lenguajes para los que la aproximación por gramáticas estocásticas no es buena (Filipski 80).

En particular, las gramáticas estocásticas modelizan implícitamente distribuciones geométricas, y en consecuencia presenta resultados peores cuando las distribuciones son, por ejemplo, uniformes, normales, o de Poisson (Wetherell 80).

A continuación se presentan experimentos para comparar la eficacia de la aproximación por gramáticas estocásticas con la ofrecida por la aproximación de GD. Asimismo se han utilizado clases correspondientes a gramáticas características **diferentes** para probar la extensión propuesta de las FDLG. También se presentan resultados sobre la evolución del comportamiento de la generalización del algoritmo "POCKET" en función de la talla del conjunto de muestras de aprendizaje.

Se parte del lenguaje generado por una gramática estocástica. Se implementa un autómata de estados finitos que reconozca el mismo lenguaje. Los símbolos de entrada son los terminales de la gramática. A cada regla de la gramática se le asocia una transición del autómata.

Para cada autómata se generan 2 conjuntos de muestras diferentes: el de aprendizaje y el de test.

El primer paso al construir una muestra es generar aleatoriamente la longitud que va a tener, dentro del rango longitud media $\pm$ desviación, con probabilidad uniforme para todos los valores del rango. De esta manera se consigue que las muestras tengan una distribución de longitud determinada, que tal como se ha mencionado constituye un caso para el que las gramáticas estocásticas no se comportan bien.

A continuación, y partiendo del estado inicial del autómata correspondiente, se generan los símbolos de acuerdo con la probabilidad de cada transición. El recorrido por el autómata se para cuando hemos llegado a la longitud predeterminada. Si no es posible llegar a esa longitud se reintenta la generación de otra frase.

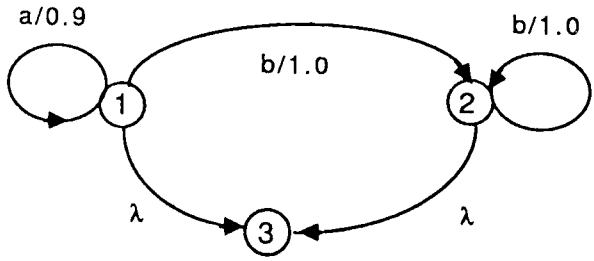
Paralelamente a la generación de la frase se va construyendo su representación vectorial en el espacio de índices estructurales. Para esto se tiene numerada cada regla (o transición del autómata en definitiva). Cada vez que se utiliza la regla i–ésima en la generación de la frase, se incrementa en uno la componente i–ésima del vector de índices estructurales.

Cuando la frase está generada y se tiene su representación vectorial, se la reconoce en cada uno de los $C-1$ autómatas distintos al que la ha generado. De esta manera se tiene la representación vectorial de la frase en las C clases. La forma de construirlas es exactamente igual que en la generación: se recorre al autómata representativo de la clase, y cada vez que se utilice una transición, se incrementa en uno la componente correspondiente del vector. Si no es reconocida una frase en una clase, su representación vectorial asociada será el vector nulo. Cabe señalar que el recorrido de reconocimiento no presenta ambigüedades, ya que las gramáticas son no ambiguas por hipótesis. También hay que observar la necesidad de la EFDLG, pues una frase determinada puede pertenecer a más de una gramática, y éstas sin embargo no tienen en general el mismo número de reglas, ni tampoco la regla i–ésima de una gramática tiene porqué corresponderse con la regla i–ésima de otra. A continuación se realiza el aprendizaje de los vectores de pesos con el algoritmo "POCKET" sobre el conjunto de muestras de aprendizaje y se clasifica el conjunto de muestras de test.

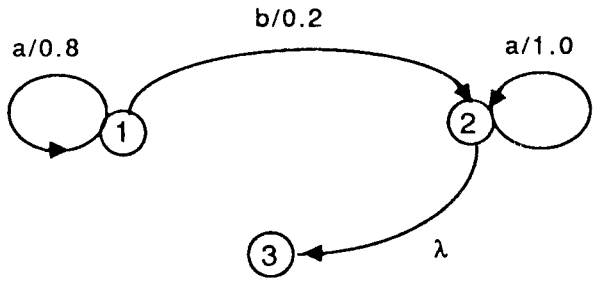
Por otro lado, y para establecer la comparación con el método propuesto, sobre el conjunto de muestras de aprendizaje, y con la gramática característica de cada una de las clases, se realiza el aprendizaje clásico (González 78, Fu 82) de las probabilidades de cada regla para definir las gramáticas estocásticas. Estas probabilidades se calculan como el cociente del número total de veces que se ha utilizado una transición determinada en la generación de todas las muestras de aprendizaje de esa clase, entre el número total de veces que se ha utilizado alguna transición con el mismo estado de partida que el de la que estamos calculando. De esta forma se tiene que el sumatorio de todas las probabilidades de las reglas o transiciones que parten de un mismo estado, es igual a uno. Después se clasifica el conjunto de muestras de test según el Criterio de Bayes, consistente en que si una frase es reconocida por más de una gramática, se dice que pertenece a aquella que la ha generado con máxima verosimilitud, siendo la verosimilitud el productorio de las probabilidades de las transiciones utilizadas en el reconocimiento de la frase.

Se han utilizado los lenguajes definidos por los siguientes autómatas:

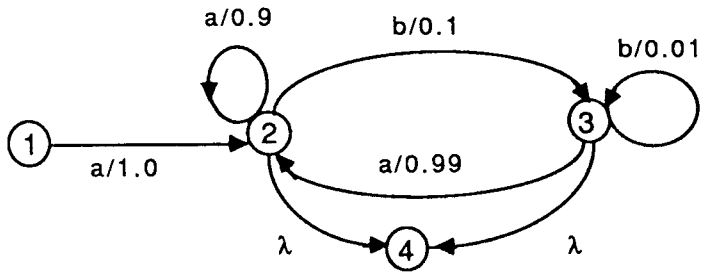
$L(GD_1) : a^*b^*$



$L(GD_2) : a^*ba^*$



$L(GD_3) : (ab^*)^+$



Se han realizado 2 experimentos. En ambos casos el conjunto de muestras de test es idéntico. La longitud media de las frases es de 20 símbolos y la máxima desviación 4, es decir, tenemos muestras de longitud mayor que l6 y menor o igual que 24 con igual probabilidad para cualquier valor del rango. La siguiente tabla expresa los parámetros del conjunto de muestras de test.

TABLA I

Características del conjunto de un total de 1500 muestras de TEST

	N. muestras	Long. media	Desviación
$L(GD1)$	420	20	4
$L(GD2)$	567	20	4
$L(GD3)$	513	20	4

EXPERIMENTO 1

Estudio de la tasa de error en la clasificación en función del número de veces que se recorre el conjunto de muestras de aprendizaje en el algoritmo "POCKET". Obviamente esta prueba solo se realiza con el método de EFDLG. El conjunto de muestras de test es el indicado anteriormente.

TABLA 2

Tasa de error (%) en función del número de iteraciones (1..15) realizadas y de la talla del conjunto (15..270)

	15	30	90	270
1	72.0	72.0	34.2	34.2
2	34.2	34.2	2.3	2.5
3	2.9	3.9	--	2.5
4	--	--		3.9
5				3.7
6				3.7
7				2.1
9				1.5
15				0.5

Nota: Los guiones '--' indican que esa medida no tiene sentido realizarla, pues el algoritmo "POCKET" encuentra las 3 clases linealmente separables en la iteración anterior.

EXPERIMENTO 2

Estudio de la tasa de error de clasificación sobre las 1500 muestras de test realizando el aprendizaje con un número de muestras cada vez mayor. Ambos métodos utilizan el mismo conjunto de muestras de aprendizaje. El número de muestras de aprendizaje de la siguiente tabla expresa el total de muestras. La longitud y la desviación media son idénticas a las de las muestras de test.

TABLA III

Tasas de error (%) comparativas entre los distintos métodos de clasificación.

Muest.	15	30	60	150	300	600	1500
BAYES	8.2	10.5	10.5	13.0	2.0	1.5	2.0
EFDLG	4.7	8.2	0.3	0.8	0.5	0.3	0.1

Hay que observar el comportamiento notablemente superior de la EFDLG sobre el método de Bayes.

5. CONCLUSIONES

Del trabajo y resultados presentados, se desprenden las conclusiones siguientes:

1.- Se puede admitir que cuando los lenguajes presenten distribuciones uniformes de longitud de (sub)cadenas (incluso con otras geométricas) la aproximación por GD es preferible a la de gramáticas estocásticas.

2.- La talla del conjunto de muestras de aprendizaje para el método propuesto debe ser suficientemente grande, pues sobre conjuntos pequeños la tasa de error presenta un comportamiento oscilante que no indica exactamente la tasa óptima que se podría conseguir.

3.- El número de iteraciones a realizar con el algoritmo "POCKET" para una clasificación óptima crece conforme crece la talla del conjunto de muestras de aprendizaje. Cuando esta talla es grande la tasa de error presenta un comportamiento oscilante con pocas iteraciones. A partir de un cierto número de iteraciones, no excesivamente grande por otro lado, la tasa de error tiene valores muy próximos al óptimo y no tiene sentido alargar el aprendizaje.

6. AGRADECIMIENTOS

Queremos hacer constar nuestro agradecimiento al revisor anónimo por sus comentarios sobre el primer borrador de este trabajo.

7. BIBLIOGRAFÍA

[1] **Casacuberta, F.; Vidal, E.** (1987). "Reconocimiento Automático del Habla". Marcombo.

[2] **Casacuberta, F.; Vidal, E.** (1988). "Speech Recognition with Difficult Dictionaries". En "Recent Advances in Speech Understanding and Dialog Systems". H. Nieman (ed) Springer-Verlag.

[3] **Corell, F.J.; Tabarés, J.C.; Vidal, E.; Casacuberta, F.** (1987). "Utilización de Funciones Discriminantes Lineales Generalizadas en el Reconocimiento de Palabras Aisladas con Ciertos Diccionarios Difíciles". Qüestiió Vol. 11. n° 1. pp. 17-35.

[4] **Duda, R.O.; Hart, P.E.** (1973). "Pattern Classification and Scene Analysis". J. Wiley & Sons.

[5] **Filipski, A.J.** (1980). "A Least Mean-Squared Error Approach to Syntactic Classification". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-2, n° 3. pp. 252-255.

[6] **Fu, K.S.; Booth, T.L.** (1975). "Grammatical Inference, Introduction and Survey" partes 1 y 2. IEEE Trans. Syst. Man and Cybern. Vol. SMC-5. pp. 95-111, pp. 409-423.

[7] **Fu, K.S.** (1982). "Syntactic Pattern Recognition and Applications". Prentice-Hall, 1982.

[8] **Gallant, S.I.** (1986). "Optimal Linear Discriminant". Proc. of the 8th Int. Conf. on Pattern Recognition. pp. 849-852.

[9] **González, R.C.; Thomason, M.G.** (1978). "Syntactic Patern Recognition. An Introduction". Addison-Wesley Publishing Company.

[10] **Hopcroft, J.E.; Ullman, J.D.** (1979). "Introduction to automata theory, languages and computation". Addison-Wesley.

[11] **Vidal, E.; Casacuberta, F.** (1989). "An Hybrid Framework combining Structural and Decision Theoretic Pattern Recognition and Applications". Int. Journal of Patt. Recognition and Artificial Intelligence. Vol. 3, N°.2 (1989), pp. 181-206.

[12] **Wetherell, C.S.** (1980). "Probabilistic Languages: A Review and Some Questions". Computing Surveys. Vol. 12(4), pp. 361-379.