NUEVOS CRITERIOS PARA LA APROXIMACION EN UNA RECIENTE VERSION CON PREPROCESAMIENTO LINEAL DEL ALGORITMO AESA

Luisa Micó y José Oncina
Departamento de Tecnología Informática y Computación, Universidad de Alicante

Abstract

The LAESA algorithm (Lineal Approximating Eliminating Search Algorithm) was introduced for reducing the large space complexity of the AESA. This is achieved using only a very small set of points, called Base Prototypes, included in the complete set of points. Moreover, as the AESA, this algorithm can be used for finding Nearest Neighbours in Metric Spaces with an average constant number of distance computations. The algorithm make use of two generic functions, CONDICION and SELECCION, to allow different heuristic strategies of Base Prototypes management. In this paper, we present a new strategy for the SELECCION function that improve the previous results. Also, we show that with this new criterion, the number of distance computations does not increase with the dimensionality, but with the average distance from the test sample to his nearest neighbour.

INTRODUCCION.

La necesidad de determinar el vecino más próximo a un elemento dado (muestra) en un conjunto de elementos (datos), aparece en muchas aplicaciones en Reconocimiento de Formas. Dentro del conjunto de técnicas aparecidas para resolver este problema [1], particularmente interesantes son aquellas que solo hacen uso de las propiedades métricas del espacio. Esto es así debido a la gran cantidad de problemas prácticos para los cuales no se dispone de las coordenadas de los datos y para los que además el cómputo de la distancia es costoso, como es el caso del Reconocimiento de Palabras Aisladas [2].

El algoritmo conocido como AESA (Eliminating Approximating Search Algorithm), introducido por Vidal en 1986, reduce el número de distancias calculadas respecto a otros métodos anteriormente propuestos, siendo además interesante por dos motivos: 1) solo hace uso de las propiedades métricas del espacio, y 2) encuentra el vecino más próximo calculando un número de distancias (aproximadamente) constante con la talla del conjunto de prototipos[3]. Este algoritmo consta de dos fases que se repiten sucesivamente: en la primera se selecciona un prototipo que, heurísticamente, esté lo más cerca posible de la muestra y se calcula la distancia a ésta. En la segunda fase se eliminan todos aquellos prototipos que ya no pueden ser el más cercano como resultado de aplicar una regla de eliminación basada en la desigualdad triangular. Posteriormente, en [4] se introduce una nueva formulación del algoritmo basada en la estrategia de Ramificacion y Poda. También se introduce una función G que es utilizada en ambas fases

del algoritmo, simplificándolo y haciendo que disminuya el número de distancias a calcular. El principal inconveniente de este algoritmo es su complejidad espacial que crece con el cuadrado del número de prototipos, ya que requiere conocer en ejecución la distancia entre cualquier par de ellos. Esto limita el uso del AESA a conjuntos de prototipos relativamente pequeños [2].

Recientemente han aparecido nuevas técnicas con el objetivo de reducir esta complejidad espacial manteniendo el comportamiento del AESA frente al número de distancias calculadas. Ramasubramanian y Paliwal [5][6] proponen varias técnicas para resolver este problema en espacios vectoriales. Paralelamente, una nueva versión del algoritmo AESA, llamada Lineal-AESA, fue introducida con el mismo fin, pero conservando la característica del AESA de utilizar únicamente las propiedades métricas del espacio.

ALGORITMO LAESA.

El proceso de búsqueda en el algoritmo LAESA consta, como en el AESA, de las fases de Aproximación y Eliminación. La principal diferencia entre ambos, es la utilización en el LAESA de una matriz de distancias, calculada en el preproceso, entre el conjunto completo de prototipos (de talla n) y un subconjunto del mismo (de talla m<<n) y al cual denominamos *conjunto de Prototipos Base*[7][8][9]. La eficiencia del algoritmo va a depender tanto del tamaño de este conjunto como de la selección que se ha hecho de los elementos que lo componen. Para la selección de estos Prototipos Base, se ha utilizado una estrategia voraz (con un coste lineal) consistente en seleccionar incrementalmente los prototipos que se encuentran maximamente separados de entre los ya obtenidos. El estudio de las características de los prototipos candidatos para ser usados como referencia fue realizado anteriormente por Shapiro[10], obteniendo mejores resultados cuando los prototipos elegidos se encontraban lo más alejado posible del conjunto de datos. Ramasubramanian y Paliwal también utilizan una técnica semejante para la selección de los Prototipos Base con la diferencia respecto al utilizado con el LAESA de que los Prototipos Base seleccionados por ellos, no pertenecen al conjunto de datos, sino que son prototipos construidos artificialmente.

En el paso de aproximación, para la selección de nuevos candidatos a prototipo más cercano, escogemos cada vez dos elementos, uno base y otro no-base. Después, con la ayuda de una función llamada SELECCION, elegiremos uno de los dos elementos. La eliminación será controlada para los Prototipos Base por una función booleana, llamada CONDICION, que solo permitirá eliminarlos según el valor de la misma.

Algoritmo LAESA

```
Entrada:   P ⊂ E,    n = |P|;                    // conjunto finito de prototipos //
           B ⊆ P,    m = |B|;                    // conjunto de Prototipos Base //
           D ∈ ℜ^(n×m) ;                         // matriz precomputada de n×m distancias //
           x ∈ E;                                // muestra//
Salida:    p* ∈ P; d* ∈ ℜ;                       // prototipo más cercano y su distancia a x //
Funciones: d: E × E ---> ℜ;                      // función distancia //
           CONDICION: Booleano;                  // controla la eliminación de Prototipos Base //
           SELECCION: B×(P-B) ---> P;            // selección de Prototipos Base o no-Base //
Variables: p, q, s, b ∈ P;
           G ∈ ℜ^n;                              // array para la cota inferior //
           dxs, gp, gq, gb ∈ ℜ;
           nc ∈ ℕ;                               // número de distancias calculadas //

empezar
    d*:= ∞; p*:= indeterminado; G:= [0]; s:= elemento_arbitrario(B); nc := 0;
    mientras |P| > 0 hacer
        dxs:= d(x,s); P:= P-{s}; nc:= nc+1;                  // cálculo de la distancia //
        si dxs < d* entonces p*:= s; d*:= dxs; fsi            // actualización de p* y d* //
        q:= indeterminado; gq:= ∞; b:= indeterminado; gb:= ∞;
        para todo p ∈ P hacer                  // bucle de eliminación y aproximación //
            si s ∈ B entonces                  // actualizar G, si es posible //
                G[p]:= max( G[p], | D[p,s]-dxs | )
            fsi
            gp:= G[p];
            si p ∈ B entonces
                si ( gp ≥ d* & CONDICION ) entonces P:= P-{p} // eliminar en B //
                sino                                  // aproximación: seleccionar en B //
                    si gp<gb entonces gb:= gp; b:= p fsi
                fsi
            sino
                si gp ≥ d* entonces P:= P-{p}                 // eliminar en P-B //
                sino                                  // aproximación: seleccionar en P-B //
                    si gp<gq entonces gq:= gp; q:= p fsi
                fsi
            fsi
        fpara
        s:= SELECCION(b, q);
    fmientras
fempezar
```

En [7][8] se han explorado diferentes elecciones para la función CONDICION. En la primera familia de condiciones propuestas, solo se permitía eliminar Prototipos Base cuando se había seleccionado previamente un número mínimo de los mismos en la fase de selección. En

otra condición propuesta, solamente se permite eliminar Prototipos Base cuando el último prototipo seleccionado no contribuyó a la fase de eliminación. De entre todas ellas, la más interesante era esta última en la cual a partir de un tamaño relativamente pequeño del conjunto de Prototipos Base, el aumento de este tamaño no afectaba al número de distancias calculadas. Este comportamiento lo podemos observar en la Figura 1 y corresponde al caso rc=∞.

Los resultados obtenidos con estas estrategias han sido bastante satisfactorios (el número de distancias requeridas en cualquier caso es menor de 1.5 veces las obtenidas con el algoritmo AESA). El número de distancias calculadas, como ocurre también con el algoritmo AESA, aumenta rápidamente con el curso de la dimensionalidad. Sin embargo, como ya indicó en su día Vidal [11], en el AESA, este aumento en el número de distancias no era consecuencia del incremento de la dimensionalidad, sino del aumento de las distancias medias entre las muestras y sus vecinos más próximos. Sin embargo, en el LAESA, con los criterios introducidos hasta el momento, este comportamiento no se observa.

NUEVOS CRITERIOS.

En este trabajo nos planteamos conseguir el comportamiento mencionado anteriormente, con la introducción de un nuevo criterio para la función SELECCION. El único criterio SELECCION introducido hasta el momento, no permite elegir prototipos no-base en la fase de selección mientras aún queden Prototipos Base. Por este motivo, el número de prototipos que se utilizan en la selección es independiente de la proximidad de las muestras a sus vecinos más próximos. Como podrá observarse más adelante, el hecho de que las muestras hayan sido elegidas de forma que estén muy cercanas a sus vecinos más próximos, no es suficiente para disminuir el rápido incremento con el curso de la dimensionalidad.

Para evitar este problema, proponemos un nuevo criterio de SELECCION con el cual se llega al mismo comportamiento observado en el AESA. La función SELECCION que introducimos aquí, permite según las condiciones que a continuación se especifican, seleccionar elementos no Base desde el primer momento, no teniendo que esperar a que el conjunto de Prototipos Base esté vacío para ello. Esto permitirá que, cuando las muestras estén muy cercanas a sus vecinos más próximos, se llegue rápidamente a encontrarlos debido a la mayor flexibilidad en la selección. Está claro que si supiésemos que el prototipo no-base seleccionado es muy cercano a la muestra, sería muy útil utilizarlo en la fase de eliminación, pues permitiría eliminar un gran número de prototipos. Es evidente que no debemos calcular en cada paso la distancia del prototipo no-base pues esto nos llevaría a incrementar el número de distancias

calculadas. Por ello utilizaremos la siguiente regla heurística: supondremos que el prototipo no-base es bastante cercano cuando este prototipo ha sido seleccionado como no-base durante un cierto número de iteraciones consecutivas. La función SELECCION queda de la siguiente forma:

$SELECCION_{rc}(b,q)$ = **si** en las *rc* iteraciones anteriores q es el mismo
entonces q
sino b
fsi para rc=1,2..∞

Este criterio no permite seleccionar dos veces seguidas prototipos que no son base. Esto es interesante porque cada vez que seleccionamos un prototipo que no es base, no estamos actualizando la cota inferior de la distancia al resto de prototipos (esto se debe a que en la matriz de distancias solo disponemos de las distancias de los Prototipos Base al resto de elementos, y en este caso concreto, solo podríamos actualizar la cota a los Prototipos Base).

EXPERIMENTOS Y RESULTADOS.

Para este nuevo criterio de selección solo se ha comprobado su eficiencia para siguiente criterio de condicion: (CONDICION= ningún prototipo fue eliminado en la iteración anterior), que es el más interesante ya que la decisión para el tamaño del conjunto de Prototipos Base no es crítico.

Se han realizado experimentos variando el parámetro rc del criterio $SELECCION_{rc}$. El caso rc=∞ correspondería al criterio de SELECCION presentado hasta el momento.

En la Figura 1 se puede observar una comparación de diferentes criterios de selección (rc = 3, 4, 6, 8, 10, ∞) que había sido introducido hasta el momento para una distribución uniforme sobre el hipercubo unidad con diferentes tamaños del conjunto de Prototipos Base. En todos los casos presentados, los resultados que se obtienen son algo mejores que los obtenidos para $SELECCION_{\infty}$(≈5%).

Para estudiar el comportamiento del algoritmo frente a muestras agrupadas, utilizaremos "intervalos de tolerancia" alrededor de cada prototipo de forma que las muestras se extraigan aleatoriamente en el interior de dichos intervalos. La tolerancia vendrá expresada como porcentajes respecto al hipercubo unidad.

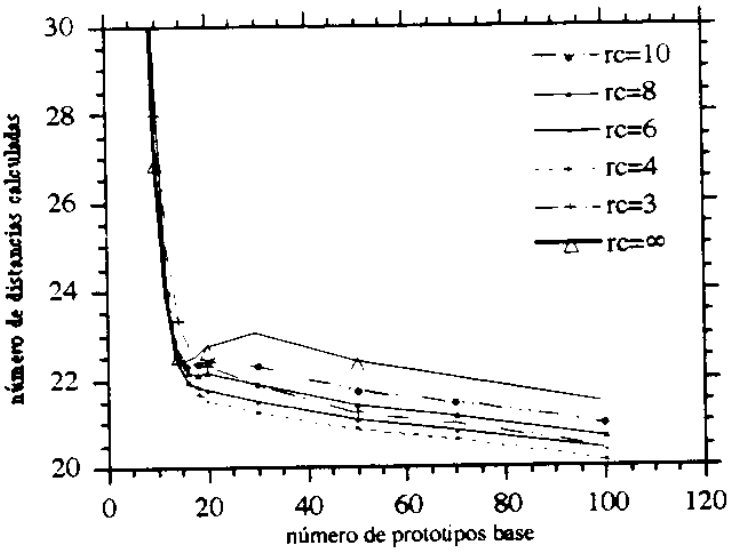
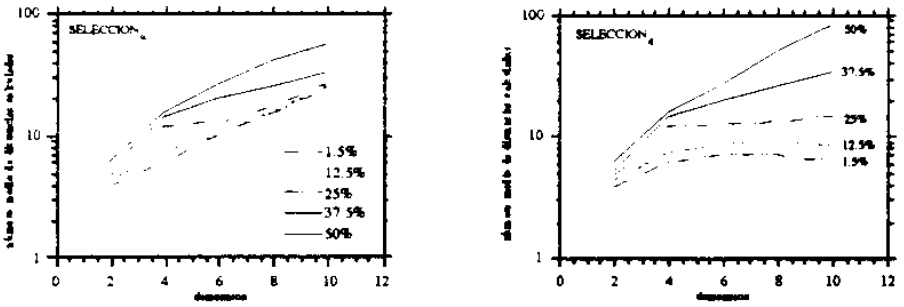


Figura1. Número medio de distancias calculadas en función del número de prototipos base para $SELECCION_{rc}$ usando la métrica euclídea sobre un conjunto de 1024 prototipos de dimensión 6.

La comparación de los dos criterios de selección para diferentes tolerancias, la podemos hacer observando las figuras 2 y 3. Los intervalos de tolerancias en cada caso, expresando en %, se refieren a que las muestras son extraídas aleatoriamente en el interior de dichos intervalos alrededor de cada prototipo del conjunto. Los experimentos realizados se han llevado a cabo utilizando para cada dimensión el número "óptimo" de Prototipos Base obtenidos en una versión anterior de CONDICION (CONDICION=falso) con el criterio $SELECCION_{\infty}$[7][8]. Estos valores son:

DIMENSION	2	4	6	8	10
Nº PROT. BASE	3	6	14	28	48



Figuras 2 y 3. Número medio de distancias para los dos criterios de elección variando la dimensión y la tolerancia (cercanía) de las muestras a sus vecinos más próximos.

Para tolerancias altas, ambos criterios tienen un comportamiento parecido, mientras que cuando las tolerancias son muy bajas (menores del 15%) (las muestras están muy cercanas a su vecino más próximo), la utilización del nuevo criterio de selección, hace que el número de distancias no aumente con el curso de la dimensionalidad, sino que incluso llega a disminuir.

CONCLUSIONES.

Los resultados presentados en el apartado anterior, muestran que el incremento en el número de distancias a calcular para encontrar el vecino más próximo no depende de la dimensión, disminuyendo incluso cuando las tolerancias son muy bajas (menores del 15%). Este ha sido el motivo de la introducción de este nuevo criterio de selección, pues para datos reales, el promedio de las distancias de las muestras a sus correspondientes prototipos, no aumenta al añadir más características a la representación de los objetos. De esta forma, por tanto, se puede decir que las prestaciones del algoritmo resultan prácticamente insensibles a la dimensionalidad del espacio.

BIBLIOGRAFIA.

[1] Dasarathy, "Nearest Neighbour(NN) norms: NN Pattern Classification Techniques", IEEE Computer Society Press, 1991.

[2] Vidal, Rulot, Casacuberta and Benedí, "On the use of a metric-space search algorithm (AESA) for fast DTW-bassed recognition of isolated words", IEEE Transactions on Acoustics, Speech, and Signal Processing., vol. 36, pp. 651-660, 1988.

[3] Vidal, "An algorithm for finding nearest neighbours in (approximately) constant average time complexity.", Pattern Recognition Letters, vol. 4, pp. 145-157, 1986.

[4] Vidal, "New formulation and improvements of the Nearest-Neigbour Approximating and Eliminating Search Algorithm (AESA). To appear in Pattern Recognition Letters.

[5] Ramasubramanian and Paliwal, "An Efficient Approximation-Elimination Algorithm for Fast Nearest-Neighbour Search Based on a Spherical Distance Coordinate Formulation.", Signal Processing V:Theories and Applications., pp. 1323-1326, 1990.

[6] Ramasubramanian and Paliwal, "Fast Algorithms for Nearest-Neighbour Search and Application to Vector Quantization", PhD dissertation. University of Bombay, 1991.

[7] Micó, Oncina and Vidal, "A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing-time and memory requirements". To be published.

[8] Micó, Oncina and Vidal, "An algorithm for finding nearest neighbours in constant average time with a linear space complexity". Int. Conference on Pattern Recognition (ICPR), Le Hague, September 1992.

[9] Micó, Oncina and Vidal, "Algoritmo para encontrar el vecino más próximo en un tiempo medio constante con una complejidad espacial lineal", Tech. Report DSIC II/14-91. Universidad Politécnica de Valencia, 1991.

[10] Shapiro, "The Choice of Reference Points in Best-Match File Searching.", Artificial Intelligence/Language Processing., vol. 20, pp. 339-343, 1977.

[11] Vidal, "Diversas aportaciones al Reconocimiento Automático del Habla", Tesis Doctoral. Fac. Físicas. Universidad de Valencia, 1985.