

EVALUACIÓN DE LA PERFORMANCE DE ESTUDIANTES MEDIANTE TÉCNICAS DE CLUSTERING

Dr. Ing. Jesús Rubén Azor Montoya (jesus.azor@um.edu.ar)
Facultad de Ingeniería - Universidad de Mendoza

CV: Profesor Titular en las asignaturas Estadística Aplicada II y Análisis de Señales en la Facultad de Ingeniería de la Universidad de Mendoza. Ex Investigador en el Instituto de Automática Industrial en Madrid (España). Miembro del Comité del Doctorado en Ingeniería de la Universidad de Mendoza. Expositor en distintos Congresos en EEUU, Puerto Rico, Australia, Sudáfrica, Cuba, etc..

Abstract

La aplicación de la informática en la administración del sistema universitario hace que se cuente con una inmensa cantidad de datos que, racionalmente tratados, pueden ser de ayuda significativa en el seguimiento de los aprendizajes de los estudiantes. Mediante técnicas de *clustering*, se propone desarrollar una herramienta útil en la estrategia de gestión que permita evaluar la progresión del comportamiento estudiantil. Esto es, dividiendo a los estudiantes en grupos homogéneos de acuerdo a sus características y habilidades. Estas aplicaciones pueden ayudar tanto al instructor como al estudiante a mejorar la calidad de la educación. La metodología seguida es la del análisis de agrupación de datos mediante lógica difusa utilizando el algoritmo *Fuzzy C-means* para, a través de un sistema experto, lograr un indicador estándar (llamado *Grade*) que permita la segmentación. En línea con lo expresado, se propone un segmento de programa de *MATLAB*[®] (MathWorks) que permite a partir de los datos crudos obtener los indicadores mencionados que permiten el agrupamiento (*clustering*).

Palabras clave: Clustering, algoritmo Fuzzy C-means, lógica difusa, sistema experto.

1 – Introducción

La teoría de Zadeh sobre conjuntos difusos aparecida en 1965, junto a las técnicas de *clustering* (agrupamiento), [1] han llevado a muchos investigadores a aplicarlas en educación [2, 3].

A partir del crecimiento explosivo de datos educativos, las instituciones tienen la oportunidad de utilizar estos datos para mejorar la calidad en las decisiones de gestión. La evaluación de los logros de aprendizaje de los estudiantes es el proceso de determinar los niveles de performance de cada estudiante en relación a los objetivos educativos.

Un sistema de evaluación de alta calidad certifica y provee las bases para la mejora individual y asegura que todos los estudiantes reciban la clasificación justa a fin de no limitar las oportunidades presentes y futuras de los mismos.

Por lo tanto, el sistema de evaluación necesita transparencia, objetividad, razonamiento lógico y fácil aplicación informática.

La evaluación de la performance es una de las bases para monitorear la progresión del aprendizaje de los estudiantes en una institución. Es por ello que el *clustering* en

diferentes categorías en función de su performance, se ha convertido en una tarea complicada.

Con las técnicas tradicionales de agrupamiento en función de promedios, es difícil obtener una visión global del estado del comportamiento estudiantil y al mismo tiempo descubrir detalles importantes en el desempeño a través del tiempo.

Con la ayuda de métodos de minería de datos, tales como *algoritmos de clustering*, es posible descubrir características claves de la performance de los estudiantes y, posiblemente, utilizarlas en predicciones a futuro.

Por ejemplo, considérese un escenario donde las notas de dos estudiantes son 50, 60, 70 y

70, 60, 50 en tres pruebas, respectivamente. La nota promedio obtenida por cada uno es 60. ¿Se puede concluir, a partir de la media, que el nivel de inteligencia de ambos estudiantes es el mismo?

Por supuesto que no. Los datos indican que un estudiante mejora, mientras que el otro se está deteriorando constantemente. Es decir, uno de ellos está aprendiendo constantemente de su experiencia.

Este ejemplo ilustra que la clasificación del estudiante o el método de modelado de la evaluación de la performance académica, debe basarse en la homogeneidad de clase.

La clasificación (*Grading*) en la educación es el proceso de aplicar medidas estandarizadas de diferentes niveles para evaluar el desempeño en un curso. El *Grading* puede ser utilizado por posibles empleadores o instituciones educativas para evaluar y comparar a los solicitantes.

La agrupación de datos es un proceso de extracción de patrones previamente desconocidos, válidos, posicionalmente útiles y ocultos de grandes conjuntos de datos.

El principal objetivo de la agrupación (*clustering*) es dividir a los estudiantes en grupos homogéneos de acuerdo a sus características y habilidades. Estas aplicaciones pueden ayudar tanto al instructor como al estudiante a mejorar la calidad de la educación.

En este trabajo se hace uso del análisis de clusters para segmentar estudiantes en grupos de acuerdo a sus características.

El *Clustering de Datos* es una técnica estadística de análisis de datos sin supervisión. Se utiliza para la clasificación en grupos homogéneos descubriendo patrones y la relación oculta ayuda a tomar decisiones rápidas y eficientes.

Con esta técnica se segmenta un gran conjunto de datos en subconjuntos llamados *clusters* (grupos). Cada *cluster* es una colección de objetos que son similares entre sí y se colocan dentro del mismo grupo, pero son diferentes a los objetos en otros clusters.

En este trabajo se habrán de utilizar los conceptos de Lógica Difusa (*Fuzzy Logic*) para realizar el *clustering*, que finalmente desembocará en la determinación del indicador Grade con el cual establecer el desempeño estudiantil.

2. Agrupamiento de datos (*clustering*)

Un algoritmo de agrupamiento (*clustering*) organiza elementos en grupos en función de criterios de similitud. El algoritmo Fuzzy c-means es un algoritmo de agrupamiento, donde cada elemento puede pertenecer a más de un grupo (de ahí la palabra Fuzzy, 'difuso'), donde el grado de pertenencia para cada elemento está dado por una distribución de probabilidad sobre los clusters.

a - Algoritmo Fuzzy c-Means (FCM). Es un algoritmo de agrupamiento desarrollado por Dunn y posteriormente mejorada por Bezdek. Es útil cuando el número requerido de

clusters está predeterminado. Por lo tanto, el algoritmo trata de poner cada uno de los puntos de datos en uno de los clusters. Lo que hace a FCM diferente, es que no decide la asignación absoluta de un punto de los datos a un grupo determinado, en su lugar, se calcula la probabilidad (el grado de pertenencia) con que dicho punto pertenecerá a ese clúster.

Por lo tanto, dependiendo de la exactitud del clustering que se requiera en la práctica, se pueden poner medidas de tolerancia apropiadas en su lugar. Dado que no se calcula la composición absoluta, FCM puede ser extremadamente rápido ya que el número de iteraciones necesarias para lograr un agrupamiento específico corresponde a la precisión requerida.

a.1 - *Iteraciones.* En cada iteración del algoritmo FCM, se minimiza la siguiente función objetivo J :

$$J = \sum_{i=1}^N \sum_{j=1}^C \delta_{ij} \cdot \|x_i - c_j\|^2$$

Aquí, N es el número de puntos dato, C es el número de clusters requeridos, c_j es el vector de centros para el cluster j , y δ_{ij} es el grado de pertenencia para el i -ésimo punto dato x_i al el cluster j . La norma, $\|x_i - c_j\|$ mide la similitud (o cercanía) del punto dato x_i al vector de centros c_j del cluster j . Nótese que, en cada iteración, el algoritmo mantiene un vector de centros para cada uno de los clusters. Estos puntos datos se calculan como el promedio ponderado de los mismos, donde los pesos se dan por los grados de pertenencia.

a.2 - *Grados de Pertenencia.* Para un punto dato x_i , el grado de su pertenencia al cluster j se calcula como sigue:

$$\delta_{ij} = \frac{1}{\sum_{k=1}^C \left[\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right]^{\frac{2}{m-1}}}$$

donde, m es el *coeficiente de borrosidad* y el vector de centros c_j se calcula como sigue:

$$c_j = \frac{\sum_{i=1}^N \delta_{ij}^m \cdot x_i}{\sum_{i=1}^N \delta_{ij}^m}$$

En la ecuación (3) de arriba, δ_{ij} es el valor del grado de pertenencia calculado en la iteración previa.

Nótese que en el arranque del algoritmo, el grado de pertenencia para el punto dato i al cluster j se inicializó con un valor aleatorio θ_{ij} , $0 \leq \theta_{ij} \leq 1$, tal que $\sum_j^C \delta_{ij} = 1$.

a.3 - *Coeficiente de borrosidad.* En las ecuaciones (2) y (3) el coeficiente de difusión m , con

$1 < m < \infty$, mide la tolerancia del clustering requerido. Este valor determina cuántos clusters pueden superponerse unos con otros. Cuanto mayor sea el valor de m , mayor es

el solapamiento entre clusters. En otras palabras, cuanto mayor sea el coeficiente de borrosidad que utiliza el algoritmo, un mayor número de puntos datos caerá dentro de una banda 'difusa', donde el grado de pertenencia no es ni 0 ni 1, sino que estará en algún punto intermedio.

a.4 - *Condición de terminación.* La precisión requerida del grado de pertenencia determina el número de iteraciones realizadas por el algoritmo FCM. Esta medida de exactitud se calcula utilizando el grado de pertenencia de una iteración a la siguiente, tomando el mayor de estos valores en todos los puntos datos teniendo en cuenta todos los grupos. Si se representa la medida de la exactitud entre iteración k y $k + 1$ con ε , se calcula su valor de la siguiente manera:

$$\varepsilon = \Delta_i^N \cdot \Delta_j^C \cdot |\delta_{ij}^{k+1} - \delta_{ij}^k|$$

donde, δ_{ij}^k y δ_{ij}^{k+1} son respectivamente los grados de pertenencia en la iteración k y $k + 1$, y el operador Δ , cuando se suministra un vector de valores, devuelve el valor más grande de ese vector.

3. Lógica Difusa (*Fuzzy Logic*)

La lógica difusa, maneja la imprecisión y la incertidumbre de una manera natural, donde es posible proporcionar una representación del conocimiento con orientación humana [4].

En la literatura de los últimos años aparecen muchos trabajos de investigación en los que la lógica difusa, las redes neuronales clásicas y la red neuronal difusa han sido empleadas en los sistemas de modelización del estudiante.

En este trabajo se utiliza el algoritmo *fuzzy C-means Clustering* para la generación automática de la función de pertenencia.

También se aplica el *Rule Based Fuzzy Expert System* que convierte automáticamente los *datos nítidos* en un *conjunto difuso* y también calcula la puntuación total de un estudiante en los exámenes del parc-1, parc-2 y parc-3.

Este tipo de conocimiento del sistema de gestión puede mejorar las políticas y las estrategias para mejorar la calidad del sistema.

Una arquitectura típica se muestra en la figura 1.

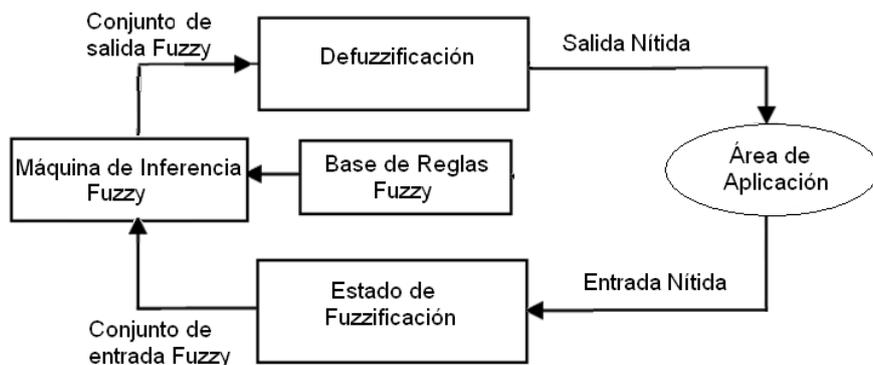


Fig. 1. Sistema básico de lógica difusa.

- i). **Valor Nítido** (*Crisp value*): el valor nítido es la nota de estudiante obtenida en los exámenes parciales constituyendo la *entrada nítida*.
- ii). **Fuzzificación**: Fuzzificación significa que valor nítido (nota del estudiante) se convierte en el valor de entrada Fuzzy con la ayuda de la función de pertenencia adecuada.
- iii). **Mecanismo de Inferencia**: Define los diferentes tipos de reglas difusas ("*Regla If/Then*") para evaluar la performance académica de los estudiantes.
- iv). **Salida Fuzzy**: Determina un valor de salida de la función de pertenencia para cada regla activa ("*Regla If/Then*").
- v). **Defuzzificación** (*Performances*): Defuzzificación significa calcular el resultado final (valor de performance) con ayuda del método de defuzzificación adecuado. En este trabajo de investigación, se utiliza Centro de Área (COA) para Defuzzificación (evaluación de performance)

La *función de pertenencia* (o membresía) es una representación gráfica de la magnitud de la participación de cada entrada. Un gráfico que define cómo a cada punto en el espacio de entrada se le asigna el valor de pertenencia entre 0 y 1. El espacio de entrada se refiere a menudo como el *universo del discurso* o *conjunto universal*, que contiene todos los posibles elementos de interés particular.

Se asocia un “peso” con cada una de las entradas que se procesan, se define la superposición funcional entre entradas y finalmente se determina una respuesta de salida.

Las reglas utilizan los valores de pertenencia de entrada como factores de ponderación para determinar su influencia sobre los conjuntos de salida difusos de la conclusión de la salida final. Una vez que las funciones se infieren, son escaladas y se combinan, ellas son defuzzificadas en una salida nítida que acciona el sistema.

4. Fuzzificación

La Fuzzificación comprende el proceso de transformación de los valores nítidos en grados de pertenencia para *términos lingüísticos* de conjuntos difusos. La función de pertenencia se utiliza para asociar una calificación a cada *término lingüístico*. También se refiere a la transformación de un término objetivo en un concepto difuso.

Toda esta actividad la realiza Matlab [5] gracias al Toolbox de Lógica Difusa, a través de la función *fcm*, cuyo formato es:

```
[centers,U,objFun] = fcm(data,Nc,options)
```

Las entradas para esta función son:

- *data*: Matriz con los datos de entrada que han de agruparse, especificada como una matriz con N_d filas, donde N_d es el número de puntos de datos. El número de columnas de datos es igual a la dimensionalidad de los datos
- *Nc*: Número de clusters elegido por el usuario
- *Options(1)*: Exponente de la matriz de partición difusa U , especificado como un escalar mayor que 1.0. Esta opción controla la cantidad de superposición difusa entre los grupos, con valores mayores indicando un mayor grado de solapamiento (default=2)
- *Options(2)*: Máximo número de iteraciones (default=100)

- *Options(3)*: Diferencia entre variaciones de centroide deseada (default=1e-5)
- *Options(4)*: Mostrar iteraciones (default=1)

Mientras que las salidas son:

- *center*: Coordenadas de los Centros de clusters finales, devuelta como una matriz con N_c filas que contienen las coordenadas de cada centro de cluster. El número de columnas en los centros es igual a la dimensionalidad de los datos a ser agrupados.
- *U*: Matriz de partición difusa, devuelta como una matriz con N_c filas y N_d columnas. El elemento $U(i, j)$ indica el grado de pertenencia del punto de datos j en el cluster i . Para un punto de datos dado, la suma de los valores de pertenencia para todos los grupos es uno.
- *objFun*: Valores de la función objetivo para cada iteración, devuelta como un vector.

Para este trabajo, se han utilizado las notas obtenidas por alumnos de la Cátedra de Bioestadística de la Facultad de Ciencias de la Salud (Universidad de Mendoza) en el Ciclo Lectivo 2012.

Estos datos conforman una matriz numérica (*data*) de dimensión $n \times 3$ donde $n=124$ es el número de alumnos evaluados y en las columnas primera, segunda y tercera figuran los valores obtenidos en *parc-1*, *parc-2* y *parc-3*,

84	84	5
70	44	45
65	44	25
.....	.	.
.....	.	.

El número de clusters elegido N_c es de 5 [*Muy Alto (VH)*, *Alto (H)*, *Promedio (A)*, *Bajo (L)*, *Muy Bajo (VL)*] sin ninguna otra opción, de modo que se toman los valores de default (2, 100, 1e-5, 1).

Como salida se obtiene la matriz *centers* (dimensión 5x3) con las coordenadas de los centros de clusters

68.2672	81.5736	52.5666
66.7560	37.5156	41.5341
81.0763	88.7203	78.3407
57.0252	39.7127	23.6533
80.7372	63.5113	76.8585

y la matriz U^T (dimensión $n \times 5$) donde los elementos de cada columna pertenecen a cada uno de los cinco grupos:

0.2935	0.1947	0.1365	0.2432	0.1320
0.0363	0.8284	0.0165	0.0833	0.0354
0.0286	0.1959	0.0122	0.7445	0.0188
0.1041	0.0393	0.6323	0.0283	0.1960
0.0881	0.0941	0.1185	0.0469	0.6525
.....
.....

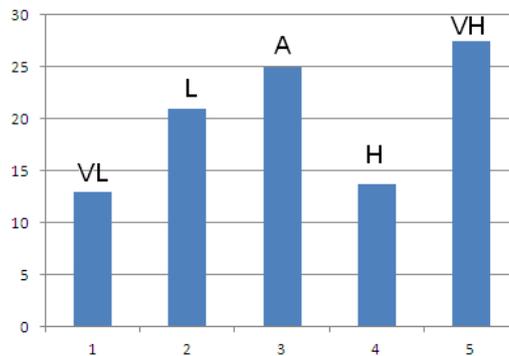
A partir de allí se determina la posición del máximo de los elementos de cada una de las filas, asignándola (índice de la posición) a un elemento en una sexta columna. Posteriormente, se agrupa cada alumno a una de las cinco categorías (de izquierda a derecha, VH: muy alto, H: alto, A: promedio, L: bajo y VL: muy bajo).

0.1365	0.1320	0.2935	0.1947	0.2432
0.0165	0.0354	0.0363	0.8284	0.0833
0.0122	0.0188	0.0286	0.1959	0.7445
0.6323	0.1960	0.1041	0.0393	0.0283
0.1185	0.6525	0.0881	0.0941	0.0469.

..... . .

De la simple observación del máximo por fila, el primer alumno está en la categoría A, el segundo en la L, el tercero en la VL, el cuarto en la VH, etc.

Con lo cual, porcentualmente, el clustering en los cinco grupos conforme a esta técnica presenta el siguiente formato:



5. Defuzzification

Cuando la inferencia ha terminado, existe la necesidad de calcular un valor único para representar el resultado. Este proceso se llama *defuzzificación*. Esto se puede lograr por diferentes métodos. Un método común es la *defuzzification* de los datos en una salida nítida lograda mediante la combinación de los resultados del proceso de inferencia y luego calculando el "centroide fuzzy" del área.

Las fortalezas ponderadas de cada función de pertenencia de salida se multiplican por sus respectivos puntos centrales de la función de pertenencia de salida y se suman. Por último, esta área se divide por la suma de las fortalezas de la función de pertenencia ponderadas y el resultado se toma como la salida nítida.

Base de Reglas Fuzzy (Knowledge Base): Las reglas Fuzzy *if-then* y el razonamiento difuso son la columna vertebral de los sistemas expertos difusos, que son las herramientas de modelado más importantes basados en la teoría de conjuntos difusos. La base de reglas se caracteriza en forma de reglas *if-then* en que los antecedentes y consecuentes implican variables lingüísticas. En este trabajo, se utiliza *muy alto*, *alto*, *medio*, *bajo* y *muy bajo* como variable lingüística.

La colección de estas reglas constituye la base de reglas para el sistema de lógica difusa. En este sistema experto difuso basado en reglas, se han utilizado las siguientes reglas para encontrar la base de conocimientos:

1. Si el estudiante pertenece a *muy alto*, entonces ($Z_1=a_1+b_1X_1+ c_1Y_1$)
2. Si el estudiante pertenecen a *alto*, entonces ($Z_2=a_2+b_2X_2+ c_2Y_2$)
3. Si el estudiante pertenece a *medio* entonces ($Z_3=a_3+b_3X_3+ c_3Y_3$)
4. Si el estudiante pertenece a *bajo* entonces ($Z_4=a_4+b_4X_4+ c_4Y_5$)
5. Si el estudiante pertenece a *muy bajo* entonces ($Z_5=a_5+b_5X_5+ c_5Y_5$)

Donde X_i e Y_i son las notas que los estudiantes agrupados en el cluster i obtuvieron en los exámenes parc-1 y parc-2, respectivamente, Z_i los correspondientes a parc-1. $a_1, \dots, a_5, b_1, \dots, b_5$ y c_1, \dots, c_5 son constantes a determinar por el método del *modelo de análisis de regresión*.

Motor de inferencia (Lógica de Toma de Decisiones): Usando el procedimiento de inferencia adecuado, se calcula el valor verdadero para el antecedente de cada regla y se aplica a la parte consecuente de cada regla.

En este caso, se ha utilizado el modelo de análisis de regresión lineal para la toma de decisiones. Esto resulta en un subconjunto borroso que se asignará a cada variable de salida para cada regla.

Una vez más, mediante el uso de procedimiento de composición adecuada, todos los subconjuntos borrosos a ser asignados a cada variable de salida se combinan entre sí para formar un único subconjunto borroso para cada variable de salida.

(iv) *Interfaz de Defuzzificación*: Defuzzificación significa convertir la salida difusa en salida nítida. Aquí, se ha utilizado la técnica defuzzificación *height* para convertir la *salida difusa* en *salida nítida* (valor de performance de los estudiantes).

La fórmula de defuzzificación (Takagi-Sugeno-Kang Model) es la siguiente:

$$Y = \frac{\mu_{VH}(x,y).Z_1 + \mu_H(x,y).Z_2 + \mu_A(x,y).Z_3 + \mu_L(x,y).Z_4 + \mu_{VL}(x,y).Z_5}{\mu_{VH}(x,y) + \mu_H(x,y) + \mu_A(x,y) + \mu_L(x,y) + \mu_{VL}(x,y)}$$

Con la ayuda de esta ecuación, se puede convertir la salida difusa en salida nítida (valor de performance de un estudiante) obteniendo el indicador estándar llamado *Grade*.

El resultado para los primeros registros se muestra a continuación:

p1	p2	p3	VH	H	A	L	VL	Grade
84	84	5	0.2935	0.1947	0.1365	0.2432	0.1320	71.0740
70	44	45	0.0363	0.8284	0.0165	0.0833	0.0354	50.7925
65	44	25	0.0286	0.1959	0.0122	0.7445	0.0188	34.1655
90	93	96	0.1041	0.0393	0.6323	0.0283	0.1960	69.9056

En el apéndice se transcribe el programa en Matlab que realiza el total de las tareas, desde los datos en la matriz *data* hasta la obtención de los *Grades* correspondientes a cada alumno.

6. Conclusiones

Conforme a lo desarrollado, se ha presentado una nueva forma de agrupación de estudiantes en cuanto a su comportamiento académico, más significativo que con la mera promediación tradicional.

Se pone a disposición de los docentes y administradores pedagógicos de una herramienta completamente desarrollada para planificar conforme a una clasificación a partir de un indicador internacionalmente reconocido.

7. Bibliografía

- [1]. Zadeh, L.A. “*Fuzzy sets*”. *Information and Control*, 8, 338-354. 1965.
- [2]. Bai, S.M., Chen, S.M. “*Evaluating Students’ Learning Achievement Using Fuzzy membership functions and Fuzzy rules. Expert System with Applications*”, ELSEVIER. 2008. 34, 399-410.
- [3]. Chen, S.M., Lee, C.H. “*New Methods for Students’ Evaluation Using Fuzzy Sets. Fuzzy Sets and System*”, 104, 209-218. 1999.
- [4]. <http://radio.feld.cvut.cz/matlab/toolbox/fuzzy/fuzzyint.html>
- [5]. The Mathworks. (2009). *Fuzzy Logic Toolbox User’s Guide*, The Mathworks Inc. Retrieved September 10 2009 from http://www.mathworks.com/access/helpdesk/help/pdf_doc/fuzzy/fuzzy.pdf

Apéndice

```
function V=student
% La matriz V entrega los resultados.
% Carga datos de N estudiantes y sus notas de parc-1 y parc-2 y parc-3
% que están almacenados en un archivo ASCII estad.txt
B=load('c:\datos\estad.txt'); % A es una matriz Nx4
S=size(B);N=S(1); % Número de filas de la matriz
% Se ejecuta el Algoritmo Fuzzy C-Means para 5 clusters
[CENTER, U, OBJ_FCN] = fcm(B,5);
V=U';
% Cataloga en la 6ta. Columna a qué grupo pertenece cada alumno
for i=1:N,
    VV= V(i,:);M=max(VV);I=find(VV==M); V(i,6)=I;
end
% Reordena columnas
X=sort(mean(CENTER'),'descend'); Y=mean(CENTER');
for i=1:5,
    Z=find(X(i)==Y); W(:,i)=V(:,Z);
end
% Estimación de la salida (ajuste lineal)
for t=1:5,
    x=[]; y=[]; z=[];
    for i=1:N,
        if V(i,6)==t,
            x=[x B(i,1)]; y=[y B(i,2)];z=[z B(i,3)];
        end
    end
    T=0;w=1:length(x'); T(w)=1;
    XX=[T' x' y']; P=(XX'*XX)^-1*XX'*z';
    in(t)=P(1);s11(t)=P(2);s12(t)=P(3);
end
% Cálculo del Grade
for i=1:N,
    for j=1:5,
```

```
        Z(j)=B(i,2)*s11(j)+B(i,3)*s12(j)+in(j);
    end
    G(i)=sum(Z.*V(i,1:5))/sum(V(i,1:5));
end
for i=1:N,
    V(i,7)=G(i);
end
```