

Tutor Educativo Usando Máquina de Soporte Vectorial

Elena Fabiola Ruiz Ledesma

Escuela Superior de Cómputo. IPN. Av. Miguel Othón de Mendizábal s/n, Col. La Escalera, Del. Gustavo A.,
Madero, C.P. 07320, México, D.F., 57296000teléfono
efruiz@ipn.mx

David Araujo Díaz

Escuela Superior de Cómputo. IPN. Av. Miguel Othón de Mendizábal s/n, Col. La Escalera, Del. Gustavo A.,
Madero, C.P. 07320, México, D.F., 57296000teléfono
daraujo@ipn.mx

Juan Jesús Gutiérrez García

Escuela Superior de Cómputo. IPN. Av. Miguel Othón de Mendizábal s/n, Col. La Escalera, Del. Gustavo A.,
Madero, C.P. 07320, México, D.F., 57296000teléfono

Resumen

En el presente artículo se muestra la construcción de un tutor educativo que puede ser auxiliar en el proceso de enseñanza y aprendizaje de cualquier curso. La característica principal es que de acuerdo a la cantidad de aciertos y errores cometidos por el usuario al responder ejercicios o problemas, el sistema le muestra otros ejercicios de menor o mayor grado de dificultad, esto provoca en el estudiante aprender nuevos temas o repasar los vistos ya que va a su ritmo de trabajo. Se apoya en un sistema de Soporte Vectorial empleando como puntos clave los errores comunes que presentan los estudiantes de un tema en específico, como ejemplo se muestra el de factorización. Las pruebas con 220 estudiantes de primer semestre de su carrera revelan la utilidad de emplear un sistema de esta naturaleza.

Palabras Claves: Álgebra, aprendizaje, Sistema de soporte vectorial.

1. Introducción

En los últimos años las escuelas de Nivel Superior han aumentado su nivel académico considerablemente a fin de que sus egresados tengan un nivel de desarrollo rentable para las empresas, [1].

El software educativo forma parte de las tecnologías para la educación, capaz de transformar positivamente los procesos de enseñanza y aprendizaje, aportando nuevos y mejores caminos para el desarrollo educativo. Se espera que para un futuro sea una de las principales fuentes de enseñanza en la educación en el país [2].

Los enormes avances tecnológicos en materia de velocidad de comunicaciones y almacenamiento masivo hacen evidente la fusión de tecnologías como el software, la TV e Internet. Los alumnos podrían disfrutar materiales revisados y avalados por instancias educativas y por personas especializadas en la investigación educativa.

Los materiales obtenidos podrán orientar al alumno en su aprendizaje, encaminándolo en su desarrollo intelectual y a un ritmo personalizado.

A través de recursos como los sistemas expertos, el lenguaje natural, el reconocimiento de patrones y las redes neuronales, entre otros, le permite al software educativo convertirse en "tutor educativo". A través de un diálogo con el usuario, se puede diagnosticar su estado académico y sugerirle acciones a seguir para alcanzar la excelencia.

En el presente artículo se describe un tutor que se apoya en máquinas de soporte vectorial.

2. Desarrollo

2.1. Máquinas de soporte vectorial (SVM)

Las Máquinas de Soporte Vectorial (Support Vector Machine por sus siglas en inglés) son un conjunto de algoritmos de aprendizaje supervisados y creados por Vladimir Vapnik [3], cuyo fin es clasificar objetos en $x \in R^n$ basándose en la construcción de una regla lineal.

Dichos objetos pueden comprenderse como puntos en el espacio n dimensional de variables predictorias (clases etiquetadas) logrando con ello construir un modelo capaz de predecir la clase a la cual pertenece una nueva muestra.

En términos generales, dado un conjunto de datos de entrenamiento, la Máquina de Soporte Vectorial busca el hiperplano que separa de forma óptima los puntos de una clase de la otra,

sin que con ello se pierdan o se aíslen datos de una misma clase [4], esto se puede apreciar en la Fig. 1.

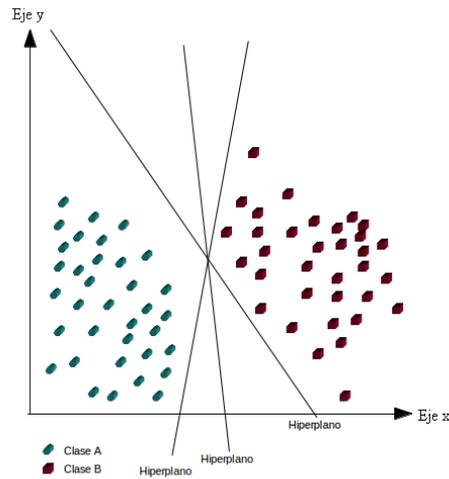


Fig. 1. Forma más simple de una máquina de soporte vectorial.

Las Máquinas de Soporte Vectorial se dividen principalmente en dos tipos:

- Modelos linealmente separables.
- Modelos no linealmente separables.

2.2. Modelos Linealmente Separables

Los datos se pueden dividir completamente por un hiperplano, el cual se ajusta de acuerdo a la máxima distancia (margen) que existe entre los puntos de las dos clases más cercanas entre sí [5].

Es importante señalar que en ciertas ocasiones una o ambas clases pueden contener uno o más puntos de la otra clase. Cuando esto sucede, a fin de permitir cierta flexibilidad, la Máquina de Soporte Vectorial maneja un parámetro C que sirve para controlar la compensación entre errores de entrenamiento y entre los márgenes planos. Creando así un margen blando o soft-margin (ver Fig. 2) el cual permite un cierto número de errores de clasificación.

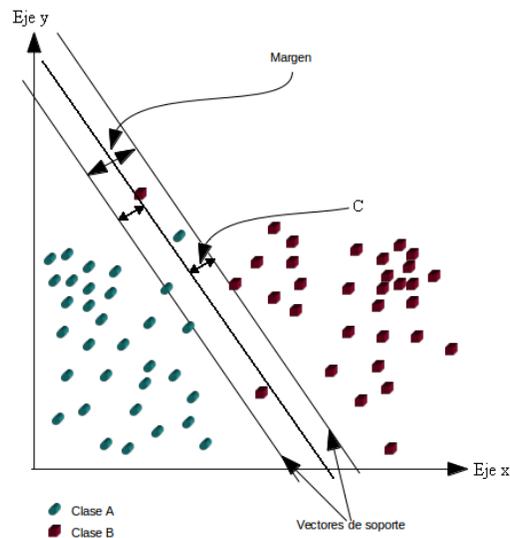


Fig. 2. Margen blando.

C sirve para determinar la holgura del margen blando y tiene una fuerte influencia en el correcto funcionamiento de la Máquina de Soporte Vectorial.

2.4.2. Modelos no linealmente separables

Los datos no se pueden separar linealmente por ello se hace necesaria un cambio de espacio mediante una función de transformación (función kernel) que aumenta la dimensionalidad de los vectores de entrada a un espacio por el cual sea posible separar los datos linealmente por un hiperplano.

Los principales tipos de funciones kernel se listan a continuación, [6], [7]:

- Kernel polinomial de grado d

$$k(x, y) = (x^T y + 1)^d \quad (1)$$

- Kernel de base radial de ancho S (gaussianas)

$$k(x, y) = e^{-\frac{\|x - y\|^2}{2\alpha^2}} \quad (2)$$

- Kernel sinusoidal con parámetros k y q

$$k(x, y) = \tanh(k x^T y + \theta) \quad (3)$$

Este kernel presenta un problema ya que no satisface la condición de Mercer (definida positiva) para k y q .

Condición de Mercer: Para toda función $g(x)$ donde:

$$\int g(x)^2 dx \tag{4}$$

$$\int k(x, y)g(x)g(y) dx dy \geq 0 \tag{5}$$

Es finita.

La condición de Mercer debe ser satisfecha para que una función se pueda considerar función kernel.

2.3. Obtención de datos de entrenamiento

Dado que el sistema emplea una Máquina de Soporte Vectorial, por cada tema y en específico por cada grupo de problemas que tengan una relación en concreto, fue necesario definir una serie de problemas clave los cuales estuvieron diseñados de tal modo que al momento de aplicarlos y de obtener los resultados, fueron arrojados los principales puntos donde los alumnos cometieron más errores y con ello fueron identificadas las dificultades que presentaron en conjunto dichos alumnos.

2.4. Problemas clave

Se trabajó con los grupos de estudiantes que recién habían ingresado a la Escuela Superior de Cómputo del Instituto Politécnico Nacional (ESCOM. IPN). Es importante hacer mención que los estudiantes que ingresan a ESCOM traen deficiencias en las Unidades de Aprendizaje de Álgebra, Trigonometría, Geometría Analítica y Cálculo, ésta es otra razón por la cual se considera relevante el diseño e implementación del sistema que se muestra en este trabajo.

Para determinar las deficiencias que tienen los estudiantes que ingresan se aplica un cuestionario diagnóstico, de tal forma que se observó en el semestre 2013/2014/1 sólo aprobaron dicho cuestionario el 14.07%.

Se consideró el tema de factorización como ejemplo para trabajar el sistema y determinar los problemas clave, es decir aquellos en donde presentan dificultades en su resolución y la

determinación de los errores cometidos. En la Fig. 3 se muestran los ejercicios aplicados a los estudiantes.

Los errores comunes encontrados consistieron en: Multiplicar exponentes en lugar de sumarlos, dividir exponentes en lugar de restarlos, sumar literales y restar literales.

El sistema Tutor se compone básicamente de dos partes:

Evaluación de conocimientos algebraicos

Por favor, resuelve cada uno de los problemas y ejercicios expuestos a continuación.

Problemas

Resolver los siguientes problemas

1. Se tiene un cuadrado con un área de $4a^2 + ab + [b^2/16]$
¿Cuál es el tamaño del lado del cuadrado?

(Otra opción de la ecuación que nos dio para el mismo ejercicio de arriba es:
 $A^{2n} - b^{-2n} - 2 + a^{-2n}b^{2n}$ Aunque más complicado).

2. Tenemos una pirámide de base cuadrada y altura h, el volumen de la pirámide es $3^{2n}h + 2h + a^{-2n}h$.
¿Calcular el tamaño de la base de la pirámide?

Ejercicios

Resolver los siguientes ejercicios.

1. $a^2 + ab + (b^2/4)$
2. $4a^2 + ab + (b^2/16)$
3. $a^{2n} + 2 + a^{-2n}$
4. $a^{2n}b^{-2n} - 2 + a^{-2n}b^{2n}$
5. $(a+b)^{2n} + 2(a+b)^n c + c^2$

Factorizar las siguientes ecuaciones.

1. $a^2 + ab + ax + bx$
2. $am - bm + an - bn$

Reducir las siguientes ecuaciones.

1. $15c^3d^2 + 60c^2d^3$
2. $a^2b^2c^2 - a^2c^2x^2 + a^2c^2y^2$

Fig. 3. Reactivos sobre factorización.

- Aplicación: Parte gráfica compuesta por la interfaz con la cual el usuario es capaz de interactuar con el núcleo.
- Núcleo: Parte interna del sistema la cual se encarga del procesamiento de la información.

Para el funcionamiento del sistema, la aplicación manda una petición al núcleo, el cual analiza dicha petición y genera una respuesta adecuada. Así por ejemplo si la petición es un nuevo tema (mensaje identificado por la etiqueta "tema_x" donde 'x' es el nombre del tema o subtema solicitado), el núcleo manda un paquete que contiene toda la información (definiciones, ejemplos, problemas, ejercicios y ayudas del mismo) necesaria para poder estudiar el tema. Cabe destacar que la información contenida en el paquete se genera tomando en cuenta las formas en las que el usuario (o alumno) ha demostrado tener mejor retención de la información. Posteriormente, cuando el usuario ha terminado de estudiar el tema y ha respondido todos los ejercicios y problemas del mismo, la aplicación envía una estructura al núcleo la cual contiene los resultados obtenidos en el estudio del tema. Con ésta, el núcleo es capaz de determinar si el tema ha sido retenido por el usuario o bien, si es necesario reforzarlo. En caso de que el usuario haya demostrado un nivel aceptable de entendimiento en la resolución de los ejercicios y problemas, el núcleo manda un nuevo paquete a la aplicación con el tema que sigue; en caso contrario, el sistema genera un paquete con el mismo tema pero con la diferencia de que el contenido estará orientado a hacer ver al usuario donde tuvo sus errores para que con ello los corrija y continúe con su aprendizaje.

De igual forma si la petición es una evaluación inicial de prueba para poder determinar las habilidades del usuario (mensaje identificado por la etiqueta "eva_cero"), la aplicación envía un mensaje al núcleo el cual responde con un paquete de ejercicios y problemas que contemplan todos los temas existentes actualmente en el sistema, ver Fig. 4.

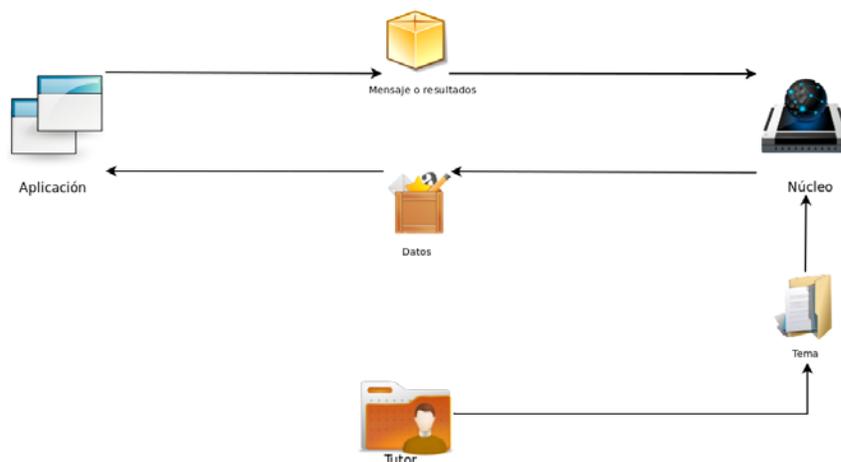


Fig. 4. Representación gráfica del sistema.

Cabe destacar que el sistema es una herramienta de apoyo al usuario y que con ello no es capaz de suplir las clases presenciales y las tutorías de un profesor. Sin embargo, es una buena herramienta para reforzar los conocimientos en los estudiantes.

Durante esta etapa, el desarrollo del software se lleva a cabo como un conjunto de programas. La prueba de unidades implica verificar que cada una cumpla su especificación de acuerdo a lo que se propuso en la etapa de diseño donde se identifican y describen las abstracciones fundamentales de la herramienta y sus relaciones. Después de esto, los programas y unidades se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software.

Cabe mencionar que los puntos que se presentan a continuación no se realizaron de forma consecutiva como se muestran, la metodología permite realizar actividades intermedias de lo que es la herramienta final.

Le etapa de desarrollo comenzó con el cambio de algunas de las especificaciones del sistema ya que la metodología de desarrollo evolutivo lo permite, y uno de estos cambios fue la implementación de una base de datos.

El sistema cuenta con 4 módulos, de los cuales solamente podrá interactuar con dos de ellos, uno como usuario y el otro como maestro para la creación de nuevos contenidos.

Los cuatro módulos fueron creados por separado ya que funcionan independientemente uno del otro y se comunican por medio de envío de mensajes.

A continuación se muestra a detalle los procesos para el desarrollo del sistema Tutor

En la Fig. 5 se muestra el caso de uso General de las funciones de la herramienta

En la Fig. 6 se muestra el caso de uso de la “Evaluación cero”:

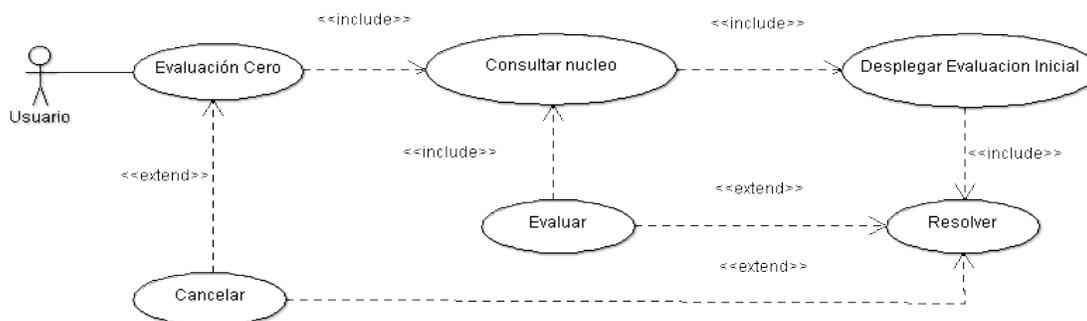


Fig. 5. Caso de uso general.

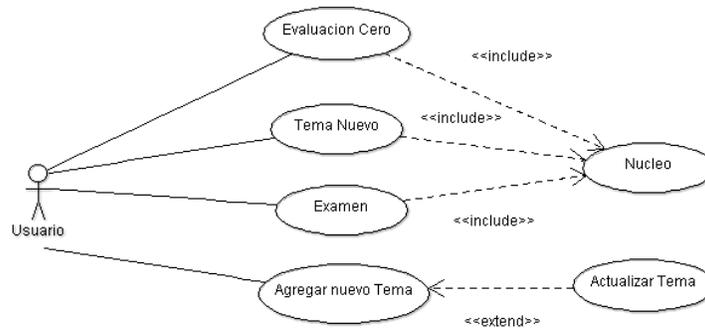


Fig. 6. Caso de uso Evaluación Cero.

Tabla 1. Descripción del uso Evaluación cero.

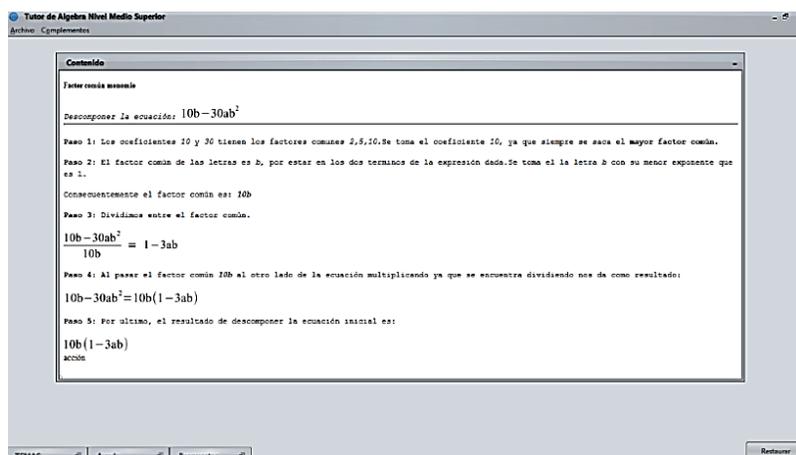
| | |
|----------------------------|--|
| Actores principales | Usuario |
| Condición inicial | Ingresar el sistema |
| Descripción | Es la presentación de inicio de la sección de evaluación cero de la herramienta. |
| Flujo principal | <ol style="list-style-type: none"> 1. SISTEMA. Muestra la ventana para aceptar realizar la evaluación cero. 2. USUARIO. Aceptar la propuesta de empezar evaluación cero. [Flujo alternativo A] 3. SISTEMA. Manda un mensaje al núcleo de que se aceptó la petición. |
| Flujo alternativo A | <ol style="list-style-type: none"> 1. USUARIO. Cancela la petición. 2. SISTEMA. Propone realizar la evaluación cero |
| Condición de salida | Petición enviada al núcleo. |

Al ejecutar el sistema Tutor se muestra la ventana principal de la aplicación como se aprecia en la figura 7, desde la cual se puede acceder a la función de Menú, además en las secciones de la ventana seleccionada se puede acceder al temario tipo árbol en el panel superior izquierdo, al desarrollo del tema en el panel superior derecho, a la ayuda del tema en el panel inferior izquierdo y a la sección de incisos en el panel inferior derecho.



Fig. 7. Pantalla principal personalizada.

En el panel, mostrado en la Fig. 8 se presenta toda la información contenida acerca del tema seleccionado en el menú tipo árbol del panel superior izquierdo. Dicho contenido es construido a partir de un XML generado por el núcleo.

**Fig. 8. Panel de contenidos.**

En el panel de ejercicios, se presentan las opciones de cada ejercicio. El usuario tiene que elegir la opción que crea correcta y pulsar el botón “siguiente” para pasar al siguiente problema/ejercicio. Al terminar los ejercicios del tema o de la evaluación pulsará el botón “Finalizar” para enviar los resultados al núcleo y recibir su evaluación. Sin embargo, si el usuario no desea terminar la evaluación solo tendrá que pulsar el botón que dice terminar evaluación, si es que se encuentra realizando una evaluación, o cambiar de tema eligiendo uno diferente si solo realiza un ejercicio de algún tema de menú del panel superior izquierdo.

3. Resultados

Se evaluaron un total de 220 alumnos de nuevo ingreso de la Escuela Superior de Cómputo del IPN. En la tabla 2 se muestran los resultados obtenidos, identificando errores y deficiencias que presentaron los alumnos.

De acuerdo a los resultados obtenidos en la aplicación del cuestionario anteriormente expuesto, se determinaron los puntos clave (errores comunes) donde los alumnos han demostrado deficiencias algebraicas.

Tabla 2 Número de respuestas por actividad.

| Tipo | Resueltos | Tiene procedimiento | Solo resultado | Resultados correctos |
|----------------------------------|-----------|---------------------|----------------|----------------------|
| Problemas (factorización) | | | | |
| Problema 1 | 140 | 40 | 100 | 80 |
| Problema 2 | 20 | 10 | 10 | 0 |
| Resolver (grupo 1) | | | | |
| Ejercicio 1 | 150 | 60 | 90 | 80 |
| Ejercicio 2 | 110 | 40 | 70 | 50 |
| Ejercicio 3 | 90 | 20 | 70 | 20 |
| Ejercicio 4 | 70 | 20 | 50 | 40 |
| Ejercicio 5 | 90 | 30 | 60 | 60 |
| Factorizar (grupo 2) | | | | |
| Ejercicio 1 | 220 | 90 | 130 | 160 |
| Ejercicio 2 | 220 | 100 | 120 | 190 |
| Reducir (Grupo 3) | | | | |
| Ejercicio 1 | 150 | 20 | 130 | 100 |
| Ejercicio 2 | 170 | 30 | 140 | 110 |

Tabla 3 Errores comunes.

| Errores comunes | | | | | |
|------------------------|----------------|--------------|-------------------|---------------|-------------------|
| | <i>Suma</i> | <i>Resta</i> | <i>Multiplica</i> | <i>Divide</i> | <i>Simplifica</i> |
| Problemas | | | | | |
| Problema 1 | 20 | | | 10 | 30 |
| Problema 2 | | | | | 10 |
| Ejercicios | Errores | | | | |
| Grupo 1 | | | | | |
| Ejercicio 1 | 10 | | | 10 | |
| Ejercicio 2 | | | 10 | | 20 |
| Ejercicio 3 | 10 | | 20 | 10 | 30 |
| Ejercicio 4 | 20 | | 10 | | 20 |
| Ejercicio 5 | 20 | | 10 | | 20 |
| Grupo 2 | | | | | |
| Ejercicio 1 | 10 | | 10 | | 20 |
| Ejercicio 2 | 10 | 20 | | | 20 |
| Grupo 3 | | | | | |
| Ejercicio 1 | | | | 10 | 40 |
| Ejercicio 2 | | | 3 | | 30 |

Los errores cometidos por los estudiantes fueron:

Multiplicar los exponentes de las literales en lugar de sumarlos.

Dividir los exponentes de las literales en lugar de restarlos.

Los 220 estudiantes de la muestra tuvieron errores en el cuestionario de factorización, y de acuerdo a la deficiencia que tenían, la computadora les presentaba una guía con ejercicios resueltos para apoyar a la comprensión de la resolución de éstos. Después de ello se solicitaba al estudiante resolver otros ejercicios con el mismo grado de dificultad al que se habían enfrentado previamente.

La Máquina de soporte vectorial empleó dos casos, debido a que los problemas y ejercicios contemplan la posibilidad de un cierto grado de certeza y error.

Para entrenar una SVM se formula el problema por medio de programación cuadrática convexa, pero debido a que se requiere una gran cantidad de datos de entrenamiento y el consumo indiscriminado de recursos computacionales, se optó por emplear el método denominado Optimización de Secuencia Mínima (SMO).

A grandes rasgos, Platt [8] y Candell [9], señalan que la estructura de un SMO se define como:

- Criterio de término: Determina eficazmente cuándo se ha llegado a una solución óptima o cercana a la óptima, definida dentro de un margen pre-establecido.
- Paso de optimización: Por lo general es analítico y sirve para calcular los nuevos valores de dos multiplicadores de Lagrange.
- Estrategia de selección: (Heurística o no) útil para elegir a los dos multiplicadores de Lagrange a ser optimizados.
- Sistema de actualización: Actualiza los valores involucrados en la selección y optimización de los dos multiplicadores de Lagrange.

3.1. Prueba número uno al módulo núcleo de la aplicación

Las pruebas realizadas con las Máquinas de Soporte Vectorial dieron como resultado valores aceptables dentro de los rangos de error y acierto, lo cual permitió definir qué tan certero o erróneo fue el procedimiento que el alumno usó al momento de ser evaluado por el sistema.

En la Fig. 9 se muestran los resultados de la primera versión del sistema.

Distancias entre los puntos obtenidos con relación al eje:

- [1] 10.254791
- [2] 2.547699
- [3] 4.1380463
- [4] 5.361397
- [5] 3.5263748
- [6] -6.6274185
- [7] -5.159401
- [8] -1.0000153
- [9] -3.6913757
- [10] -5.6487427

```
==Kernel lineal==
Archivo base: Arch/Datos_prueba
#> Iteración = 0
[0]Elemento libre[2]->[1] = 0.049772244 < 0.5
[1]Elemento libre[6]->[1] = 0.0074392324 < 0.5
[2]Elemento libre[8]->[1] = 0.04233301 < 0.5
[3]Elemento libre[10]->[1] = 1.6298145E-9 < 0.5
---
#> Iteración = 1
[0]Elemento libre[2]->[1] = 0.05445463 < 0.5
[1]Elemento libre[8]->[1] = 0.049772244 < 0.5
[2]Elemento libre[10]->[1] = 0.0046823863 < 0.5
---
#> Iteración = 2
[0]Elemento libre[2]->[1] = 0.13162065 < 0.5
[1]Elemento libre[8]->[1] = 0.13162066 < 0.5
---
#> Iteración = 3
[0]Elemento libre[2]->[1] = 0.13162065 < 0.5
[1]Elemento libre[8]->[1] = 0.13162066 < 0.5
---
#> Iteración = 4
[0]Elemento libre[2]->[1] = 0.13162065 < 0.5
[1]Elemento libre[8]->[1] = 0.13162066 < 0.5
---

==Kernel lineal==
Archivo base: Arch/Datos_prueba
Iteraciones=5
vectores soporte libres= 2      vectores soporte unidos= 0
Umbral=128.0949
Tasa de error=0.0
Tiempo transcurrido = 0.378
```

Fig. 9 Resultados de la Prueba uno de la Consola de Java,

La gráfica de esta prueba (ver Fig. 10), muestra que al encontrarse prácticamente separados los puntos con valor positivo de los puntos con valor negativo, la MSV ha realizado la agrupación y el cálculo de distancias de la recta que divide a los dos grupos.

4. Discusión

Este trabajo se desarrolló como una nueva propuesta de software educativo, la cual aún puede crecer y expandirse. Al hacer uso de archivos XML como base de datos permite que personas especializadas en el área puedan agregar más contenidos y enriquecer los contenidos de la aplicación. Además no está destinada solo al estudio del álgebra, esta herramienta puede expandirse a una gran variedad de materia y temas.

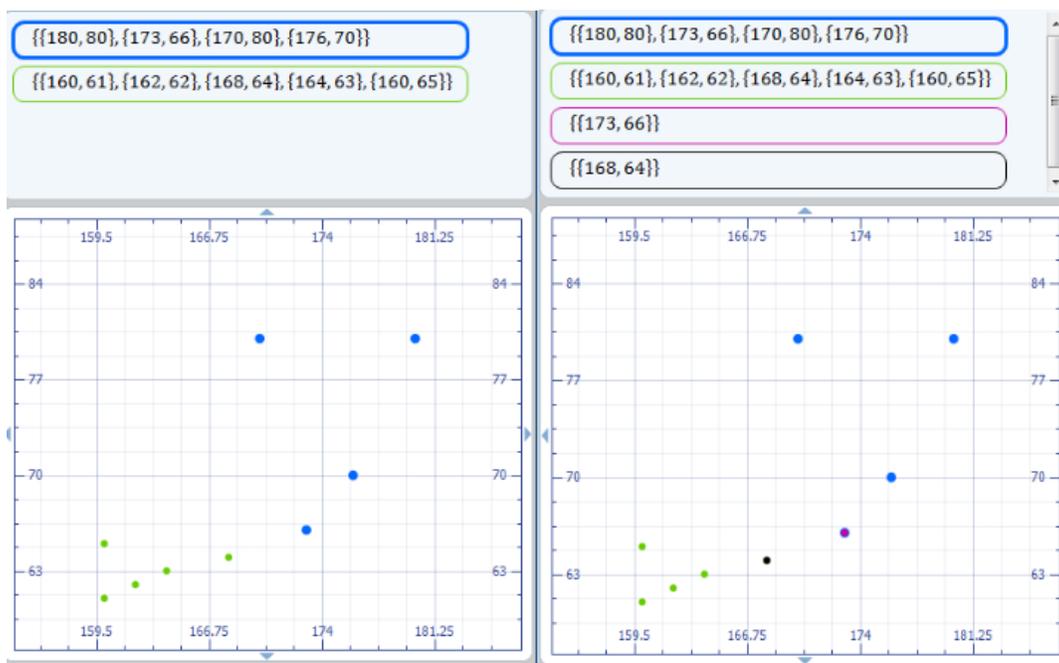


Fig. 10. Prueba.

5. Conclusiones

A lo largo del presente trabajo se ha desarrollado un sistema que se puede implementar en todas las escuelas o en instituciones que busquen el aprendizaje y comprensión de los diversos temas impartidos al alumno, tomando en cuenta que la aplicación no puede reemplazar al personal docente, si no, como un complemento y ayuda de los mismos.

Con base en lo anterior se puede concluir que el uso del software educativo ha crecido, pero es importante tener en cuenta aspectos relevantes para su desarrollo e implementación como la tecnología y los fundamentos educativos que contiene. Es por ello que en el Tutor Educador en comparación con otras propuestas de software educativo para el área de Matemáticas se implementó una nueva herramienta, como el uso de la Máquina de Soporte Vectorial.

Como punto final hay que tomar en cuenta que el sistema Tutor de Álgebra Nivel Medio Superior es una que puede crecer aún más, no solo concentrándose en el álgebra, si no en muchas ramas de la educación, siendo las matemáticas la principal opción al haber encontrado un nivel tan bajo en los estudiantes que ingresan a nivel medio superior.

6. Referencias

- [1] S. Pérez y M. Pérez, "Repositorio digital IPN", 2007. [En línea] Disponible: http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/2475/134_-_134-P%C3%A9rezVeraSandraMercedes.pdf?sequence=1. [Último acceso: 2 de mayo de 2013].
- [2] R. E. Pintos, S. I. Mariño, M. V. Godoy, "La realidad virtual como herramienta en la enseñanza-aprendizaje de la anatomía humana para el nivel EGB II" [En línea] Disponible: http://sedici.unlp.edu.ar/bitstream/handle/10915/19521/Documento_completo____.pdf?sequence=1 [2] M. Carretero, "Construir y enseñar, Las Ciencias Experimentales", Buenos Aires Argentina, 1998.
- [3] Vapnik, V. (1995). "Support-vector networks". *Machine Learning* 20 (3): 273. doi:10.1007/BF00994018
- [4] L. González A. (2003), "Modelos de Clasificación basados en Máquinas de Vectores: <http://www.asepelt.org/ficheros/File/Anales/2003%20-%20Almeria/asepeltPDF/55.pdf> [Ultimo acceso: 19 de abril 2013].
- [5] J. Vargas 2012, "Máquinas de soporte vectorial: metodología y aplicación en R.": <http://conferencias.unc.edu.ar/index.php/xclatse/clatse2012/paper/view/265>. [Ultimo acceso: 26 de abril 2013].
- [6] S. Amari. (1999). Improving support vector machine classifiers by modifying kernel functions *Neural Networks*, Volume 12, Issue 6, Pages 783-789.
- [7] Perea Estrada B. M. (2007), "Support Vector Machine." [En línea] https://www.u-cursos.cl/ingenieria/2007/1/IN60E/1/material_docente/objeto/125942 [Último acceso: 30 de abril 2013]

- [8] C. Platt J. (1998), "Fast Training of Support Vector Machines Using Sequential Minimal Optimization." <http://research.microsoft.com/pubs/68391/smo-book.pdf> [Último acceso: 26 de abril 2013].
- [9] C. Candel D. (2010), "Algoritmo tipo smo para la ad-svm aplicado a clasificación multicategoría." [En línea] <http://www.alumnos.inf.utfsm.cl/~dcontard/tesis.pdf> [Último acceso: 30 de abril: 2013]

7. Autores

Dra. Elena Fabiola Ruiz Ledesma obtuvo su título de doctorado en Ciencias con especialidad en Matemática Educativa en CINVESTAV. IPN México. D. F.

M. en C. David Araujo Díaz obtuvo su título de maestro en Ciencias especialidad computación en CINVESTAV. IPN México, D. F.

M. en C. Juan Jesús Gutiérrez García obtuvo su título de maestro en Ciencias en la especialidad Dispositivos móviles en la ESCOM. IPN

8. Agradecimientos

Los autores agradecen a la secretaría de Investigación y Posgrado del IPN por las facilidades para realizar el proyecto registrado con número 20131825.