

Aproximación de funciones implícitas de decisión por medio de máquinas de soporte vectorial

Jorge Hurtado y Diego Alvarez

Universidad Nacional de Colombia,
Apartado 127, Manizales, Colombia
Tel.: 57-68-74 86 97; Fax: 57-68-86 39 06
e-mail: jhurtado14@epm.net.co

Resumen

Tanto en sistemas mecánicos como estructurales, los análisis de optimización, de confiabilidad y de estabilidad dinámica se formulan en términos de funciones, generalmente no lineales, que definen la separación entre dominios. En el caso de optimización los dominios son los de diseño viable e inviable, en el estudio de confiabilidad son los de diseño seguro e inseguro y en la estabilidad dinámica los de estable o inestable. Al emplear programas de cálculo numérico como los basados en elementos finitos o elementos de contorno, dichas funciones se presentan de manera implícita. Sin embargo, para todos los análisis mencionados conviene que se hagan explícitas con el fin de reducir el costo de computación. Por ello es importante aproximarlas lo mejor posible por una técnica que requiera un mínimo esfuerzo de cálculo. En este trabajo se propone lograr este propósito por medio la técnica conocida como máquinas de soporte vectorial, desarrollada en las áreas de reconocimiento de patrones y de aprendizaje estadístico, que ha encontrado gran aplicación en el análisis de imágenes y de señales. Para ello se ha desarrollado un algoritmo que permite su aplicación de excelente precisión con un número reducido de llamadas del algoritmo de cálculo estructural. Esto tiene su fundamento en la flexibilidad y la insensibilidad a la dimensionalidad del problema que caracterizan a dichas máquinas de aprendizaje.

Palabras clave:

Optimización, confiabilidad estructural, estabilidad dinámica, aprendizaje estadístico, reconocimiento de patrones, simulación de Monte Carlo.

APPROXIMATION OF IMPLICIT DECISION FUNCTIONS WITH SUPPORT VECTOR MACHINES

Summary

In mechanical and structural systems the tasks of reliability analysis, dynamic stability assessment and optimization require the knowledge of decision functions, which are usually nonlinear. These functions separate the feasible and unfeasible regions in the optimization case, the safe and unsafe ones in the reliability analysis and the stable and unstable domains in the dynamic stability problem. When using numerical codes such as those based on finite or boundary elements, these functions are implicit. However, for those tasks their explicit knowledge would imply important savings of the computation efforts. In this work it is proposed to render explicit the decision functions by means of a technique known as support vector machines, which has been developed in the fields of pattern recognition and statistical learning and which has found wide application for image and signal processing. To this purpose an algorithm allowing their application in this context with a reduced number of solver calls and excellent accuracy has been developed. This latter is due to the flexibility and the dimensionality independence of the learning machines.

Keywords:

Optimization, structural reliability, dynamic stability, statistical learning, pattern recognition, Monte Carlo simulation.

INTRODUCCIÓN

El análisis de confiabilidad de sistemas estructurales y mecánicos se formula generalmente como la estimación de la probabilidad de que el sistema sobrepase un cierto umbral considerado como crítico, bajo la perturbación de algunas variables aleatorias¹. La expresión matemática de esta definición es

$$P_f = \int_{g(\mathbf{x}) \leq 0} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (1)$$

donde P_f es la probabilidad de fallo, \mathbf{x} es el vector de variables aleatorias (llamadas generalmente variables básicas) que hacen incierta la respuesta del sistema, $p_{\mathbf{x}}(\mathbf{x})$ su densidad de probabilidad conjunta y $g(\mathbf{x})$ la función que define el estado límite, es decir, el umbral que separa las regiones segura e insegura (Figura 1). En muchos casos el problema se formula con respecto a más de un estado límite debido a que el fallo puede ocurrir por múltiples causas que actúan conjuntamente. Para ello se necesita calcular la confiabilidad con respecto a varias funciones de decisión $g_1(\mathbf{x}), g_2(\mathbf{x}), \dots$, etc. Solamente en casos simples, tales como elementos estructurales prismáticos, las funciones $g_j(\mathbf{x}), j = 1, 2, \dots, m$ se encuentran definidas de manera explícita en términos de las variables \mathbf{x} . En el caso general de una estructura compleja, sin embargo, los valores de las funciones en diversos puntos solamente pueden obtenerse por medio de un método numérico de alto costo de computación, tal como el método de elementos finitos. En tales casos, el conocimiento previo de la función de estado límite permite estimar la probabilidad de fallo, bien por métodos de segundo orden o bien por simulación de Monte Carlo, la cual se facilita enormemente al no requerirse más llamadas del programa numérico de la estructura. Comoquiera que esta última técnica es la adecuada en el caso en que la función de decisión es implícita, se ha propuesto el uso del método clásico de superficie de respuesta^{2,3,4} o de redes neuronales^{5,6,7} para aproximar la función con técnicas de regresión, aproximación que es en lo sucesivo usada como sustituto del algoritmo de cálculo estructural. La diferencia de este enfoque con el propuesto en este trabajo es que la función de decisión se aproxima con técnicas de clasificación en lugar de regresión, ya que, en última instancia, el uso de la aproximación es siempre como clasificador.

Otro problema estructural en el que se presentan funciones implícitas de decisión surge en la optimización condicionada⁸. En efecto, la formulación de este problema es

$$\begin{aligned} &\text{minimizar} && J(\mathbf{x}) \\ &\text{sujeto a} && g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m \end{aligned} \quad (2)$$

En esta ecuación \mathbf{x} es el vector de las, así llamadas, variables de diseño, $J(\mathbf{x})$ es la función de costo (normalmente definida en términos de peso de la estructura) y $g_j(\mathbf{x})$ son funciones, generalmente lineales, de algunas respuestas estructurales, que obviamente dependen de tales variables. Al igual que en el caso de confiabilidad, cada una de estas funciones condicionantes del problema de optimización divide el espacio de variables en dos regiones, en las que se dan respectivamente soluciones viables o inviables (Figura 2). La minimización condicionada se puede transformar en una incondicionada si se conocen las funciones de restricción, lo cual muestra claramente la importancia de hacer explícitas dichas funciones de manera previa a la minimización de la función de costo o simultáneamente con ella. Alternativamente, el uso de algunos métodos como las funciones de penalización o el de direcciones factibles⁸ exigen dicho conocimiento explícito. El método de superficie de respuesta también ha sido usado para aproximar las funciones de restricción.

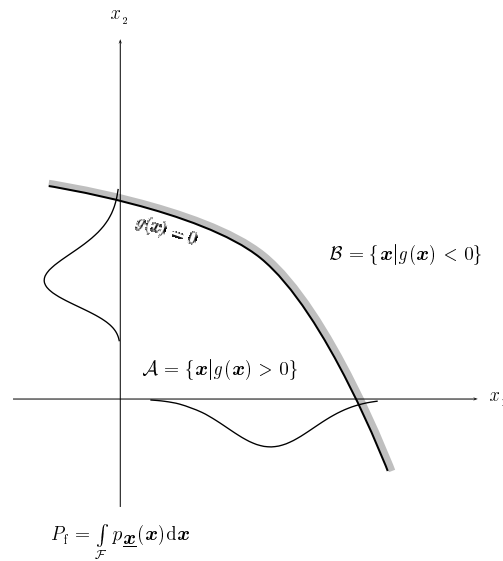


Figura 1. Problema de confiabilidad. La función $g(\mathbf{x}) = 0$ separa los dominios de seguridad y de fallo

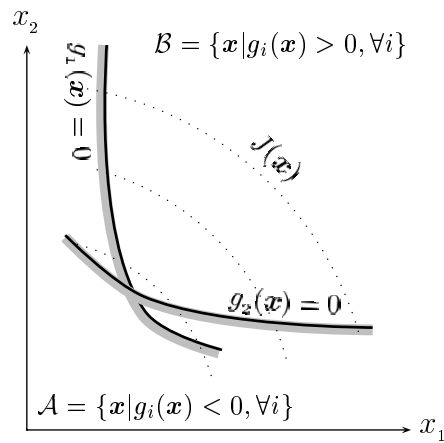


Figura 2. Problema de optimización. Las funciones de restricción $g_i(\mathbf{x}) = 0$ acotan el dominio de diseños viables

Si se va más lejos, se puede formular una optimización basada en confiabilidad en la forma⁹

$$\begin{aligned} &\text{minimizar} && J(\mathbf{x}) \\ &\text{sujeto a} && P[f_j(\mathbf{x}, \mathbf{y}) \leq 0] \leq P_j, \quad j = 1, 2, \dots, m \end{aligned} \tag{3}$$

donde $P[A]$ es la probabilidad del suceso aleatorio A y las P_j son sus cotas superiores especificadas. En este caso se trata de funciones de decisión que a su vez dependen de otras funciones de decisión.

Finalmente, otro problema estructural en el que se hace necesario hacer explícitas funciones de decisión es el de estabilidad estocástica ante cargas dinámicas aleatorias¹⁰. No obstante su complejidad, este problema es de gran importancia práctica, toda vez que tales cargas pueden causar fuertes vibraciones y, posiblemente, colapso, como se dio en los famosos casos de los puentes de Brighton Chain Pier (1836) y Tacoma Narrows (1940), entre otros¹¹. La formulación de este complejo problema parte de la modelación de la dinámica de la estructura como un sistema de ecuaciones diferenciales estocásticas¹²

$$d\mathbf{q}(t) = \boldsymbol{\theta}(\mathbf{q}(t), t)dt + \boldsymbol{\psi}(\mathbf{q}(t), t)d\mathbf{b}(t) \quad (4)$$

donde $\mathbf{q}(t)$ es el vector de estado de dimensión N , $\boldsymbol{\theta}$ es un vector n -dimensional de funciones de deriva, $\boldsymbol{\psi}$ una matriz de difusión de orden $N \times R$ y $\mathbf{b}(t)$ un vector de procesos incorrelacionados de Wiener. El recurso a la fórmula de Ito¹³, la aplicación del operador de valor esperado y, finalmente, la consideración de la teoría de estabilidad de Lyapunov¹⁴, conducen a que la estructura es estable en momentos estadísticos de orden J si todos los autovalores λ_i de las matrices \mathbf{A}_J en el sistema de ecuaciones diferenciales ordinarias

$$\dot{\boldsymbol{\mu}}_J = \mathbf{A}_J \boldsymbol{\mu}_J, \quad J = 1, 2, \dots \quad (5)$$

tienen parte real negativa. Esto implica que existe una función implícita del vector de variables \mathbf{x} que controlan los autovalores λ_i de la matriz, tal que establece una separación de las regiones de estabilidad e inestabilidad dinámica de la estructura (Figura 3). La complejidad computacional de este problema reside en que a) el tamaño de la matriz \mathbf{A}_J crece de forma factorial con el número de grados de libertad y el orden de los momentos estadísticos; b) en que se han de calcular todos sus autovalores con el fin de establecer la estabilidad de la estructura. Debido al tamaño de la matriz \mathbf{A}_J , la función que separa los dominios $\{\Re(\lambda_i) < 0\}$ y $\{\Re(\lambda_i) > 0\}$ se encuentra dada de manera implícita, salvo en casos muy simples¹⁰. Normalmente, para el cálculo de la función límite se procede por prueba y error^{15,16}, lo cual evidentemente es costoso en términos de computación.

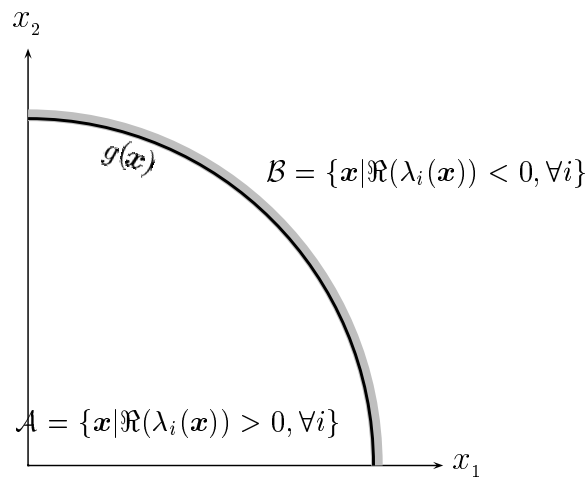


Figura 3. Problema de estabilidad dinámica. En el dominio de estabilidad la parte real de todos los autovalores de la matriz de momentos estadísticos es positiva

En síntesis, funciones implícitas de decisión surgen en tres importantes áreas del análisis estructural, a saber, confiabilidad, optimización y estabilidad dinámica. Con el fin de facilitar los análisis que se realizan en estos ámbitos, resulta altamente conveniente disponer de una aproximación de estas funciones de manera explícita, de tal forma que se pueda evitar el recurso a los programas que calculan sus valores, cuales son las técnicas de elementos finitos en los dos primeros casos y un programa de cálculo de todo el espectro de la matriz de estado en el último problema.

En todos estos casos, la función $g(\mathbf{x}) = 0$ divide el espacio \mathbf{x} en dos dominios diferenciados por el signo de una función. Esto sugiere el tratamiento de este problema con técnicas de clasificación de patrones. Ésta es una rama de la estadística que busca asignar una muestra a una clase determinada con base en unas muestras disponibles, ya clasificadas, de manera que el error en la asignación sea el menor posible¹⁷. En nuestro caso, las clases están dadas por los dominios en que queda dividido el espacio de variables de diseño y su etiqueta es justamente el signo de la función subyacente $g(\mathbf{x})$ a uno y otro lado. Por otra parte, las muestras disponibles (o muestras de entrenamiento) pueden ser generadas por un proceso de simulación determinista o aleatoria.

En general, hay dos diferencias relevantes entre los problemas estructurales en cuestión y los problemas típicos en los que se aplican técnicas de reconocimiento de patrones. Ellas son:

- Las muestras con las que se trabaja en análisis de objetos, imágenes y señales son obtenidas de la naturaleza y, por tanto, acarrean ruido. Esto hace que la separación entre las clases no siempre es perfecta. Esto se ilustra en la Figura 4, que muestra dos variables geométricas de las hojas de las plantas *Iris setosa*, *Iris virginica* e *Iris versicolor*, que constituyen una ilustre base de datos sobre la cual se prueban métodos de reconocimiento de patrones. Es evidente que mientras una de las especies es claramente distinta de las otras dos, el trazado de una frontera entre estas últimas es difícil debido a la presencia de muestras de una clase en medio de la otra. Por el contrario, las clases en los problemas estructurales mencionados, aunque juntas a nivel infinitesimal, son perfectamente separables, debido a que están determinadas por una función continua $g(\mathbf{x})$.

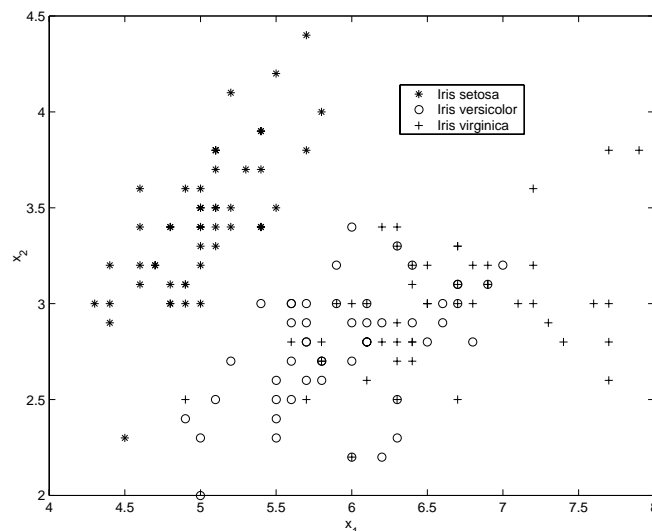


Figura 4. Problema de reconocimiento de patrones: Clasificación de especies de una planta

- Las muestras en los problemas estructurales que nos ocupan pueden ser generadas sintéticamente, mientras que esto no se da en el caso de muestras naturales. Esta posibilidad debe ser aprovechada para buscar un algoritmo que permita reducir al mínimo el número de muestras de entrenamiento, ya que éstas son obtenidas a partir del método numérico de solución del problema estructural.

Tradicionalmente, el problema de clasificación se ha tratado con técnicas clásicas, como la discriminación de Fisher o la estadística bayesiana¹⁷. Sin embargo, el desarrollo en las últimas décadas de la, así llamada, inteligencia artificial o, de manera más cauta, computación blanda¹⁸, ha permitido incorporar algunos instrumentos de esta tecnología, principalmente las redes neuronales artificiales, al tratamiento de diversos problemas estadísticos, entre los que se encuentran los de regresión y de clasificación, abarcados por el ya mencionado dominio de reconocimiento de patrones^{19,20}. Paralelamente, se ha dado el desarrollo de la teoría de aprendizaje estadístico, principalmente en Rusia desde los años setenta^{21,22}. Esta vertiente ha confluído con la de investigación en redes neuronales y las técnicas clásicas para dar como resultado un tratamiento unificado de estos métodos desde la perspectiva pura del aprendizaje estadístico²³.

Dejando a un lado el criterio de clasificación de Fisher, que es puramente lineal, la clasificación de patrones ha sido dominada fuertemente por el método bayesiano, el cual, para un problema de dos clases, consiste simplemente en asignar el vector \mathbf{x} a la clase que presente la mayor probabilidad posterior $P(\mathcal{C}_k|\mathbf{x})$, $k = 1, 2, \dots$. Esto equivale a asignar la muestra a la clase k , en contra de la clase j si²⁰

$$p(\mathbf{x}|\mathcal{C}_k)P(\mathcal{C}_k) > p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j) \quad (6)$$

donde $P(\mathcal{C}_k)$ es la probabilidad de que una muestra pertenezca a la clase \mathcal{C}_k y $p(\mathbf{x}|\mathcal{C}_k)$ la densidad de probabilidad de \mathbf{x} dado que este vector pertenece a la clase \mathcal{C}_k . Estos términos se pueden calcular a partir de una población abundante de muestras que, para nuestro caso, resulta altamente costosa. Por el contrario, las técnicas flexibles de aprendizaje estadístico (a saber, redes neuronales y máquinas de soporte vectorial, entre otras) permiten obtener una función de clasificación con un número mínimo de muestras, ya que, en el lenguaje de la estadística, son no paramétricas, es decir, no requieren la estimación de funciones de densidad de probabilidad como en la técnica bayesiana. Por otra parte, su adaptación a contornos fuertemente no lineales está garantizada por su naturaleza flexible y adaptable. Estos adjetivos se explican bajo la consideración de su modelo general, cual es

$$g(\mathbf{x}) = \sum_i w_i h(\mathbf{x}_i, \mathbf{x}, \boldsymbol{\omega}) \quad (7)$$

donde \mathbf{x}_i son los vectores de entrenamiento, w_i parámetros de ponderación, $\boldsymbol{\omega}$ un conjunto de parámetros y $h(\cdot)$ una función no lineal. La flexibilidad para adaptarse a contornos complejos se obtiene si la función $h(\cdot)$ (o su derivada, como en el caso de la función sigmoide) tiene un soporte eficiente reducido, mientras que la adaptación a las muestras dadas se consigue haciendo la función dependiente de ellas, como se indica. Por el contrario, un método como el de superficie de respuesta, ampliamente usado en confiabilidad y optimización como ya se ha dicho, opera sobre modelos del tipo

$$\sum_i w_i h_i(\mathbf{x}) \quad (8)$$

en los que las funciones $h_i(\mathbf{x})$ son generalmente potencias de órdenes uno y dos, por lo cual tienen soporte eficaz infinito y, además, no son adaptables a las muestras dadas, ya que no están indexadas por ellas. Por este motivo, su aplicación para el problema que nos ocupa es limitada.

En este trabajo se propone tratar el problema de hacer explícitas las funciones de decisión como un problema de reconocimiento de patrones con técnicas flexibles de aprendizaje a partir de muestras. La teoría sobre tales técnicas, principalmente redes neuronales y máquinas de soporte vectorial, favorece a éstas últimas para su aplicación en asociación con evaluaciones costosas de las muestras de entrenamiento, tales como las que surgen por medio de la aplicación del método de elementos finitos u otra técnica similar. Esto se debe a que las máquinas de soporte vectorial comprenden un elemento adicional, a saber los márgenes del clasificador, lo cual permite reducir el espacio de búsqueda, como se indica más adelante. La selección de esta técnica también se sustenta en otras razones matemáticas que se enuncian más adelante.

En la sección siguiente se hace un resumen del método de aprendizaje de soporte vectorial. Luego se describe el algoritmo de cálculo desarrollado para el cálculo de las funciones de decisión, las potencialidades del cual se ilustran con algunos ejemplos. Finalmente, se enuncian unas conclusiones y unas líneas de desarrollos futuros.

CLASIFICACIÓN POR MEDIO DE MÁQUINAS DE SOPORTE VECTORIAL

Como se ha dicho antes, actualmente se explora la solución de algunos problemas clásicos de la estadística por medio de la teoría de aprendizaje estadístico y uno de sus instrumentos principales, las máquinas de soporte vectorial. Los más importantes entre tales problemas son: clasificación y reconocimiento de patrones, regresión y estimación de la densidad de probabilidad. En esta sección se resumen los elementos esenciales de esta técnica antes de abordar su aplicación a la estimación de funciones de decisión en ingeniería de estructuras.

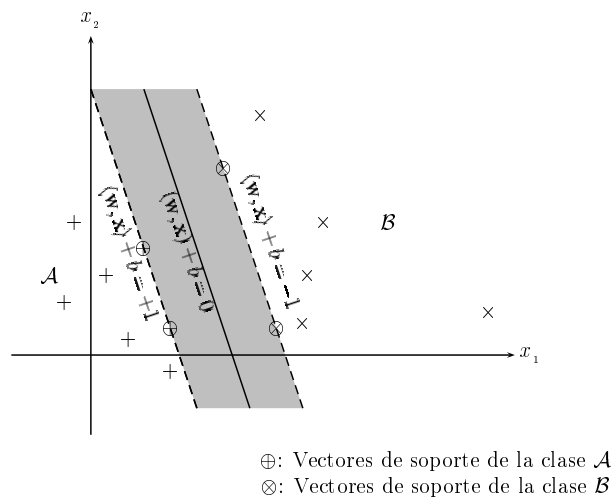


Figura 5. Hiperplano de separación, márgenes y vectores de soporte

Clases linealmente separables

Supongamos que tenemos un conjunto $\{\mathbf{x}_i, c_i\} \in \mathbb{R}^m \times \{-1, 1\}$, $i = 1, 2, \dots, n$ que puede ser separado linealmente, es decir, por medio de un hiperplano (Figura 5). El propósito es

usar este conjunto para calcular los parámetros óptimos del hiperplano, en el sentido en que se definirá más adelante. Por tanto, este conjunto se denominará conjunto de entrenamiento, en analogía con el lenguaje usado en la literatura sobre redes neuronales. El hiperplano está definido por

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \quad (9)$$

donde $\langle \cdot, \cdot \rangle$ denota el producto escalar, \mathbf{w} es un conjunto de pesos que define el vector normal al hiperplano y b es una constante, que corresponde al llamado intercepto en el análisis de regresión o sesgo o umbral en la literatura neuronal. Se buscará un hiperplano que haga máxima la distancia a los vectores de entrenamiento (esto es, que guarde la máxima separación posible con respecto a ambas clases simultáneamente), lo que equivale a resolver el siguiente problema de valores extremos:

$$\max_{\mathbf{w}, b} \min_i \{ \|\mathbf{x} - \mathbf{x}_i\| : \mathbf{x} \in \mathbb{R}^m, \langle \mathbf{w}, \mathbf{x} \rangle + b = 0, i = 1, 2, \dots, n \} \quad (10)$$

Para cada vector de datos \mathbf{x}_i , definiremos un funcional de margen γ como

$$\gamma_i = c_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \quad (11)$$

Nótese que $\gamma_i > 0$ indica una clasificación correcta. Para fines de cálculo e interpretación es conveniente normalizar los pesos y el umbral por el factor $1/\|\mathbf{w}\|$, de manera que el margen óptimo deviene en

$$\gamma = \frac{2}{\|\mathbf{w}\|} \quad (12)$$

como se puede demostrar fácilmente. Esta normalización implica que los puntos más cercanos al hiperplano satisfacen $|\langle \mathbf{w}, \mathbf{x} \rangle + b| = 1$, como se muestra en la figura. Estas ecuaciones definen, entonces, el siguiente problema de optimización:

$$\begin{aligned} &\text{Minimizar } \Omega(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} \\ &\text{sujeto a: } c_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, i = 1, 2, \dots, n. \end{aligned}$$

Al usar multiplicadores de Lagrange $\alpha_i \geq 0$, surge la siguiente función de coste

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [c_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1] \quad (13)$$

Por tanto, el problema consiste ahora en minimizar la función con respecto a las variables primarias (\mathbf{w}, b) y maximizarla con respecto a los multiplicadores de Lagrange. Esto significa que se debe hallar un punto de ensilladura. Al hacer estacionaria la derivada

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} \quad (14)$$

se encuentra que

$$\mathbf{w} = \sum_{i=1}^n \alpha_i c_i \mathbf{x}_i \quad (15)$$

mientras que la derivación con respecto al umbral da como resultado

$$\sum_{i=1}^n \alpha_i c_i = 0 \quad (16)$$

Al tomar en cuenta la condición $\alpha_i \geq 0$, esto significa que el vector de pesos se puede expandir solamente en términos de los patrones que tienen multiplicadores de Lagrange positivos, de suerte que los restantes ya no son necesarios. Tales patrones se denominan vectores de soporte, ya que son los que sustentan la expansión de aquí en adelante. Al considerar la bien conocida ecuación de Karush–Kuhn–Tucker, llamada condición complementaria de optimización²⁴

$$\alpha_i [c_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1] = 0, i = 1, 2, \dots, n \quad (17)$$

se concluye que los vectores de soporte están justo sobre las líneas del margen definido anteriormente. Por tanto, la ecuación (15) se puede poner en la forma más simple

$$\mathbf{w} = \sum_{j=1}^{\ell} \alpha_j c_j \mathbf{x}_j \quad (18)$$

donde ℓ es el número de vectores de soporte, que es mucho menor que el número de muestras disponibles n , pues son solamente las que yacen en el margen. Esto significa que el hiperplano se define sólo en términos de los puntos más cercanos a él, lo que es intuitivamente sugestivo.

Al sustituir las ecuaciones (15) y (16) en (13), se obtiene el siguiente problema cuadrático en términos de las variables duales:

Maximizar

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i c_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle c_j \alpha_j \quad (19)$$

sujeto a:

$$\begin{aligned} \alpha_i &\geq 0 \\ \sum_{i=1}^n \alpha_i c_i &= 0 \end{aligned} \quad (20)$$

Al resolver este problema de optimización, las ecuaciones (17) y (18) dan los valores finales de las variables primarias. Esto representa el final del entrenamiento de las máquinas de

soporte vectorial, la cual puede ser eventualmente materializada como un chip de manera similar a las redes neuronales. Una vez entrenada, la máquina puede ser usada para clasificar un nuevo vector \mathbf{x} en las clases $\{-1, +1\}$ por medio de

$$c(\mathbf{x}) = \text{sgn}(g(\mathbf{x})) = \text{sgn}\left(\sum_{i=1}^{\ell} \alpha_i c_i (\langle \mathbf{x}_i, \mathbf{x} \rangle) + b\right) \quad (21)$$

Separación no lineal por medio de núcleos

En la práctica, la anterior separación por medio de hiperplanos está limitada por el hecho de que muchas clases exhiben una frontera no lineal, lo que hace necesario buscar un criterio más adecuado que el ofrecido por la ecuación (9). Sin embargo, es deseable mantener las ventajas matemáticas de una forma lineal, como la que se da en dicha ecuación, pues, si bien es posible realizar una clasificación correcta con base en las muestras cercanas a a la frontera por medio de redes neuronales⁶, la fuerte no linealidad de la función de coste implicada en este algoritmo de aprendizaje conlleva a que la minimización puede quedar atrapada en algún mínimo local diferente en cada intento¹⁹. Esta dificultad se evita con optimización de funciones lineales o cuadráticas, como la que surge en este caso [ec. (13)]. Por otra parte, está demostrado que la probabilidad de error con clasificación por hiperplanos disminuye con la relación n/m ²⁵, lo cual indica que para un número de muestras n es positivo aumentar la dimensionalidad m del problema.

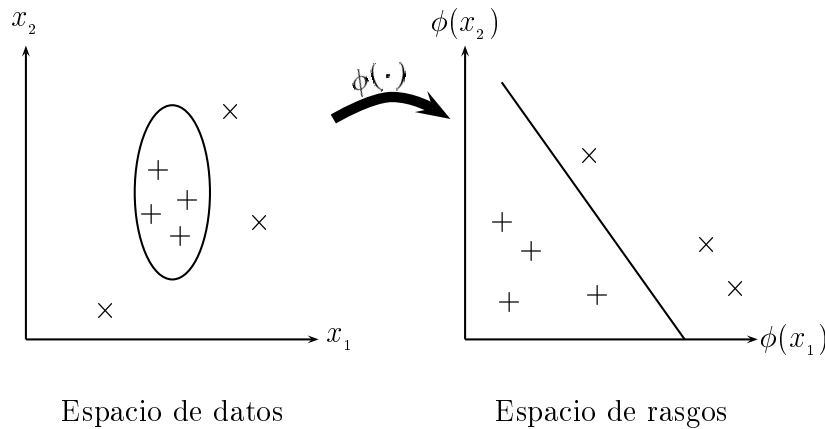


Figura 6. Mapeo no lineal para formación de núcleos de máquinas de soporte vectorial

Todo lo anterior sugiere hacer el mapeo $\phi : \mathbb{R}^m \mapsto \mathbb{R}$, ilustrado en la Figura 6. El espacio \mathcal{R} se denomina el espacio de rasgos, el cual tiene una dimensión mayor que el espacio básico. En efecto, sea \mathbf{x} un vector en un espacio de m coordenadas (x_1, x_2, \dots, x_m) . La proyección al espacio de rasgos con un mapeo polinomial de grado dos, por ejemplo, da origen a un espacio de $M = m(m+3)/2$ coordenadas²²

$$\begin{aligned} n \text{ coordenadas} &: z_1 = x_1, \dots, z_n = x_n \\ n \text{ coordenadas} &: z_{n+1} = x_1^2, \dots, z_{2n} = x_n^2 \\ \frac{n(n-1)}{2} \text{ coordenadas} &: z_{2n+1} = x_1 x_2, \dots, z_M = x_{n-1} x_n \end{aligned} \quad (22)$$

Para el mapeo $\phi : \mathbb{R}^m \mapsto \mathbb{R}$ la ecuación (21) toma la forma

$$c(\mathbf{x}) = \text{sgn}(g(\mathbf{x})) = \text{sgn}\left(\sum_{i=1}^{\ell} \alpha_i c_i (\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle) + b\right) \quad (23)$$

Puede verse que el objetivo es hallar una función $\phi(\cdot)$, tal que los datos transformados sean linealmente separables. El hecho crucial es que esta búsqueda no es necesaria, dado que no se requiere conocer la transformación como tal sino solamente su representación en forma de producto interno

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \quad (24)$$

tal como indica la ecuación (23). Esto quiere decir que el núcleo $K(\cdot, \cdot)$ realiza un mapeo implícito que no es necesario explicitar. Este funcional simétrico puede ser construido bajo la consideración del teorema de Mercer sobre núcleos definidos positivos²⁶. El teorema establece que si $K(\cdot, \cdot)$ es un núcleo continuo de un operador integral positivo T , es decir,

$$\begin{aligned} (Tf)(\mathbf{y}) &= \int K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) d\mathbf{x}, \\ \int \int K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} &\geq 0 \end{aligned} \quad (25)$$

entonces puede ser expandido en series convergentes de autofunciones $\psi(\cdot)$ del operador, correspondientes a los J autovalores positivos λ_j :

$$K(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^J \lambda_j \psi_j(\mathbf{x}) \cdot \psi_j(\mathbf{y}) \quad (26)$$

En la práctica, en virtud del llamado teorema de Moore–Aronszajn, la positividad del núcleo puede ser verificada por la definición positiva de la matriz asociada \mathbf{K} , obtenida por discretización del núcleo sobre una rejilla cuadrada de puntos²⁷.

Resulta fácil demostrar que el espacio de rasgos puede ser representado como un Espacio de Hilbert Reprodutor de Núcleos (RKHS, por sus siglas en inglés) \mathcal{H} , es decir, que para una función

$$f(\mathbf{x}) = \sum_{j=1}^{\infty} \psi_j \phi_j(\mathbf{x}) \quad (27)$$

su producto interno con el núcleo la reproduce

$$\langle f(\mathbf{x}), K(\mathbf{x}, \mathbf{y}) \rangle_{\mathcal{H}} = f(\mathbf{y}) \quad (28)$$

En esta ecuación el producto interno se define como

$$\left\langle \sum_{j=1}^{\infty} \psi_j \phi_j(\mathbf{x}), \sum_{j=1}^{\infty} \psi'_j \phi_j(\mathbf{x}) \right\rangle_{\mathcal{H}} \equiv \sum_{j=1}^{\infty} \frac{\psi_j \psi'_j}{\lambda_j} \quad (29)$$

La consecuencia más importante de construir núcleos en esta forma es que el espacio de Hilbert \mathcal{H} no contiene funciones que no puedan ser representadas en la forma dual dada por la ecuación (27)^{27,28}. Para los fines de este trabajo, es conveniente reescribir la ecuación (23) en función de los núcleos

$$c(\mathbf{x}) = \text{sgn}(g(\mathbf{x})) = \text{sgn} \left(\sum_{i=1}^{\ell} w_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (30)$$

donde $w_i = \alpha_i c_i$. Con esta transformación, la función $g(\mathbf{x})$ toma una forma idéntica a la empleada en las redes neuronales de base radial.

Los siguientes son algunos de los núcleos más estudiados y aplicados en el dominio de la teoría de aprendizaje estadístico:

Núcleo de funciones de base radial:	$K(\mathbf{x}, \mathbf{y}) = \exp \left(-\zeta \ \mathbf{x} - \mathbf{y}\ ^2 \right)$	
Núcleo polinomial:	$K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$	(31)
Núcleo de Fourier:	$K(\mathbf{x}, \mathbf{y}) = \frac{(m+1/2)(\mathbf{x}-\mathbf{y})}{2 \sin((\mathbf{x}-\mathbf{y})/2)}$	
Núcleo sigmoidal:	$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \langle \mathbf{x}, \mathbf{y} \rangle + \theta)$	

Finalmente, huelga decir que la calidad de la aproximación del clasificador con la ecuación (30) está determinada a) por el número de muestras usadas en el entrenamiento, especialmente por las más cercanas a la frontera, como se desprende de la noción de vectores de soporte; y b) por una propiedad de la función de aproximación llamada dimensión de Vapnik–Chervonenkis²². Como una descripción, así sea somera, de esta dimensión está más allá del alcance de este artículo, baste con decir que la reducción del error de clasificación de muestras no usadas en el entrenamiento (error de generalización) puede controlarse solamente con esta dimensión, variando los parámetros del núcleo, sin necesidad de aumentar el número de muestras de entrenamiento. Algo similar se puede lograr por medio de la clasificación con redes neuronales, aunque en menor medida^{25,29}. En última instancia, la preeminencia de la dimensión de Vapnik–Chervonenkis conduce a una superación del llamado anatema de la dimensionalidad en las máquinas de soporte vectorial. Para el problema de clasificación que nos ocupa, este anatema consiste en que el número necesario de muestras de entrenamiento crece exponencialmente con el número de dimensiones del problema¹⁹. La relativa independencia de las máquinas de soporte vectorial de la dimensionalidad del problema hace que esta técnica sea preferible incluso sobre las redes neuronales para la aproximación de funciones de decisión complicadas en las que cada muestra tenga un costo computacional elevado.

ALGORITMO

Con base en las características de las máquinas de soporte vectorial, se ha desarrollado un algoritmo para hacer mínimo el número de muestras de entrenamiento para aproximar la función de decisión. Sus elementos esenciales son como sigue: a una máquina de soporte vectorial predefinida se le enseña un conjunto de datos sin clasificar (es decir, sin precisarle si pertenecen a la clase \mathcal{A} o \mathcal{B}) para que ella los clasifique. Comoquiera que la rotulación de las muestras tiene un costo computacional alto, es deseable que este conjunto sea pequeño, justamente para lo cual es óptima la clasificación con tales máquinas, como se dijo al final de la sección anterior. Ahora bien, en dependencia del rótulo del punto y de su localización se puede extraer información nueva de él o no. De hecho, rotular un punto que yaza dentro del margen siempre reporta un aumento de la información, independientemente del rótulo que reciba. Por el contrario, la evaluación de un punto que se encuentre más allá de los márgenes no reporta nueva información, a menos que haya sido clasificado incorrectamente por una versión anterior de la máquina de soporte vectorial.

El algoritmo aparece en la Figura 7. Su explicación es la siguiente: El banco de datos $D(\cdot)$ contiene muestras de las variables básicas, las cuales se pueden generar a partir de distribuciones uniformes independientes. Ahora bien, estas muestras del banco de datos no son dadas al algoritmo de cálculo estructural; solamente se escogen inicialmente dos grupos pequeños de muestras (digamos, de uno a tres), pertenecientes respectivamente a las zonas \mathcal{A} y \mathcal{B} , selección que siempre se puede hacer sobre criterios de ingeniería. De hecho, basta con escoger cuantificadores vectoriales que correspondan a condiciones conservadoras y liberales, respectivamente. Con estos puntos se entrena una máquina de soporte vectorial inicial.

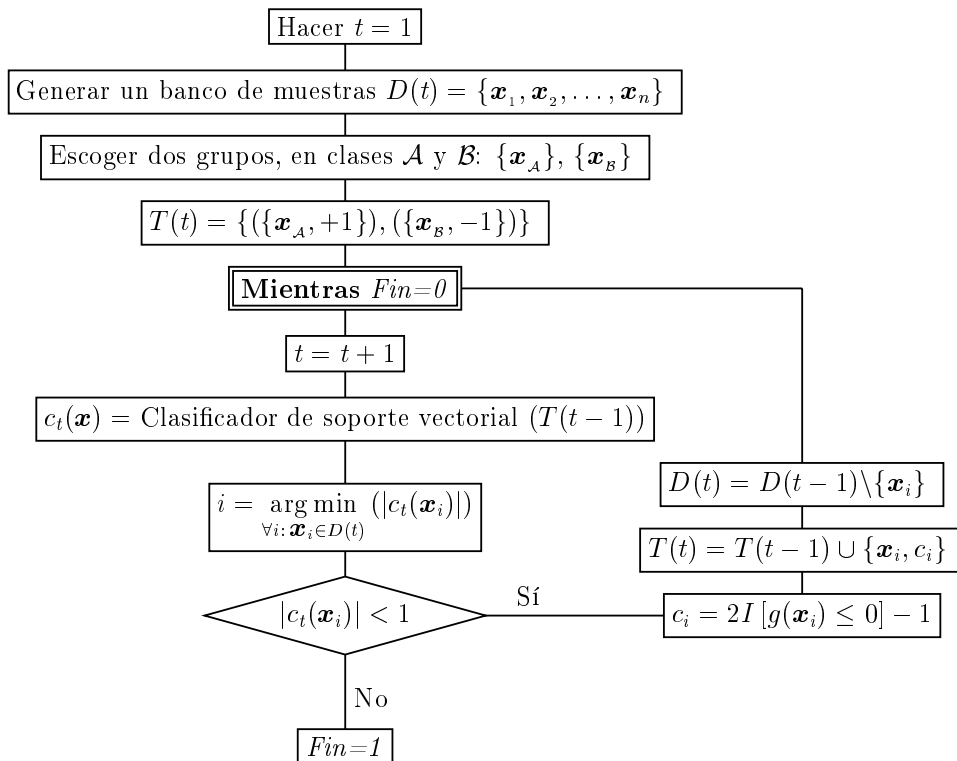


Figura 7. Algoritmo propuesto

En el bucle principal del algoritmo, el conjunto de entrenamiento $T(\cdot)$ se incrementa sucesivamente probando un punto, escogido aleatoriamente, entre los que yacen dentro del margen vigente, al tiempo que se retira del banco de datos. El algoritmo de cálculo estructural es invocado sólo si se cumple la condición $|c_t(\mathbf{x}_i)| < 1$; si no, se descarta la muestra y su correspondiente respuesta estructural no es calculada en este paso (fortuitamente puede serlo en un paso futuro si llega a ser escogida aleatoriamente como las muestras anteriores, pero esto es poco probable, pues los márgenes se hacen cada vez más estrechos). Este criterio, que se basa en la noción misma de los soportes vectoriales, minimiza el número de llamadas del algoritmo estructural. Al proceder así llega un momento t cuando todas las muestras del banco de datos han sido probadas, de forma que éste queda vacío. Entonces termina el entrenamiento. Si el tamaño del banco de datos inicial es suficientemente grande, el valor final del margen será muy pequeño y los vectores de soporte quedarán muy cerca de la función de decisión. Esto significa que el tamaño n del banco de datos controla la precisión del clasificador.

EJEMPLOS NUMÉRICOS

Función bidimensional conocida

Mostraremos en primer lugar las bondades del algoritmo propuesto con respecto a una función bi-dimensional, ya que esto permite ver sus etapas de convergencia. La función

$$g(x_1, x_2) = 4 - 4 \frac{(x_1 - 1)^2}{25} - x_2 = 0 \quad (32)$$

ha sido tomada de la referencia 30. Se escogió un núcleo de funciones de base radial [ec. (31)]. Se generó la población de valores gaussianos en el segundo cuadrante del plano que aparece en la Figura 8 rodeando a la función de decisión; para ello, de una población mayor se seleccionaron los 180 valores más distantes del origen y se clasificaron con la función objetivo. Luego se inició el algoritmo con solamente dos muestras del valor de la función, correspondientes a ambos lados de la frontera. La Figura 8 muestra cuatro instancias de aproximación de la función. Se puede notar que la flexibilidad del núcleo permite aproximar adecuadamente la función de decisión, la cual termina definida con seis vectores de soporte, obtenidos después de 27 llamadas de la función $g(\cdot)$.

Condición en un problema de optimización

Consideraremos ahora una función implícita de múltiples dimensiones, vinculada a un problema de optimización estructural. Con referencia a la armadura mostrada en la Figura 9, supongamos que la optimización de la estructura consiste en hacer mínimo el peso de la estructura bajo la condición de que el máximo desplazamiento vertical no sea superior a $3,5 \times 10^{-5}$ m. Esto implica que las variables de diseño son las nueve áreas de las barras, ya que éstas determinan la matriz de rigidez \mathbf{k} . Se trata entonces de hacer explícita la función $g(\mathbf{x}) = 3,5 \times 10^{-5} - \max[u(\mathbf{x})] = 0$.

Para resolver este problema, se generó un banco de datos de las áreas seccionales compuesto por una matriz de orden 1000×9 por medio de distribuciones uniformes en el rango $[0,001 \ 0,004]m^2$. Para dar una idea de la dificultad técnica del problema, considérese la Figura 10, en la que aparecen las áreas, tomadas del banco de datos, de las barras (5) y (7), las cuales son las que más determinan el desplazamiento vertical del nodo 4, que es donde se espera que suceda el máximo buscado. Se puede ver que, a pesar de haber seleccionado las variables más decisivas, no se ve claramente cómo las clases puedan ser separadas, lo cual hace evidente la necesidad del mapeo no lineal ilustrado por la Figura 6, que caracteriza a las máquinas de soporte vectorial.

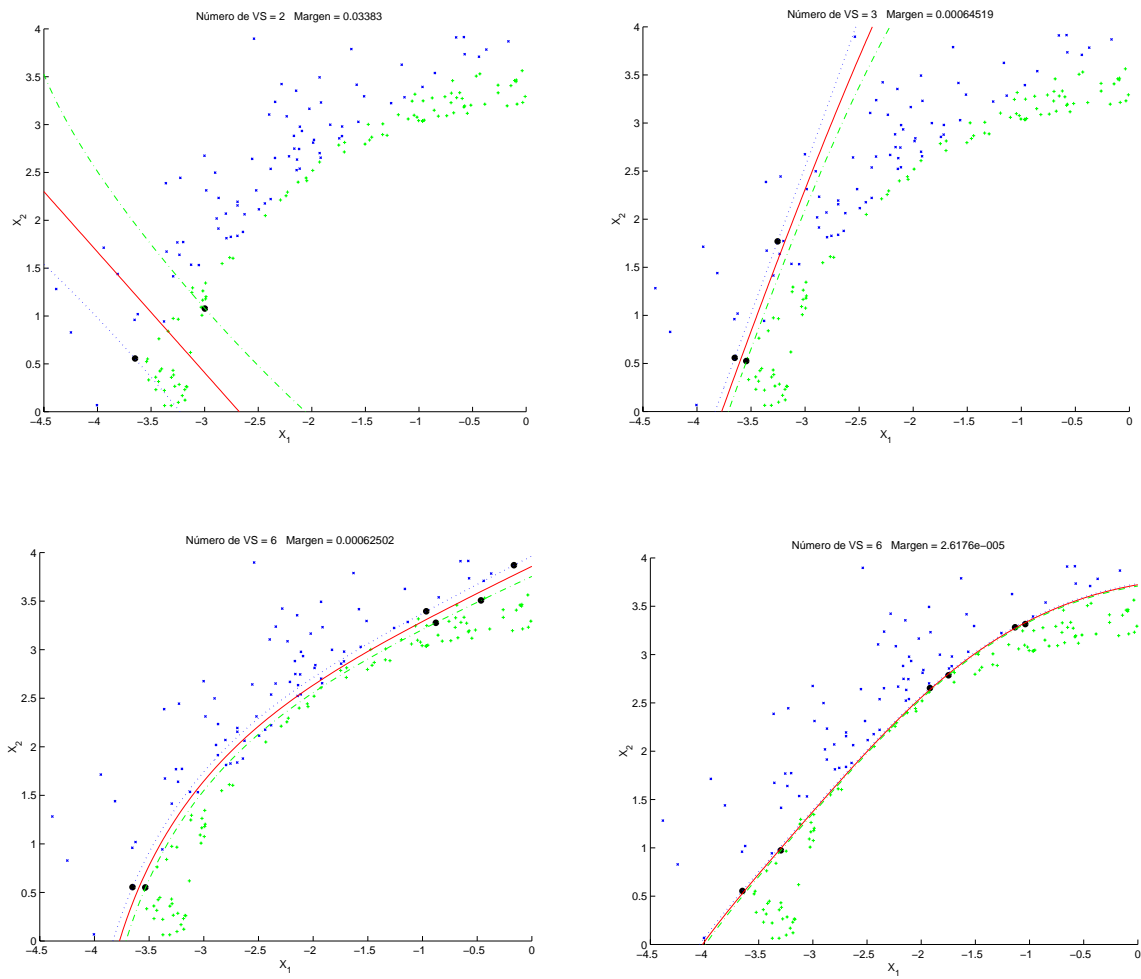


Figura 8. Secuencia de aproximación a la función de decisión

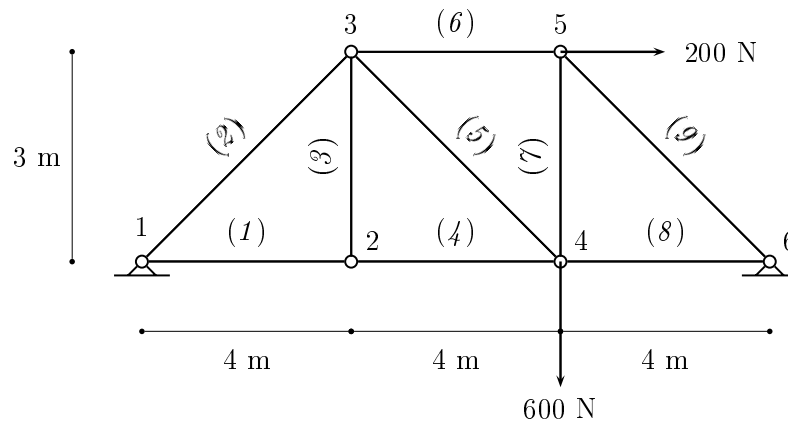


Figura 9. Estructura del ejemplo de optimización

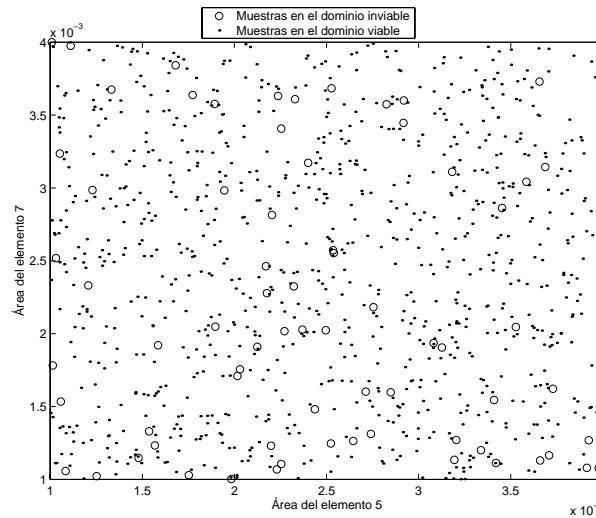


Figura 10. Muestras aleatorias de dos barras en los dominios viable e inviable

Se utilizó un núcleo de base radial con $\zeta = 0,01$. Después de solamente 52 evaluaciones del problema (las cuales van formando de manera creciente la población de entrenamiento de los diferentes clasificadores en las varias iteraciones del algoritmo) se obtuvo una aproximación a la función de decisión consistente en los diez vectores de soporte que aparecen en la Tabla I y un umbral $b = 3264,9$. El tiempo de cálculo en un ordenador personal convencional es inferior a un minuto. Una comparación rápida con el problema anterior muestra que el paso de dos a nueve dimensiones no implica un gran aumento del número de evaluaciones de la función de decisión, lo cual comprueba lo dicho anteriormente sobre la relativa independencia de las máquinas de soporte vectorial de la dimensionalidad del problema.

i	w_i	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}	x_{i7}	x_{i8}	x_{i9}
1	-5,55e+09	1,31	2,46	1,41	2,38	1,05	1,15	3,23	2,60	1,44
2	-9,05e+09	1,68	1,32	1,77	1,07	2,54	3,21	2,55	2,33	1,13
3	-3,55e+09	2,11	1,74	2,96	3,73	1,59	3,99	1,92	1,97	1,11
4	-1,63e+10	3,19	3,03	3,43	2,69	1,03	3,84	2,52	1,31	1,19
5	-4,58e+10	2,34	1,55	1,81	1,40	1,25	1,34	1,02	2,96	1,47
6	-3,59e+10	1,32	3,98	1,66	1,72	3,19	2,45	1,13	1,94	1,07
7	1,11e+10	3,65	1,43	2,81	2,13	2,98	2,75	2,10	1,95	1,13
8	1,14e+10	1,22	1,95	3,18	2,51	3,83	1,87	2,40	2,32	1,21
9	7,82e+10	2,04	2,62	1,67	1,50	1,49	2,37	1,11	2,23	1,37
10	1,55e+10	1,45	3,37	2,27	2,39	1,96	1,60	2,51	3,07	1,15

Tabla I. Vectores de soporte en el ejemplo de optimización (Los valores de x_{ij} aparecen multiplicados por 10^3)

El error de clasificación de las muestras de entrenamiento fue de 1,6 %, mientras que el error calculado sobre un conjunto adicional de validación, igualmente de 1000 muestras, fue de sólo 2,2 %. Esto hace que la estimación de la posición de una muestra en el proceso de optimización dada por esta aproximación sea confiable, más aún si se tiene en cuenta que, debido a la continuidad de la función $g(\mathbf{x})$ que subyace a la función de decisión $g(\mathbf{x}) = 0$, las muestras mal clasificadas necesariamente se encuentran muy cerca del límite aproximado, por lo cual para aplicaciones prácticas este error puede ignorarse.

Problema de confiabilidad estructural

En este ejemplo se calculará la probabilidad de fallo del pórtico sometido a fuerzas horizontales que aparece en la Figura 11. Los datos de las variables aleatorias que componen el vector \boldsymbol{x} figuran en la Tabla II. La función de estado límite es

$$g(\boldsymbol{x}) = 11 - u(\boldsymbol{x}) \quad (33)$$

donde $u(\boldsymbol{x})$ es el desplazamiento del último piso del pórtico medido en centímetros. La probabilidad de fallo estimada con 350 000 muestras de Monte Carlo es $P_f = 5,451 \times 10^{-3}$. Para el cálculo se empleó un núcleo de funciones de base radial. El entrenamiento requirió sólo 146 llamadas del algoritmo de elementos finitos necesario para calcular el pórtico. Una vez entrenado el clasificador de soporte vectorial, se lo utilizó para clasificar las mismas 350 000 muestras empleadas en la simulación de Monte Carlo de referencia, de las cuales solamente cinco fueron clasificadas erróneamente. Por tanto, la probabilidad de fallo estimada fue $\hat{P}_f = 5,457 \times 10^{-3}$, lo cual representa un error de estimación de sólo 0,052 con respecto al resultado de referencia.

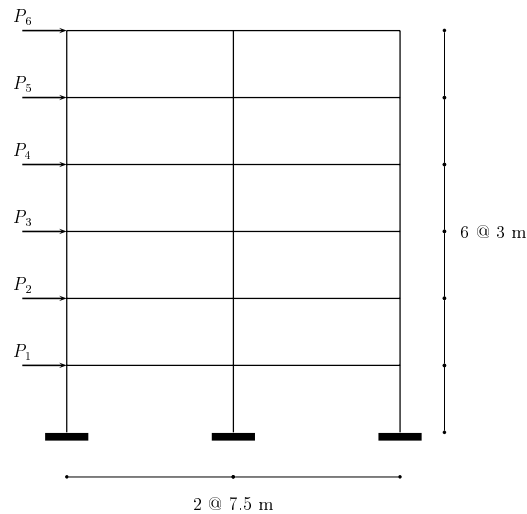


Figura 11. Estructura del ejemplo de confiabilidad

Variable	Distribución	Media	Desviación típica
E de las columnas	lognormal	1,960 kN/cm ²	196 kN/cm ²
E de las vigas	lognormal	1,960 kN/cm ²	196 kN/cm ²
I de las columnas	lognormal	100,000 cm ⁴	10,000 cm ⁴
I de las vigas	lognormal	150,000 cm ⁴	15,000 cm ⁴
P_1	normal	24,50 kN	6,125 kN
P_2	normal	27,44 kN	6,860 kN
P_3	normal	28,42 kN	7,105 kN
P_4	normal	29,40 kN	7,350 kN
P_5	normal	30,38 kN	7,595 kN
P_6	normal	31,36 kN	7,840 kN

Tabla II. Distribuciones de probabilidad de las variables básicas

La Figura 12 muestra la evolución del número de vectores de soporte, mientras que la Figura 13 muestra la del margen. Se puede ver que el número de vectores de soporte tiende a un valor estacionario al tiempo que el margen se estabiliza en un valor ínfimo (obsérvese la escala semi-logarítmica de la Figura 13). Esto indica que el entrenamiento bien podría haberse detenido con un número de llamadas aproximadamente igual a 60 sin menoscabo de la precisión, lo que implica una fuerte reducción del esfuerzo computacional. Obsérvese que esta evolución del margen puede ser seguida en tiempo real.

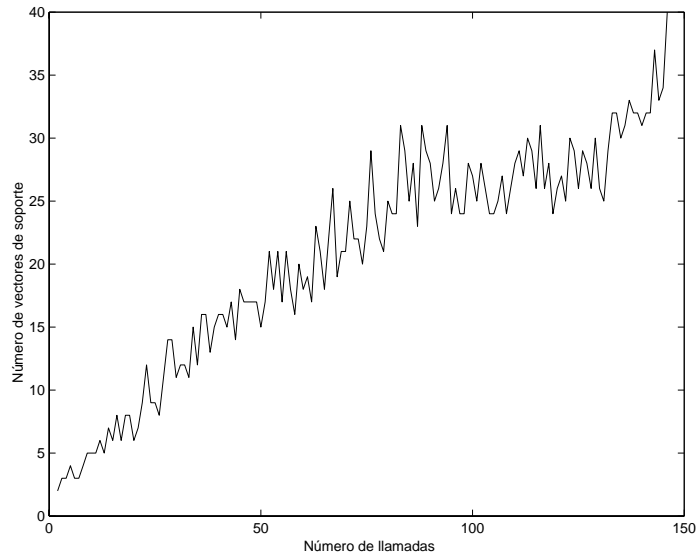


Figura 12. Evolución de los vectores de soporte

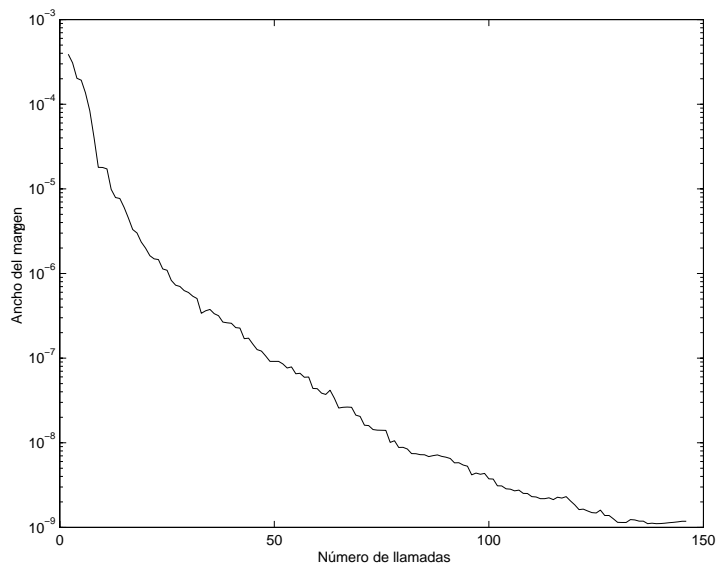


Figura 13. Evolución del margen

CONCLUSIONES

En al menos tres áreas de la ingeniería estructural (optimización, confiabilidad y estabilidad) se presentan funciones de decisión que dividen el espacio de variables en dos regiones claramente diferenciadas. Desafortunadamente, estas funciones se dan normalmente de manera implícita a través de un programa de cálculo numérico, tal como el método de los elementos finitos. Sin embargo, con el fin de reducir el costo de computación de tales problemas es conveniente disponer de una forma explícita. Para ello se ha propuesto en este trabajo tratar el problema como uno de clasificación estadística por medio de la técnica conocida como máquinas de soporte vectorial, las cuales tiene las siguientes prerrogativas: a) No se requiere un conjunto abundante de muestras para su entrenamiento, ya que el clasificador queda definido en términos de las muestras contenidas en un margen; b) Se trata de una técnica no-paramétrica, en el sentido que se le da en Estadística a este término, lo cual hace innecesaria la estimación de las funciones de probabilidad de las muestras en cada clase; c) La aproximación es de tipo flexible, con lo cual se adapta fácilmente a funciones fuertemente no lineales, al igual que las redes neuronales; d) Sobre estas últimas, las máquinas de soporte vectorial tienen la ventaja de ser más independientes de la dimensionalidad del problema, lo cual se traduce en una reducción de la labor computacional. Los ejemplos del artículo demuestran la precisión y la economía del algoritmo desarrollado para aplicar este método de clasificación en el contexto de funciones de decisión en mecánica estructural, en el que las muestras son sintéticas y las clases son perfectamente separables.

En este artículo se ha dado un tratamiento común a los varios tipos de funciones de decisión que surgen en ingeniería estructural. Sin embargo, es lógico que puede ser conveniente modificar el algoritmo para adaptarlo a las condiciones específicas de cada tipo de problema en particular. Por ejemplo, en el caso de confiabilidad estructural, se puede tener en cuenta que las funciones de estado límite normalmente corresponden a situaciones extremas, lo que traducido en términos de teoría de probabilidades corresponde a situaciones en las que las muestras de ambas clases se encuentran más separadas que en situaciones corrientes, si se usa la densidad real de las variables para generar el banco de datos, lo cual es un hecho afortunado que puede facilitar el cálculo de una función de clasificación. Por otra parte, en el caso de optimización, el cálculo del gradiente de la función de costo, requerido en muchos métodos, puede ser aprovechado para generar sub regiones, en las cuales se puede realizar el entrenamiento de un clasificador local. Trabajos en estas direcciones de investigación realizan actualmente los autores.

AGRADECIMIENTOS

Para la realización de esta investigación se contó con el apoyo de la Universidad Nacional de Colombia, entidad a la cual los autores expresan su agradecimiento.

REFERENCIAS

- 1 O. Ditlevsen y H. O. Madsen, “*Structural Reliability Methods*”, John Wiley and Sons, Chichester, (1996).
- 2 G.E.P. Box y N.R. Draper, “*Empirical Model Building and Response Surfaces*”, John Wiley and Sons, New York, (1987).
- 3 L. Faravelli, “Response-surface approach for reliability analysis”, *Journal of the Engineering Mechanics*, Vol. **115**, pp. 2763–2781, (1989).
- 4 S.H. Kim y S.W. Na, “Response surface method using vector projected sampling points”, *Structural Safety*, Vol. **19**, pp. 3–19, (1997).

- 5 M. Papadrakakis, V. Papadopoulos y N.D. Lagaros, "Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation", *Computer Methods in Applied Mechanics and Engineering*, Vol. **136**, pp. 145–163, (1996).
- 6 J.E. Hurtado y D.A. Alvarez, "Neural network-based reliability analysis: A comparative study", *Computer Methods in Applied Mechanics and Engineering*, Vol. **191**, pp. 113–132, (2001).
- 7 J.E. Hurtado, "Analysis of one-dimensional stochastic finite elements using neural networks", *Probabilistic Engineering Mechanics*, Vol. **17**, pp. 35–44, (2002).
- 8 R.T. Haftka, Z. Gurdal y M.P. Kamat, "Elements of structural optimization", Kluwer Academic Publishers, Dordrecht, 2ª edición, (1990).
- 9 Dan Frangopol, "Reliability-based structural design", en "*Probabilistic structural mechanics handbook*", C. (Raj) Sundararajan (ed.), Chapman & Hall, pp. 352–387, (1995).
- 10 Y.K. Lin y C.Q. Cai, "*Probabilistic structural dynamics. Advanced theory and applications*", McGraw-Hill, New York, (1995).
- 11 E. Simiu y R.H. Scanlan, "*Wind effects on structures*", John Wiley and Sons, New York, (1996).
- 12 P.E. Kloeden y E. Platen, "*Numerical solution of stochastic differential equations*", Springer-Verlag, Berlín, (1995).
- 13 M. Di Paola, "Stochastic differential calculus", "*Dynamic motion: chaotic and stochastic behaviour*", F. Casciati, (ed.) pringer-Verlag, Wien, pp. 29–92, (1993).
- 14 F. Szidarovszky y A.T. Bahill, "*Linear systems theory*", Boca Ratón, CRC Press, (1992).
- 15 A. Sternberg, "Stability investigation of long-span bridges using indicial functions with oscillatory terms", "*Stochastic Structural Dynamics 2 - New Practical Applications*", I. Elishakoff y Y.K. Lin (eds.), Springer-Verlag, Berlín, pp. 128–145, (1991).
- 16 C.G. Bucher y Y.K. Lin, "Stochastic stability of bridges considering coupled modes", *Journal of Engineering Mechanics*, Vol. **114**, pp. 2055–2071, (1988).
- 17 R.O. Duda, P.E. Hart y D.G. Stork, "*Pattern classification*", John Wiley and Sons, New York, (2001).
- 18 J.S.R. Jang, C.T. Sun y E. Mizutani, "*Neuro fuzzy and soft computing*", Prentice Hall, Upper Saddle River, EE.UU., (1997).
- 19 C.M. Bishop, "*Neural networks for pattern recognition*", Oxford University, Oxford, Press, (1995).
- 20 B.D. Ripley, "*Pattern recognition and neural networks*", Cambridge University Press, Cambridge, (1996).
- 21 V.N. Vapnik, "*Statistical learning theory*", John Wiley and Sons, New York, (1998).
- 22 V.N. Vapnik, "*The nature of statistical learning theory*", Springer-Verlag, New York, (2000).
- 23 T. Hastie, R. Tibshirani y J. Friedman, "*The Elements of Statistical Learning*", Springer-Verlag, New York, (2001).
- 24 P. Kall y S.W. Wallace, "*Stochastic programming*", John Wiley and Sons, Chichester, (1995).
- 25 T. Fine, "*Feedforward neural network methodology*", Springer-Verlag, New York, (1999).
- 26 F. Riesz y S. Nagy, "*Functional analysis*", Dover Publications, New York, (1990).
- 27 G. Wahba, "Support vector machines, reproducing Kernel Hilbert spaces and randomized GACV", "*Advances in Kernel Methods*", B. Schölkopf, C.J.C. Burges y A. Smola (eds.), The M.I.T. Press, Cambridge, pp. 69–88, (1999).
- 28 N. Cristianini y J. Shawe-Taylor, "*An Introduction to support vector machines and other Kernel-based Methods*", Cambridge University Press, Cambridge, (2000).
- 29 M. Anthony y P.L. Bartlett, "*Neural Network Learning: Theoretical Foundations*", Cambridge University Press, Cambridge, (1999).
- 30 R.E. Melchers, "*Structural Reliability: analysis and prediction*", John Wiley and Sons, Chichester, (1999).