Regresión con SVM para reducir la inconsistencia de la matriz AHP

Fabián E. Favret, Federico Matías Rodríguez, Marianela Daianna Labat

Departamento de Ingeniería y Ciencias de la Producción, Universidad Gastón Dachary
Posadas, Misiones, Argentina.

{fabianfavret, rfedericomatias, daiannalabat}@gmail.com

Resumen. Entre las herramientas de toma de decisiones, el proceso analítico jerárquico (AHP: Analytic Hierarchy Process) es una de las técnicas más usadas. Las propiedades complejas de su estructura permiten tener en cuenta las subjetividades en los juicios de los expertos pero también surge un considerable grado de inconsistencia cuando se computan las prioridades entre las alternativas de decisión. El presente trabajo de investigación analiza la herramienta de aprendizaje automático denominada máquina de vectores de soporte (SVM: Support Vector Machine), específicamente la versión de regresión denominada regresión de vectores de soporte épsilon (εSVR: Epsilon Support Vector Regression) para la reducir el grado de consistencia (CR: Consistency Ratio) en las matrices de comparativas de paridad AHP inconsistentes. Las pruebas obtenidas dieron como resultado que SVM tiene un porcentaje de precisión elevado en la predicción para reducir la inconsistencia de las matrices cuando se presentan entradas desconocidas para la red y una convergencia rápida con pocas iteraciones.

Palabras claves: Sistemas de toma de decisión multicriterio, AHP, SVM.

1 Introducción

Existe una variada gama de herramientas que ayudan al proceso decisorio y que permiten elegir una entre múltiples alternativas, utilizando diversos criterios para la comparación, como son los puntos de vista, las características, las propiedades, etc. Un método ampliamente usado y difundido se denomina AHP, una herramienta poderosa para la valuación de las alternativas disponibles y se diferencia de las demás técnicas en que permite incluir todos los factores relevantes para el proceso decisorio, sean éstos medibles, cuantificables o relacionados con la fuerza de las preferencias, sentimientos o subjetividades. Los resultados pueden ser clasificados y ordenados, y es posible medir el nivel de consistencia de los juicios emitidos [1], es decir, si existen errores debido a las subjetividades surgidas de las comparaciones entre pares.

Éste método presenta un inconveniente notable que surge luego de la evaluación de los criterios y es que al realizar las valoraciones de a pares suelen existir errores de transitividad, asignando erróneamente valores de importancia que están cargados con imprecisiones, incertidumbre o falta de completitud en los datos [2], provocando inconsistencias que son complejas de detectar y corregir a medida que el orden de las matrices se va incrementando [3]. Las matrices consistentes son esenciales ya que cuando se trabaja con intangibles, el juicio del ser humano es, por necesidad, inconsistente, y si con nueva información se puede mejorar la inconsistencia llevándola a la cercana consistencia, entonces esto mejoraría la validez de las prioridades en una decisión [1].

A manera de antecedentes al presente trabajo se observó que, en el 2006, Yi-Chung Hu y otros [4] habían propuesto un modelo de Red Neuronal utilizando perceptrón multicapa con redes propagación hacia atrás (BPN: Back-Propagation Network) para poder identificar la relación implícita entre las comparativas faltantes y las asignadas y así estimar un juicio de paridad faltante desde otras entradas asignadas minimizando el nivel de consistencia. Luego en 2010; Marcelo Karanik y otros [5] también presentan un modelo de perceptrón multicapa con BPN; para que dada una matriz de paridad incompleta, el modelo la complete con valores adecuados para las entradas faltantes y, al mismo tiempo, la matriz pueda ser mejorada hasta alcanzar un nivel satisfactorio de consistencia. En 2013, Marcelo Karanik y Leandro Wanderer [6] presentan la utilización de algoritmos genéticos aplicados a la recuperación de valores faltantes en matrices AHP y asistencia del usuario al momento de completar las matrices de comparación entre pares.

En la presente investigación se implementa SVM para corregir la inconsistencia de las matrices usadas en el proceso decisorio de AHP, evaluando el comportamiento de ésta herramienta en función de diversos criterios de comparación. Ésta herramienta de clasificación y regresión posee grandes ventajas como la ausencia de mínimos locales espurios en el proceso de optimización, el hecho de que se sintonizan pocos parámetros permitiendo desarrollos rápidos de las aplicaciones, la modularidad en el diseño y una excelente capacidad de generalización debido a la minimización del riesgo estructurado [7].

2 Construcción del Modelo AHP

En el presente trabajo se utilizan matrices AHP de orden 3, 4, 5, 6 y 7 para acotar el dominio y para una manipulación controlada. Se utilizan un total de 2000 matrices de paridad AHP (para cada orden) que son normalizadas, es decir, se utiliza la parte triangular superior de las mismas debido a que la diagonal principal es siempre 1 y la porción triangular inferior está compuesta por los recíprocos de la parte superior. Se dividen las 2000 matrices en dos porciones; 1000 de esas matrices son inconsistentes, es decir tienen un CR > 0,1 y se utilizarán como entrada a los métodos, mientras que las otras 1000 son consistentes y se utilizarán como ideales. Trabajos relacionados han demostrado que la utilización de ésta porción de la matriz ha sido suficiente para permitir la generalización en los modelos de aprendizaje automático [4] [5].

Para tratar el problema del sobreajuste que suele suceder en los métodos de aprendizaje se proporciona un conjunto de datos dividido en las siguientes partes: entrenamiento, validación y prueba. El tamaño del conjunto de entrenamiento y validación es de 700 matrices y el conjunto de prueba será de 300 matrices.

Cada matriz AHP es ingresada como entrada al modelo a manera de arreglo unidimensional (ver Figura 1), es decir cada entrada es un vector que se normaliza antes de ser introducido.

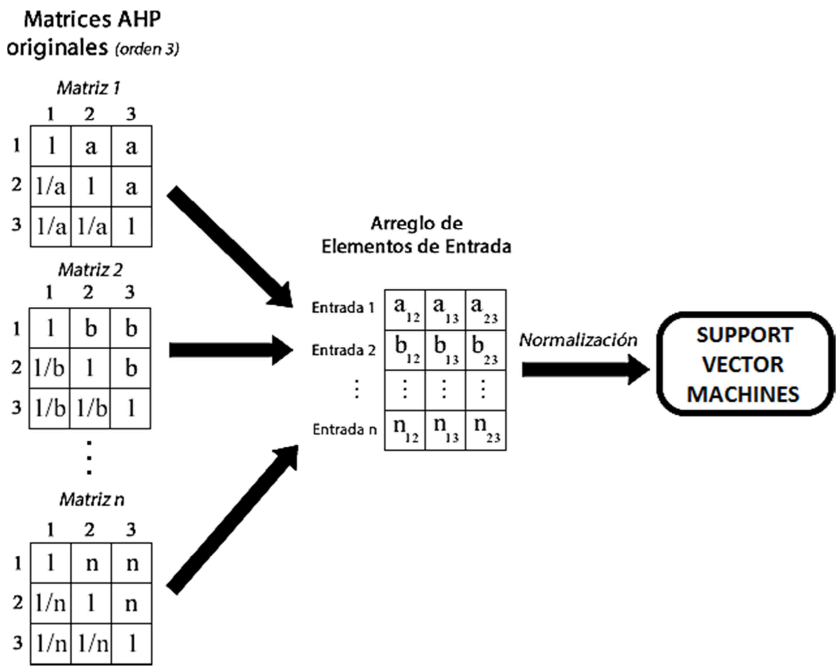

Fig. 1. Esquema de la estructura del arreglo de entrada de matrices AHP

Es necesario un pre procesamiento de los datos de entrada antes de trabajar con ellos, en SVM se utiliza un kernel Gausiano denominado de “base radial” (RBF: Radial Basis Function) y que abarca valores en el intervalo [0-1].

Antes de proceder a la normalización directa de éstos valores es necesario una distribución de cada ponderación de la escala de Saaty en una nueva escala entre el 1 y el 17, debido a que en el método propuesto la salida es un valor continuo y por ello se necesita que haya una diferencia considerable entre cada elemento de la escala para una correcta separación de clases. Luego se distribuyen éstos valores enteros uniformemente en valores reales entre 0 y 1.

3 Método Propuesto de Máquina de Vectores de Soporte

En ésta arquitectura del modelo propuesto se utilizan múltiples SVM donde para cada una los parámetros utilizados son los siguientes: el tipo de SVM a utilizar y el tipo de kernel. El tipo de SVM se denomina εSVR (ilustrado en la Figura 2 como Epsilon SVR), el kernel es un kernel RBF el cual fue elegido primeramente por un componente teórico [8] y en segundo lugar evaluando su comportamiento durante las pruebas. Se entrena un SVM por cada elemento de la matriz de la parte triangular superior, es decir se tendrán $\frac{n(n-1)}{2}$ salidas, siendo n el orden de la matriz (el esquema puede observarse en la Figura 2).

Los parámetros de la SVM son gamma (γ), la constante (C) y el elemento ε. Estos parámetros son seleccionados de manera particular acorde al comportamiento observado en el entrenamiento utilizado en éste modelo, es por ello que son valores aplicables exclusivamente a éste dominio según las observaciones realizadas. Al entrenar, el método evalúa en cada iteración una combinación de γ y C, si el error disminuye entonces prosigue a la siguiente iteración, si no disminuye vuelve a probar automáticamente (sin interacción del usuario) con otra combinación hasta agotar todos los valores posibles del rango de valores que puede utilizar; se definió un rango para $\gamma = [1-100]$ y para $C = [1-100]$ (límites superiores a 100 de éstos dos parámetros no son necesarios ya que el método rápidamente encuentra una combinación antes de arribar al último valor de algunas de las variables). Las iteraciones se detienen cuando el error es menor a $0{,}01$.

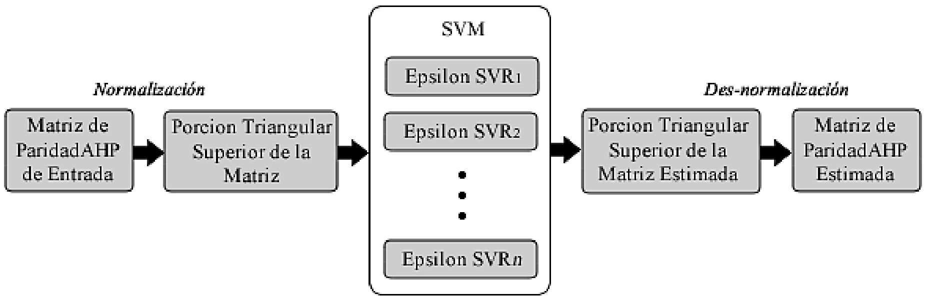

Fig. 2. Arquitectura del modelo propuesto de múltiples SVM.

Para poder implementar SVM se utiliza una librería Java de aprendizaje automático para investigación científica denominada Encog[1]. Se utiliza la clase `SVM` para crear una Máquina de Vectores de Soporte por cada elemento a estimar y la clase `SVMSearchTrain` que realiza un entrenamiento de SVM mucho más profundo y preciso.

4 Análisis e interpretación de resultados

En base a las pruebas que se han realizado podemos observar una síntesis de resultados del método SVM (ver Tabla 1) del comportamiento en cuanto a la velocidad y cantidad de iteraciones en el entrenamiento del modelo; durante el proceso de prueba

1 http://www.heatonresearch.com/encog

se ha observado se mantiene un porcentaje elevado de aciertos sin importar el tamaño del orden de la matriz; gráficamente se puede observar el comportamiento en cuestión de precisión obtenida en las pruebas, basado en los órdenes de las matrices (ver Figura 3). También puede verse que la cantidad de iteraciones es de poca cantidad y por ende el tiempo que le lleva llegar a la convergencia es de tan solo unos pocos segundos (15,31 segundos en el peor de los casos).

Tabla 1. Síntesis de resultados del método SVM.

Orden de la Matriz	Elementos Entrada	Número de Iteraciones	Tiempo de Entrenamiento (segundos)	Número de Aciertos Prueba	Aciertos Prueba (%)
3	3	126	15,31	276	92,00
4	6	6	2,45	294	98,00
5	10	10	5,26	283	94,33
6	15	15	8,81	266	88,67
7	21	21	11,80	285	95,00

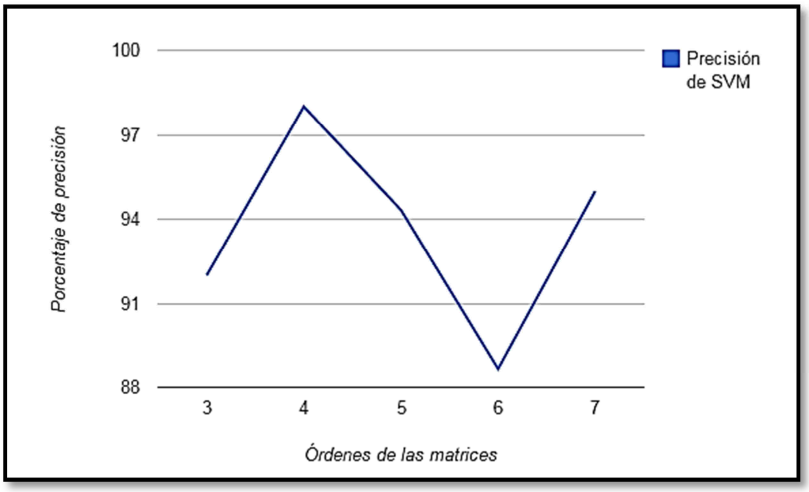

Fig. 3. Porcentaje de precisión de SVM obtenido en las pruebas.

Al observar los valores promedio de CR consistentes sobre los conjuntos de prueba (ver Tabla 2 y Figura 4) se pone de manifiesto la consistencia promedio obtenida en base a las entradas de prueba, comparando con la inconsistencia de las mismas, previa al procesamiento con los modelos propuestos, en ella se puede visualizar que SVM ha logrado desempeñar un rol satisfactorio en la corrección de la inconsistencia, es decir, teniendo valores de porcentaje de reducción de CR similares para cada orden de matrices.

Tabla 2. Valores promedio de CR obtenidos en aciertos de conjuntos de prueba.

Orden de la Matriz	CR Promedio Inconsistente	CR Promedio Consistente	Reducción obtenida (%)
3	0,149150253	0,034303319	77,00
4	0,152962688	0,043543081	71,53
5	0,159132193	0,057162816	64,08
6	0,166571810	0,066960748	59,80
7	0,174561902	0,063129357	63,84

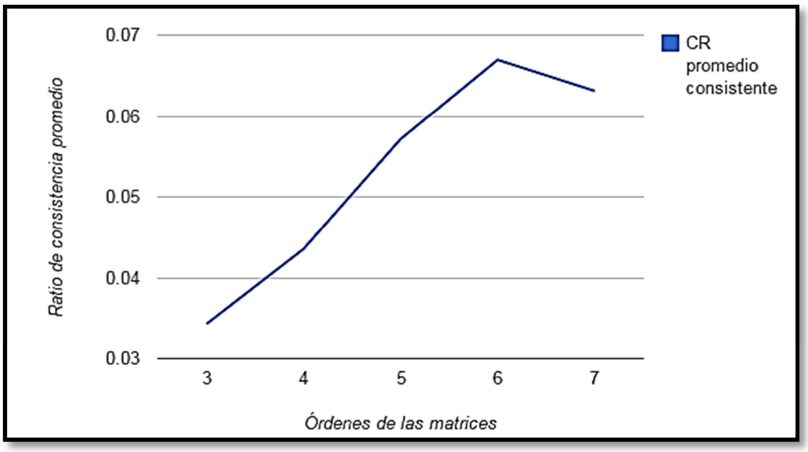

Fig. 4. Valores promedio de CR consistentes sobre los conjuntos de prueba.

5 Conclusión

Al utilizar SVM para realizar las correcciones de la inconsistencia de las matrices del AHP, se encontró que el método tiene un porcentaje de precisión elevado en la predicción cuando se presentan entradas desconocidas para la red y aporta la ventaja de una convergencia notablemente rápida, al tener poca cantidad de iteraciones.

Podemos concluir que SVM es una herramienta que puede ser utilizada para desarrollos ágiles y adaptables a nuevas propuestas de órdenes de matrices, ya que ha mostrado un comportamiento invariable y homogéneo sin importar la modificación que se haga a los tamaños de las matrices.

Si bien el modelo propuesto SVM, realiza una estimación de las matrices logrando reducir la inconsistencia, cabe notar que los juicios de comparativas de paridad sugeridos por éstas técnicas pueden no estar en consonancia con lo que desee el experto que está realizando las ponderaciones, es por ello que como trabajo a futuro se propone realizar una evaluación estadística de las sugerencias propuestas por el método y la aceptación del cambio por parte del experto. Si el experto desea o no realizar un cambio, y si ésta sugerencia está direccionada hacia el valor de ponderación que desea el mismo, o no, de ésta manera se podría evaluar la precisión de la herramienta en el campo de aplicación.

Bibliografía

[1] R. W. Saaty, "The Analytic Hierarchy Process - What it is and how it is used," in *Mathematical Modelling*, vol. IX, Londres, 1987, pp. 161-176.

[2] S. O. Mujgan and T. L. Saaty, "The unknown in decision making: What to do about it," in *European Journal of Operational Research*, vol. I, 2006, p. 349–359.

[3] G. Coyle, "The Analytic Hierarchy Process (AHP)," in *Practical Strategy: Structured tools and*, 2004, pp. 1-11.

[4] H. Yi-Chung and T. Jung-Fa, "Backpropagation multi-layer perceptron for incomplete pairwise comparison matrices in analytic hierarchy process," in *Applied Mathematics and Computation*, vol. 180, 2006, pp. 53-62.

[5] J. A. Gomez-Ruiz, M. Karanik, and J. I. Peláez, "Estimation of missing judgments in AHP pairwise matrices using a neural network-based model," in *Applied Mathematics and Computation*, vol. 216, 2010, p. 2959–2975.

[6] M. Karanik and L. Wanderer, "Aplicación de algoritmos genéticos a la corrección de matrices inconsistentes en el proceso analítico jerárquico," *XV Workshop de Investigaciones en Ciencias de la Computación*, pp. 818-822, 2015.

[7] B. Schölkopf and A. J. Smola, "Support Vector Machines," in *The Handbook of Brain Theory and Neural Networks*. Cambridge: MIT Press, 2003, pp. 1119-1125.

[8] C. Burgues, "A Tutorial on Support Vector Machines for Pattern Recognition," in , Boston, 1998.

[9] T. L. Saaty, "How to make a decision: The Analytic Hierarchy Process," in *European Journal of Operational Research*, vol. 48, Holanda del Norte, Paises Bajos, 1990, pp. 9-26.

[10] T. L. Saaty, "Decision-making with the AHP: Why is the principal eigenvector necessary," in *European Journal of Operational Research, Issue 1*, vol. 145, 2003, p. 85–91.

[11] T. L. Saaty, *The Analytic Hierarchy Process*. New York, USA: McGraw-Hill, 1980.

[12] R. Larson and D. C. Falvo, "Numerical Methods," in *Elementary Linear Algebra*. Boston, USA: Houghton Mifflin Harcourt Publishing Company, 2009, ch. 7, pp. 421-434.

[13] V. Vapnik and C. Cortes, "Support-Vector Networks," in *Machine Learning*, vol. 20, Nueva Jersey, 1995, pp. 273-297.