SELECCIÓN DE HIPERPARÁMETROS EN MÁQUINAS DE SOPORTE VECTORIAL UTILIZANDO ADAPTACIÓN DE MATRIZ DE COVARIANZA

RESUMEN
Se presenta un método de selección automática de hiperparámetros en máquinas de soporte vectorial (SVM) utilizando algoritmos evolutivos y cotas efectivas del error de validación. La estrategia evolutiva analizada es la *Adaptación de Matriz de Covarianza*, la cual reduce el tiempo de convergencia, al necesitar un menor número de evaluaciones de la función objetivo. Se emplean dos cotas del error de validación: la *validación cruzada,* como forma generalizada del esquema LOO, y el *span* como medida efectiva en el sentido de no requerir múltiples evaluaciones de la SVM, que siendo continua requiere una carga computacional considerablemente pequeña. Los resultados numéricos obtenidos con las bases de datos de la colección UCI y StatLog, para el caso de análisis de Conjunto Multi Clase y Kernel Polinomial muestran un desempeño competitivo con otras técnicas de uso común.

PALABRAS CLAVES: Máquinas de Soporte Vectorial, Selección de Hiperparámetros, Estrategias Evolutivas, Adaptación de Matriz de Covarianza.

RICARDO HENAO
Ingeniero Electrónico, Ms.C.
Profesor Auxiliar
Universidad Tecnológica de Pereira
rhenao@ohm.utp.edu.co

JORGE EDUARDO HURTADO
Ingeniero Civil, Ph.D.
Profesor Universidad Nacional de Colombia sede Manizales
Jhurtado14@epm.net.co

GERMAN CASTELLANOS D
Ingeniero Electrónico, Ph.D.
Profesor Universidad Nacional de Colombia sede Manizales
gcastell@ieee.org

ABSTRACT
This paper describes a method for automated Hyperparameter Selection of SVM, using a combination of a concrete Evolving Strategy and error validation restrictions (empirical risk). The proposed method includes a evolving strategy known as Covariance Matrix Adaptation procedure because it's small time of convergence. Two different measures of empirical risk were analysed: a cross validation and span schemes. The last one requires small amount of computational resource. Obtained results for UCI and StatLog Databases, considering a polynomial kernel, show a better performance in comparison to other common methods of Hyperparameter Selection for SVM.

KEYWORDS: *Support Vector Machines, Hyperparameter Selection, Evolving Strategies, Covariance Matrix Adaptation.*

1. INTRODUCCIÓN

Las máquinas de soporte vectorial (SVM) se emplean ampliamente como herramienta de clasificación y corresponden a la familia de funciones de la forma:

$$f(\mathbf{x},\alpha) = \mathbf{w}\cdot\mathbf{\Phi}(\mathbf{x}), \qquad (1)$$

donde $\mathbf{x}\in\Re^d \equiv \ell$ es un vector compuesto de d características que lo describen, $\mathbf{w}$ es el vector de parámetros ajustables y calculados a partir de un conjunto de entrenamiento dado y $\mathbf{\Phi}$ representa la función de mapeo sobre el espacio de Hilbert $\boldsymbol{H}$, tal que $\mathbf{\Phi}:\ell\rightarrow H$, el cual es generalmente de una dimensión mucho más alta que la dimensión del espacio original ℓ, en el que se espera que los datos de entrenamiento sean linealmente separables. Cuando se presenta un vector $\mathbf{x}$ a la SVM entrenada, el signo de su evaluación en (1) debe proporcionar la correcta clase de cada dato, asumiendo el caso de clasificación binaria.

En todo caso, las SVM requieren del ajuste del vector de parámetros $\mathbf{w}$, lo cual se realiza entrenando el clasificador. Así, dado un conjunto de entrenamiento $\{\mathbf{x}_i\}$, sus clases asociadas están representadas, respectivamente, por $\{y_i\}\in\{-1,1\}$ para $i = 1,\ldots, l$; el vector de pesos $\mathbf{w}$ se expresa como la combinación lineal:

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{\Phi}(\mathbf{x}_i),$$

donde α se encuentra al minimizar la función cuadrática:

$$L(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{\Phi}(\mathbf{x}_i)\mathbf{\Phi}(\mathbf{x}_j), \qquad (2)$$

La expresión (2) está sujeta a la restricción $\boldsymbol{C}\geq\alpha_i\geq\boldsymbol{C}$, para $i = 1,\ldots, l$, donde $\boldsymbol{C}$ es el valor a seleccionar por el usuario, el cual se utiliza para hacer que el clasificador sea tolerante a la falta de separabilidad de los datos de entrenamiento en el espacio $\boldsymbol{H}$, donde un valor grande de $\boldsymbol{C}$ corresponde al caso en que los datos de entrenamiento son perfectamente separables. Aquellos puntos para los

Fecha de recepción: 31 Enero de 2005
Fecha de aceptación: 20 Marzo de 2005

cuales $\alpha_i > 0$, se conocen como los *vectores de soporte* y representan los puntos más cercanos al hiperplano frontera de decisión en el espacio $\boldsymbol{H}$.

El producto interno $\boldsymbol{\Phi}(\mathbf{x}_i)\cdot\boldsymbol{\Phi}(\mathbf{x}_j)$ en $\boldsymbol{H}$ se realiza mediante el uso de la función $K(\mathbf{x}_i,\mathbf{x}_j)$, conocida como el *núcleo* o *kernel* de la SVM, el cual debe cumplir con las condiciones de Mercer [1]. Dos kernel de amplia aceptación son el Gaussiano RBF (*Radial Basis Function*) y el polinomial no homogéneo, descritos respectivamente como:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2), \qquad (3)$$

$$K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i'\mathbf{x}_j + c)^d \text{, para } c > 0, \qquad (4)$$

Para obtener un buen desempeño de la SVM, algunos de sus parámetros deben ser escogidos cuidadosamente. Estos parámetros incluyen:

- El valor de regularización $\boldsymbol{C}$.
- Los valores concernientes al kernel (γ, $\boldsymbol{C}$ y d) que implícitamente definen la aplicación no lineal del espacio de entrada a alguno de representación de mayor dimensión.

Estos parámetros de *alto nivel* son usualmente llamados *hiperparámetros*, cuya selección es generalmente realizada minimizando la estimación del error de generalización. Dos cotas del error de validación se analizan: la validación cruzada y la cota *dejar uno afuera* (LOO - *Leave One Out*) con la restricción de requerir la solución de múltiples SVM bajo un esquema de búsqueda en malla, de manera que por eficiencia es bastante útil contar con estimaciones más simples que sean mucho más económicas computacionalmente. Durante los últimos años, algunas formas de estimación han sido propuestas, entre ellas: $\xi\alpha$ [2], *Radio/Margen* [3] y la forma *span* de los vectores de soporte [4].

2. ADAPTACIÓN DE MATRIZ DE COVARIANZA

La estrategia evolutiva (ES) analizada en este trabajo corresponde a la llamada $(\mu/\mu,\lambda)$-(CMA-ES) (*Covariance Matrix Adaptation Evolution Strategy*) [5], por considerarse altamente eficiente. En este caso, cada individuo representa un vector objeto n-dimensional con valores reales alterados o actualizados mediante mutaciones y recombinaciones de ellos, en particular por recombinación intermedia global, la cual implica el cálculo del centro de masa de los μ individuos en la población inmediatamente anterior, así como de la mutación añadiendo vectores aleatorios normal distribuidos con media cero, de modo que toda la matriz de covarianza sea adaptada durante la evolución para mejorar la estrategia de búsqueda. Formalmente, los parámetros objeto $x_k^{(g+1)}$ con progenie $k=1,\ldots,\lambda$ creada en la generación g están dados por:

$$x_k^{(g+1)} = \langle x \rangle^{(g)} + N_k^{(g)}(0, \sigma^{(g)^2} C^{(g)})$$

donde $\langle x \rangle^{(g)}$ denota el centro de masa de la población en la generación g y el segundo término de (5), son realizaciones independientes de un vector aleatorio n-dimensional normal distribuido con media cero y matriz de covarianza $\sigma^{(g)^2}\boldsymbol{C}^{(g)}$. Los parámetros estratégicos, es decir, la matriz $\boldsymbol{C}^{(g)}$ y el paso global $\sigma^{(g)^2}$ son actualizados en línea, que se efectúan utilizando el algoritmo de *adaptación de matriz de covarianza* (CMA), el cual desarrolla conceptos importantes de la adaptación de parámetros estratégicos, en particular la desaleatorización en el sentido que la distribución de mutación es alterada de manera determinística, tal que la probabilidad de reproducir pasos en el espacio de búsqueda tienda a un incremento en el tamaño de la población, de ese modo, el algoritmo detecta correlaciones entre las variables objeto y se vuelve ortogonal ante transformaciones en el espacio de búsqueda (independiente de la inicialización). Otro concepto es la cumulación, que con miras a utilizar la información obtenida en generaciones previas eficientemente, tiene en cuenta la trayectoria de búsqueda de la población en un número dado de generaciones anteriores.

3. MÉTODO PROPUESTO

El algoritmo propuesto consta básicamente de tres partes: la máquina de soporte vectorial (SVM), el algoritmo evolutivo de adaptación de matriz de covarianza (CMA-ES) como esquema de adaptación del tamaño del paso en la estrategia evolutiva que sirve como forma de optimización de los hiperparámetros. Finalmente, el algoritmo consta de la función que se utiliza como cota del error de generalización y objetivo de CMA-ES. Con relación a la misma función objetivo, se analizan dos tipos: la validación cruzada (CMA-ES-SVM-CV), que en este caso es de 4 particiones con la idea de mantener la simplicidad del algoritmo de búsqueda en malla y el *span* de los vectores de soporte (CMA-ES-SVM-SB) [4]; básicamente para evitar múltiples evaluaciones de la función de SVM en cada iteración del algoritmo de optimización. En el caso de la función *span* de los vectores de soporte, se escogió debido a que supone resultados aceptables [4] y es menos exigente en términos computacionales con relación a la que es tal vez la más conocida - *Radio/Margen*.

La función *span* se calcula como:

$$T = \frac{1}{N}\sum_{p=1}^{N}\Psi(\alpha_p^0 S_p^2 - 1)$$

donde $S_p^2 = 1/(\widetilde{K}_{sv}^{-1} + D)_{pp} + D_{pp}$, siendo D una matriz diagonal con elementos $D_{ii} = 1/\alpha_i^0$ y $D_{n+1,n+1} = 0$. En [4] se sugiere hacer $\Psi(x) = (1+\exp(-5x))^{-1}$ para suavizar la aproximación del error de validación. Las constantes en $\Psi(x)$ se obtienen de calcular la estimación de densidad *a posteriori* para una SVM [6]. En el caso multi-clase, el *span* se puede replantear de la siguiente manera:

$$LOO \leq T_i \leq T_{span\_mc}$$

donde T_{span_mc} y T_i son el *span* del problema multi-clase y de cada subproblema binario bajo el esquema *uno contra uno* que requiere $k(k\text{-}1)/2$ clasificadores para k clases [7]. Con la SVM y la función objetivo cubiertas, el algoritmo CMA-ES utiliza un tamaño de paso inicial con valor 0.3 [8], mientras los hiperparámetros que van a ser buscados (por ejemplo $\boldsymbol{C}, \sigma^2$ para el caso del kernel RBF) se considera la transformación de las variables: $u_1=\ln(\boldsymbol{C})$ y $u_2=\ln(\sigma^2)$ sugerida por [4]. Los valores iniciales para los hiperparámetros son $\ln \boldsymbol{C}=0$ y $\ln\sigma^2=0$ para el kernel RBF considerado, con tolerancia de parada 10^{-3}, es decir, $|f(u+1)\text{-}f(u)| \leq 10^{-3} f(u)$ [8], y tolerancia de los parámetros de entrada 10^{-5} de la misma forma.

Considerando que el algoritmo CMA-ES-SVM no utiliza la diferenciabilidad de las funciones CV y SB, tampoco lo requiere en la función kernel, de manera que en lo que se refiere a la función kernel, la única limitación es que la matriz **K** tenga inversa, que no supone un inconveniente si el kernel cumple las condiciones de Mercer, y en cuanto a esquemas multi-clase, ninguna de las dos funciones utilizadas presenta inconvenientes.

4. RESULTADOS

Los resultados numéricos fueron obtenidos realizando tareas de clasificación sobre tipos diferentes de datos estándares tomados de las bases de datos **UCI** (*Repository Of Machine Learning Databases and Domain Theories*) y **StatLog** (*Department of Statistics and Modelling Science*) [10] para problemas binarios, conjuntos multi-clase y resultados para el kernel polinomial. Los experimentos fueron llevados a cabo utilizando búsqueda en malla (para $\boldsymbol{C}$ = [-5,5], γ = [-5,5] y Δ = 0.5, Radio/Margen para L1 y L2 [4] y el algoritmo propuesto CMA-ES-SVM para una SVM entrenada con el algoritmo de descomposición SMO [6] considerando las modificaciones propuestas por [2,9] para mejorar el desempeño del clasificador. Los kernel empleados fueron el RBF de la ecuación (3) y el polinomial no homogéneo de la ecuación (4). Las medidas de desempeño consideradas en este trabajo son: la precisión del clasificador (%), el número de vectores de soporte del modelo construido (NSV) y el número de iteraciones requerido para encontrar dicho modelo (#).

4.1 Conjuntos Estándares

Los conjuntos de datos empleados, correspondientes a la colección **UCI** y **StatLog**, comprenden las siguientes casos: *Breast*, *Diabetes*, *German*, *Heart* e *Ionosphere*; todos ellos de diferentes naturalezas y tamaños respecto a observaciones y características. La estructura de los conjuntos mencionados se muestra en la tabla (1). En la mayoría de los casos la cantidad de datos que deben ser utilizados para entrenamiento es sugerida por el autor, por esta razón es tenida en cuenta donde es aplicable.

Conjunto	Caracte-rísticas	Entrena-miento	Valida-ción	Total
Breast	9	383	300	683
Diabetes	8	468	300	768
German	24	700	300	1000
Heart	13	170	100	270
Ionosphere	34	200	151	351

Tabla 1. Estructura de los conjuntos estándares

Conjunto	*Métodos*	*%*	*NSV*	*#*
Breast	Malla	98.0	144	484
	R/M L2	98.7	211	10
	R/M L1	97.7	194	22
	CMA-CV	98.7	53	68
	CMA-SB	82.0	280	8
Diabetes	Malla	73.0	441	484
	R/M L2	76.7	468	14
	R/M L1	99.7	38	26
	CMA-CV	99.0	68	80
	CMA-SB	90.3	68	20
German	Malla	68.3	699	484
	R/M L2	69.3	700	13
	R/M L1	59.3	700	31
	CMA-CV	69.0	700	24
	CMA-SB	69.7	700	24
Herat	Malla	56.0	170	484
	R/M L2	70.0	170	17
	R/M L1	50.0	100	16
	CMA-CV	56.0	170	24
	CMA-SB	69.7	700	24
Ionosphere	Malla	98.0	123	484
	R/M L2	97.4	176	12
	R/M L1	87.8	156	26
	CMA-CV	97.4	118	33
	CMA-SB	98.8	165	32

Tabla 2. Rendimiento de estrategias de selección de hiperparámetros de las SVM

Los resultados para los conjuntos estándar mostrados en la tabla (2) establecen que para *Breast*, R/M L2 y CMA-ES-SVM-CV obtienen la misma precisión pero el segundo requiere casi cuatro veces menos vectores de soporte. Con el conjunto *Diabetes*, R/M L1 obtiene el mejor resultado con un requerimiento de iteraciones superior al de R/M L2. El conjunto *German* presenta soluciones claramente sobre-entrenadas con número de iteraciones simi-

lar excluyendo la búsqueda en malla y un ligero mejor pero no substancial desempeño en términos de precisión por parte de CMA-ES-SVM-SB, lo cual puede ser debido a que la cantidad de información disponible no es suficiente para describir el problema o que es necesario hacer un escalamiento del espacio de entrada [10]. El conjunto *Heart* presenta dos soluciones no sobre-entrenadas, sin embargo, en comparación a la mejor considerando la precisión, CMA-ES-SVM-SB presenta la diferencia menos crítica igual a 5%. Los resultados para *Ionosphere* exhiben un mejor valor de precisión para CMA-ES-SVM-SB pero un menor número de vectores de soporte para CMA-ES-SVM-CV con un decremento pequeño en la precisión (~1.5%) y número de iteraciones para ambos en un rango razonable en comparación al requerido por R/M L2.

4.2 Conjunto Multi Clase

Con la finalidad de probar el funcionamiento del algoritmo propuesto bajo un esquema de entrada multi-clase, se tomaron tres conjuntos de datos estándar denominados: *Iris*, *Satimage* y *Segment*. El primero se escogió debido a que es, tal vez el más común en reconocimiento de patrones, mientras los otros dos brindan la posibilidad de probar el desempeño del algoritmo propuesto ante espacios de entrada grandes.

Conjunto	*Características*	*Clases*	*Entrenamiento*	*validación*
Iris	3	4	120	30
Satimage	36	7	4435	2000
Segment	20	7	1848	462

Tabla 3. Estructura de los conjuntos multiclase

Conjunto	*Método*	*%*	*NSV*	*#*
Iris	CMA-CV	96.7	43	24
	CMA-SB	96.7	45	20
Satimage	CMA-CV	90.5	1405	60
	CMA-SB	89.7	1494	20
Segment	CMA-CV	96.2	401	122
	CMA-SB	94.8	478	8

Tabla 4. Resultados para conjuntos de prueba multi-clase

La estructura de los conjuntos se muestra en la tabla (3), la cual muestra un mejor resultado en términos de precisión y número de vectores de soporte para CMA-ES-SVM-CV. En el caso CMA-ES-SVM-SB revela un número menor de iteraciones en los tres conjuntos con un sacrificio en la precisión relativamente pequeño (ver tabla 4). Considerando que se trata un esquema multi-clase y dos de las bases de datos son grandes se puede decir que las dos formas del algoritmo propuesto se comportan bastante bien.

4.3 Resultados con Kernel Polinomial

El objetivo de esta prueba es observar el funcionamiento del algoritmo propuesto a un kernel diferente a RBF, para esto se utiliza el kernel polinomial no homogéneo de la ecuación (4) para dos conjuntos de datos anteriormente utilizados: uno binario, *Breast* y otro multi-clase, *Iris*. El grado del polinomio se escogió en el rango $d=(1,6)\in \mathbf{N}$, $c=[0,1]\in \Re$ y la transformación para $u=\ln \boldsymbol{C}$ se mantiene. Como valores iniciales se seleccionaron $d=3$, $c=0$ y $u=\ln \boldsymbol{C}=0$.

Conjunto	*Método*	*%*	*NSV*	*#*
Iris	CMA-CV	93.3	13	28
	CMA-SB	93.3	17	23
Breast	CMA-CV	92.3	31	28
	CMA-SB	94.7	26	9

Tabla 5. Resultados para el *kernel* polinomial

Los resultados obtenidos en la tabla (5) muestran que efectivamente al algoritmo propuesto funciona para este kernel en específico.

De la formulación del kernel polinomial se puede anotar que en este caso se deben ajustar 3 parámetros en vez de dos y que considerando esto la cantidad de iteraciones sigue siendo considerablemente pequeña. Aunque las pruebas presentadas en esta sección son para un solo kernel diferente, son concluyentes pues el número de parámetros se incrementó, el rango de d y c debe ser limitado para que el kernel cumpla con las condiciones de Mercer y en general el kernel polinomial es problemático.

5. CONCLUSIONES Y RECOMENDACIONES

En términos generales, los resultados obtenidos en este trabajo muestran que el algoritmo propuesto en sus dos variantes (CV y SB) es bastante competitivo en comparación a las técnicas basadas en la diferenciabilidad de las cotas del error (L1 y L2), si se observa que en varias ocasiones, las desarrolladas en este trabajo obtuvieron desempeños similares y otras tantas veces superiores en los experimentos, sin tener en cuenta el beneficio adicional proporcionado por el soporte multi-clase, la posibilidad de utilizar kernels no diferenciables y la selección de múltiples parámetros.

Si se observa cuidadosamente, el rango de precisión para la diferentes pruebas efectuadas no fue muy grande, sin embargo, muchas veces superior para el caso de CMA-ES-SVM-SB, además, en los casos en los que las técnicas utilizadas para realizar la comparación obtuvieron mejores valores de precisión no lo eran tanto respecto del algoritmo desarrollado en este trabajo. Por otro lado, el número de vectores soporte obtenido en los diferentes experimentos es satisfactorio y considerablemente mejor

para CMA-ES-SVM-CV, si se parte de el hecho que NSV también es una medida de generalización para SVM y sin mencionar que entre más pequeña sea esta cantidad menor es la carga computacional requerida tanto para el entrenamiento como para la validación.

Los resultados obtenidos en este trabajo hacen que el algoritmo propuesto sea ideal para aplicaciones específicas donde se requiera un clasificador con arquitectura robusta sobre todo en términos de automatización, debido a que en muchas de ellas se necesita una máquina que se actualice o se construya independientemente sin la necesidad de reformular completamente la estructura del problema.

Si se trata de evaluar independientemente las dos variantes del algoritmo presentadas en este trabajo se debe decir que: CMA-ES-SVM-SB es la elección apropiada teniendo en cuenta la precisión y el número de iteraciones y CMA-ES-SVM-CV si en el problema que se está tratando se considera más importante un número de vectores de soporte menor ya sea por el costo computacional o por que se tiene un número limitado de observaciones. En contraste, si de antemano se tiene conocimiento que el problema a resolver es muy complejo tal vez CMA-ES-SVM-SB ofrezca más ventajas dados los resultados obtenidos y la cantidad de iteraciones que naturalmente tenderían a incrementar.

6. BIBLIOGRAFÍA

[1] BURGES, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition, Knowledge Discovery and Data Mining, v. 2, n. 2, pp. 121-167, 1998

[2] JOACHIMS T., Estimating the Generalization Performance of a SVM Efficiently, "Proceedings of ICML-00, 17th International Conference on Machine Learning, Pat Langley, pp. 431-438, 2000

[3] Vapnik. V.N., The Nature of Statistical Learning Theory. Wiley - Interscience.1998

[4] Chapelle O., Vapnik V. N., Choosing Multiple Parameters for Support Vector Machines, Machine Learning, v. 46, n. 1-3, pp. 131-159, 2002

[5] HANSEN, N. and OSTERMEIER, A., Completely Derandomized Self-Adaptation in Evolution Strategies, Evolutionary Computation, v. 9, n. 2, The MIT Press, pp 159-195, 2001

[6] PLATT, J.C., Probabilities in Support Vector Machines. Advances in Large Margin Classifiers, MIT Press, 2000

[7] KNERR, S., L. PERSONNAZ, and G. DREYFUS, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in Neurocomputing: Algorithms, Architectures and Applications, Springer, 1990

[8] HENAO, R., Selección de Hiperparámetros en Máquinas de Soporte Vectorial, Tesis de Maestría, Universidad Nacional de Colombia, Manizales, 2004.

[9] BLAKE, C.L. and MERZ, C.J. UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences, 1998

[10] HSU, C.W., CHANG C.C. LIN , C.J. , A Practical Guide to Support Vector Classification, Department of Computer Science and Information Engineering, National Taiwan University, 2003