

Optimización de hiperparámetros de SVM utilizando PSO

Algoritmos Evolutivos I - Carrera de Especialización en Inteligencia Artificial (CEIA) - UBA

Alumna: Ing. Natalia Espector - nataliaespector@gmail.com

1. Introducción

En este trabajo se aborda la optimización de hiperparámetros de un modelo de Support Vector Machine (SVM) para clasificación, empleando el algoritmo Particle Swarm Optimization (PSO). El objetivo principal es encontrar los valores óptimos de C y γ que maximicen la accuracy del modelo. Los resultados obtenidos mediante PSO se compararán con los provistos por RandomizedSearchCV, utilizado como baseline.

Además, se analizará el impacto del factor de inercia w en el desempeño del algoritmo, evaluando su efecto tanto en la velocidad de convergencia como en la calidad de las soluciones alcanzadas. Para ello se estudiará el compromiso entre exploración y explotación que este parámetro introduce en la dinámica de búsqueda.

Los experimentos se realizaron utilizando el dataset **Breast Cancer Wisconsin**, disponible en *scikit-learn*, el cual contiene características morfológicas de tumores de mama clasificadas en dos categorías: benignos y malignos.

Se desarrolló una implementación propia del algoritmo PSO, en lugar de emplear la librería *pyswarm*, con el fin de disponer de mayor flexibilidad para el análisis y, en particular, para obtener y estudiar las curvas de convergencia.

El código completo del proyecto se encuentra disponible en el siguiente repositorio: <https://github.com/nataliaespector/CEIA-TP-AlgoritmosEvolutivosI.git>

2. Materiales y métodos

2.1 Configuración general

Para el entrenamiento y la evaluación del modelo se utilizó el dataset *Breast Cancer Wisconsin*, junto con validación cruzada estratificada (*Stratified K-Fold*), a fin de preservar la proporción entre clases en cada partición. Se fijó una semilla aleatoria para asegurar reproducibilidad en los experimentos.

Los hiperparámetros a optimizar fueron C y γ , correspondientes al clasificador *Support Vector Machine (SVM)* con kernel RBF. Dado que estos parámetros impactan exponencialmente en el modelo y sus valores relevantes suelen encontrarse dispersos en varios órdenes de magnitud, se decidió realizar la búsqueda en escala logarítmica, considerando los siguientes rangos:

- C entre $[10^{-2}, 10^3]$
- γ entre $[10^{-4}, 10^1]$

2.2 Función objetivo

Se definió una función objetivo que recibe los parámetros $\log(C)$ y $\log(\gamma)$, los datos (X, y) y el objeto de validación cruzada. Para cada par de hiperparámetros se crea un *pipeline*, compuesto por:

- Estandarización de características (Standard Scaler)
- SVC (Support Vector Machine para clasificación) con los parámetros C y γ correspondientes

Luego se calcula la *accuracy* utilizando validación cruzada, y finalmente se devuelve el valor promedio de esa *accuracy*. Como PSO es un algoritmo de minimización, se devuelve el valor negativo de la *accuracy*.

2.3 Función PSO

Se implementó una versión propia del algoritmo Particle Swarm Optimization basada en el material visto en clase, incorporando modificaciones para adaptarla al problema y permitir mayor versatilidad para el desarrollo del trabajo. Las principales modificaciones con respecto a la implementación original consistieron en:

- Admite límites diferentes para cada hiperparámetro
- Permite recibir enjambres iniciales (posiciones y velocidades), permitiendo realizar comparaciones controladas para distintas configuraciones
- Permite inicializar velocidades iniciales no nulas, favoreciendo la exploración inicial
- Aplica restricción (clip) en velocidad además de posición, para regular la velocidad de convergencia
- Devuelve el historial completo del *fitness*, lo que permite graficar y analizar la curva de convergencia

La métrica utilizada para evaluar el desempeño del PSO fue el mejor valor de *accuracy* alcanzado al finalizar cada corrida.

2.4 Modelo baseline

Como punto de comparación, se utilizó *RandomizedSearchCV* utilizando el mismo pipeline (estandarización + SVC), misma validación cruzada, misma semilla y mismos límites de búsqueda de C y γ . Este procedimiento se tomó como *baseline* para evaluar la efectividad del PSO en la optimización de hiperparámetros.

3. Resultados

3.1 Análisis del factor de inercia (w)

En la figura 1 se presenta la curva de convergencia del PSO para tres valores del factor de inercia w (0.3, 0.7 y 0.9), utilizando el mismo enjambre inicial (posiciones y velocidades) y la misma semilla aleatoria. Si bien el resultado presenta variaciones entre ejecuciones debido al carácter estocástico del algoritmo, se observa un comportamiento esperable de las curvas en función de w .

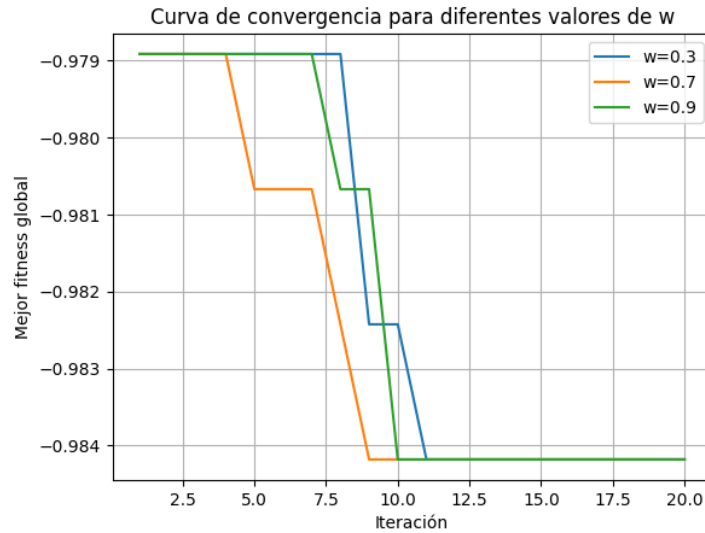


Figura 1. Curva de convergencia de PSO para distintos valores de w .

El parámetro w controla el compromiso entre exploración y explotación del enjambre: valores bajos de w (0.3) favorecen la explotación temprana (movimientos más cortos y rápidos hacia el mejor global), mientras que valores altos de w (0.9) promueven la exploración (movimientos más amplios que permiten recorrer con mayor amplitud el espacio de búsqueda antes de estabilizarse). El valor intermedio de $w=0.7$ muestra un comportamiento más balanceado.

Dado que se trata de un problema de baja dimensionalidad y con una superficie de búsqueda suave, la convergencia se alcanza rápidamente en todos los casos. Sin embargo, las dinámicas son diferentes: curvas más suaves y estables para $w=0.7$ y trayectorias más oscilatorias para $w=0.9$. En los tres escenarios se alcanza un valor máximo de *accuracy* de 0.9842, superior al obtenido por el *baseline* (0.9789).

Durante el desarrollo de este punto, surgieron algunos puntos importantes que llevaron a las modificaciones del algoritmo inicial descritas en el punto 2.3. Por un lado, fue necesario adaptarlo para recibir un enjambre inicial de partículas y velocidades, de manera de utilizar el mismo en todas las corridas y que la comparación tuviera sentido. Por otro lado, al tratarse de un problema de baja complejidad, la convergencia se alcanzaba rápidamente y no se veía el impacto del factor de inercia en las curvas. La introducción de las modificaciones permitió obtener una velocidad más lenta de convergencia, de manera de observar más adecuadamente la diferencia entre las curvas.

3.2 Comparación de resultados entre modelos baseline y PSO

La figura 2 muestra las combinaciones óptimas de C y γ halladas mediante PSO y mediante *Randomized Search*, sobre un mapa de calor de *accuracy*. Se observa que las soluciones encontradas por PSO se concentran en una misma región de alta *accuracy*, mientras que el *baseline* se ubica más alejado de dicha zona. Esto respalda la capacidad del PSO para identificar hiperparámetros más cercanos al óptimo global.

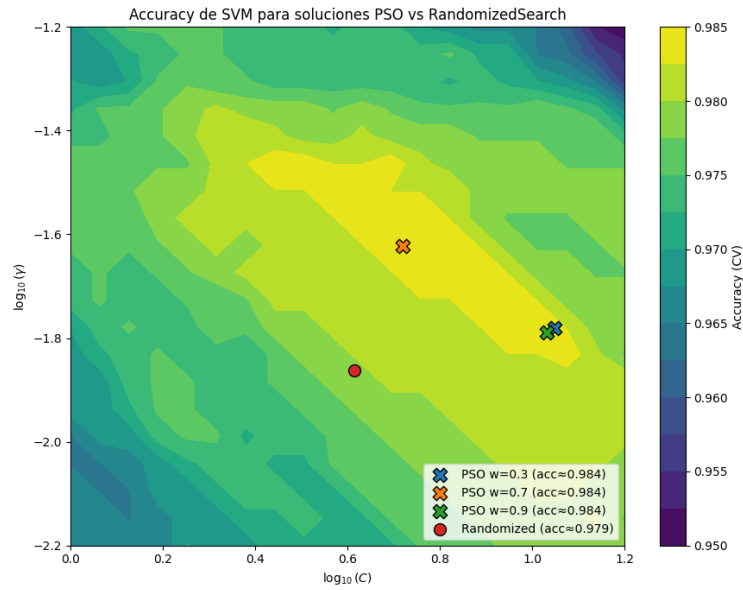


Figura 2. Mapa de calor de accuracy en función de combinaciones de C y γ halladas mediante RandomizedSearch y PSO con distintos valores de w .

3.3 Análisis de variabilidad de resultados en algoritmo PSO

Con el objetivo de evaluar robustez, se realizaron 20 corridas independientes del algoritmo PSO para cada valor de w , inicializando el enjambre de manera aleatoria en cada ejecución. En la figura 3 se presenta un boxplot del *accuracy* final y en la tabla 1 se resumen el valor mínimo, máximo, promedio y desvío estándar obtenidos en cada caso.

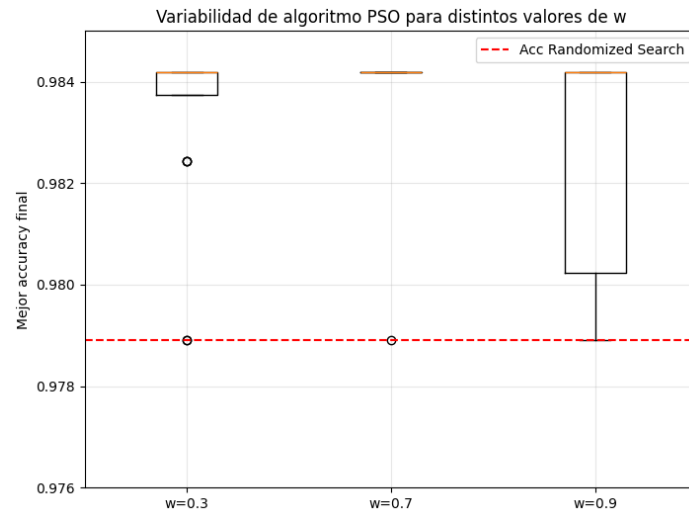


Figura 3. Variabilidad del algoritmo PSO para distintos valores de w (20 ejecuciones).

w	Mínimo	Promedio	Máximo	Desviación Estándar
0.3	0.9789	0.9834	0.9842	0.0016
0.7	0.9789	0.9839	0.9842	0.0011
0.9	0.9789	0.9826	0.9842	0.0023

Tabla 1. Tabla comparativa del algoritmo PSO para distintos valores de w (20 ejecuciones).

Los resultados muestran que, aunque las medianas son similares para los tres valores de w , la dispersión es considerablemente menor en el caso $w=0.7$, confirmando que este valor ofrece el mejor equilibrio entre estabilidad y desempeño. Valores bajos de w tienden a converger prematuramente, mientras que valores altos muestran mayor variabilidad debido a una exploración más intensa. En la mayoría de las corridas, PSO supera al *Randomized Search*, lo cual refuerza su efectividad para este problema.

4. Conclusiones y discusión

En este trabajo se abordó la optimización de hiperparámetros de un modelo Support Vector Machine para clasificación, empleando el algoritmo *Particle Swarm Optimization (PSO)* sobre el dataset **Breast Cancer Wisconsin**, utilizando *accuracy* como métrica objetivo. Los resultados se compararon contra un *baseline* obtenido mediante *Randomized Search*.

Se desarrolló una implementación propia para PSO, sin utilizar librerías específicas como *pyswarm*, para contar con mayor control sobre el algoritmo y facilitar el análisis de su comportamiento.

Se estudió el impacto del factor de inercia (w) en la curva de convergencia del algoritmo. Se observó que valores bajos de w favorecen la explotación temprana, produciendo trayectorias más cortas hacia el óptimo, mientras que valores altos promueven una exploración más amplia y mayor variabilidad entre corridas. Con $w=0.7$ se alcanzó el mejor compromiso entre exploración y explotación, logrando además el mayor valor de *accuracy* (0.9842).

La comparación visual de las soluciones en un mapa de calor evidenció que todas las ejecuciones de PSO convergen hacia la misma región de alta *accuracy*, mientras que la solución obtenida por *Randomized Search* se encuentra más alejada del óptimo.

Dada la variabilidad encontrada en las diferentes corridas del algoritmo, se realizaron 20 corridas independientes para cada valor de w , analizando la distribución de los mejores resultados obtenidos. Los experimentos mostraron que tanto w bajos como altos presentan mayor dispersión, mientras que $w=0.7$ ofrece mayor estabilidad y el mejor desempeño promedio, confirmando la importancia de un equilibrio exploración–explotación. Además, se observó que PSO superó a *Randomized Search* en la gran mayoría de las corridas.

Como trabajo a futuro, podrían explorarse otras métricas de evaluación en lugar de *accuracy* (como precisión, *recall* o *f1-score*), analizar variantes del algoritmo utilizando un factor de inercia adaptativo y aplicar la metodología a problemas de mayor dimensión o modelos más complejos.

En conclusión, si bien el caso analizado es relativamente simple, el trabajo permitió establecer una metodología para la aplicación de PSO en la optimización de hiperparámetros de modelos de clasificación, sentando las bases para su utilización en escenarios más exigentes.