

Práctica de Laboratorio MLP y SVM CComp9-1

Ricardo Manuel Lazo Vásquez
ricardo.lazo@ucsp.edu.pe

Universidad Católica San Pablo

1 Introducción

La clasificación en datos de multiples dimensiones se ha vuelto mas importante con el actual crecimiento en grandes volúmenes de datos, así como sus diversas aplicaciones en el mercado. En el presente trabajo se clasificara dos datasets (Iris y Enfermedad Cardíaca) con k-folds de 3 usando un MLP y SVM. El trabajo esta dividido de la siguiente manera en la sección 2 se hablara de la implementación, en la sección 3 se explicaran los experimentos y se mostraran los resultados; finalmente en la sección 4 se daran las conclusiones.

2 Implementación

El código se encuentra disponible en <https://github.com/TheReverseWasp/TIA-SupportVect-and-MLPerceptron> y se organiza en las siguientes carpetas que se muestran en la Tabla 1.

Carpeta	Descripción
./Datos	Carpeta donde se ubican los dos datasets a utilizar: iris.csv y Enfermedad_Cardiaca.csv.
./Implementacion	Carpeta donde se ubican los scripts de la implementación.
./Instrucciones	Carpeta donde se ubican el pdf con las especificaciones para este trabajo.
./Resultados	Carpeta donde se almacenan los resultados mostrados en este informe.

Table 1: Carpetas del Github.

Así también en la Tabla 2 se explica los archivos de la carpeta Implementación. la ejecución para replicar los experimentos se da con la siguiente línea de código:

```
python3 main.py
```

Donde mostrará un menú de selección para ejecutar los dos experimentos de manera independiente y guardar los resultados de manera automática una vez finalizados.

Script .py	Detalles
complementos.py	Funciones complementarias para la librería my_lib.py
my_lib.py	Script con las funciones a evaluar en esta implementación.
experimentos.py	Implementación de los dos experimentos a evaluar.
main.py	Archivo donde se lanza el menú de selección de experimentos
cross_calc.py	Script semimanual para hallar la validación cruzada.

Table 2: Scripts de implementación.

3 Experimentos y Resultados

3.1 Experimento 1

El experimento consiste en utilizar k-folds para y la implementación de MLP para obtener un promedio de accuracy en ambos datasets. Los resultados del experimento para el dataset de enfermedades cardiacas esta en 3 y para el dataset de iris esta en la tabla 4.

	0.7	0.4	0.1	0.07	0.05	0.01
500	1.0	0.9375	0.6875	0.5625	0.5625	0.5
1000	1.0	1.0	0.875	0.6875	0.5625	0.5
1500	0.9375	0.75	0.8125	0.625	0.75	0.5
2000	0.9375	0.75	0.6875	0.75	0.5625	0.5
2500	0.9375	0.9375	0.625	0.875	0.5625	0.5
3000	0.875	0.875	0.8125	0.75	0.5625	0.5
3500	0.9375	0.9375	0.8125	0.75	0.8125	0.5

Table 3: Accuracy en la data de Enfermedad Cardiaca

	0.7	0.4	0.1	0.07	0.05	0.01
500	0.6282	0.561	0.3511	0.3148	0.2092	0.1721
1000	0.622	0.5973	0.4129	0.345	0.2462	0.1721
1500	0.6467	0.5294	0.3257	0.2462	0.2462	0.1721
2000	0.6776	0.5795	0.3998	0.3635	0.2092	0.1721
2500	0.5726	0.5926	0.3889	0.2462	0.2277	0.1721
3000	0.6158	0.5911	0.4129	0.2585	0.2092	0.1906
3500	0.6652	0.598	0.3627	0.2901	0.2153	0.1721

Table 4: Accuracy en la data de Iris

Para esta implementación se utilizó un MLP de 4 capas usando la función sigmoidea, en los resultados se puede destacar que la arquitectura tiene mejor performance en ambos datasets cuando el learning rate es un poco alto, además se puede destacar que para el dataset de iris se podría mejorar aumentando las capas del perceptron.

3.2 Experimento 2

Para el segundo experimento se debe utilizar una SVM con diferentes kernels y diferentes parametros de regularización, esto con el fin de encontrar los mejores parametros de entrenamiento usando k-folds buscando el mejor accuracy.

Los resultados usando SVM con los kernels lineales, poly, rbf y sigmoid, con indice de regularizacion de 1, 100, 500 y 1000 en ambos datasets se encuentran en las Tablas 5 y 6.

	C=1	C=100	C=500	C=1000
linear	0.9647	0.9674	0.9701	0.9647
poly	0.3234	0.9511	0.9647	0.9647
rbf	0.3152	0.962	0.9701	0.9701
sigmoid	0.5652	0.3668	0.3668	0.3668

Table 5: Accuracy en la data de Enfermedad Cardiaca

	C=1	C=100	C=500	C=1000
linear	1.0	1.0	1.0	1.0
poly	1.0	1.0	1.0	1.0
rbf	1.0	1.0	1.0	1.0
sigmoid	0.0638	0.0144	0.0144	0.0144

Table 6: Accuracy en la data de Iris

En el caso del dataset de iris se optiene un mejor resultado que en el dataset de enfermedades cardiacas, en el dataset de enfermedades cardiacas se obtiene mejor performance en los tres primeros kernels al igual que en el de iris, y en el caso del kernel sigmoide el dataset de enfermedades cardiacas supera al de iris.

4 Conclusiones

References

1. Andrew Ng. Specialization in Deep Learning.
<https://www.coursera.org/specializations/deep-learning>