Práctica 6 - parte I
Categorización de imágenes de histopatología

Mario Berrios Carmona

Fundamentos de Sistemas Inteligentes en Visión
Grado en Ingeniería Informática

Diciembre 2022

Resumen

En este documento se detallan los diferentes algoritmos y extractores de características utilizados para la realización de la práctica 6 en la asignatura de Fundamentos de Sistemas Inteligentes en Visión. Se muestra a su vez la evolución de dichos algoritmos a medida que los hiperparámetros van cambiando y su efecto en la precisión de las predicciones

1. Comandos a probar

A continuación se detallan una serie de comandos simples para probar directamente el funcionamiento de cada uno de los métodos así como con diferentes extractores e hiperparámetros. Todos estos comandos deben ser ejecutados desde `/i12bercm/histopathology/`. Para que funcionen correctamente los datasets deben estar directamente en `/i12bercm/histopathology/data/`. En el directorio `data/` se guardarán los modelos entrenados.

1) Entrenamiento KNN con *LBP*
```
./build/train_clf --knn_K=25 -f=3 -v=20 ./data/train_labels.txt ./data/k_nn_25_LBP
```

2) Test KNN con *LBP*
```
./build/test_clf ./data/test_labels.txt ./data/k_nn_25_LBP
```

3) Entrenamiento SVM con *Histograma en HSV*
```
./build/train_clf --clf=1 -f=2 --svm_C=0.5 --svm_G=1 --svm_K=2 -v=20 ./data/train_labels.txt
./data/svm_C0-5_D3_G1_K2_Hist
```

4) Test SVM con *Histograma en HSV*
```
./build/test_clf ./data/test_labels.txt ./data/svm_C0-5_D3_G1_K2_Hist
```

5) Entrenamiento RF con *Histograma en HSV*
```
./build/train_clf --clf=2 -f=2 --rtrees_E=01 --rtrees_T=50 -v=20 ./data/train_labels.txt
./data/rf_E01_T50_Hist
```

6) Test RF con *Histograma en HSV*
```
./build/test_clf ./data/test_labels.txt ./data/rf_E01_T50_Hist
```

2. Extractores de características

En este apartado se comentará los distintos extractores de características que se han implementado a lo largo de la práctica. Se describe también el extractor implementado por defecto, ya que se utilizará más adelante para la comparativa de resultados.

2.1. Nivel de gris

Este extractor realiza un cambio de color a escala de grises un cambio de tamaño a 16×16. Estos niveles funcionan como el descriptor de la imagen.

2.2. Histograma en HSV

Este extractor realiza un cambio en el espacio de colores de RGB a HSV. Se realiza a continuación el histograma de cada uno de los canales de la imagen y se concatenan, siendo esta concatenación el descriptor de la imagen.

2.3. Patrón binario local (LBP)

Este extractor realiza el histograma del LBP de la imagen en cada uno de los canales, siendo la concatenación de estos el descriptor de la misma.

3. Algoritmo K vecinos más cercanos (K-NN)

Este algoritmo trata de clasificar una imagen según los descriptores a los que más se asemeje. El hiperparámetro K que utiliza este método indica el número que de descriptores que más se asemejan que deben tenerse en cuenta para clasificar el descriptor de entrada.

3.1. Análisis de variación de hiperparámetro *K*

Cuando $K = 1$ el modelo está sobreentrenado, ya que la precisión en el dataset de entrenamiento es 1. A medida que K va aumentando esta precisión en el conjunto de entrenamiento va reduciéndose, mientras que en el conjunto de validación va aumentando. A continuación se muestra las gráficas correspondientes a cada uno de los extractores de características indicadas en la sección anterior.

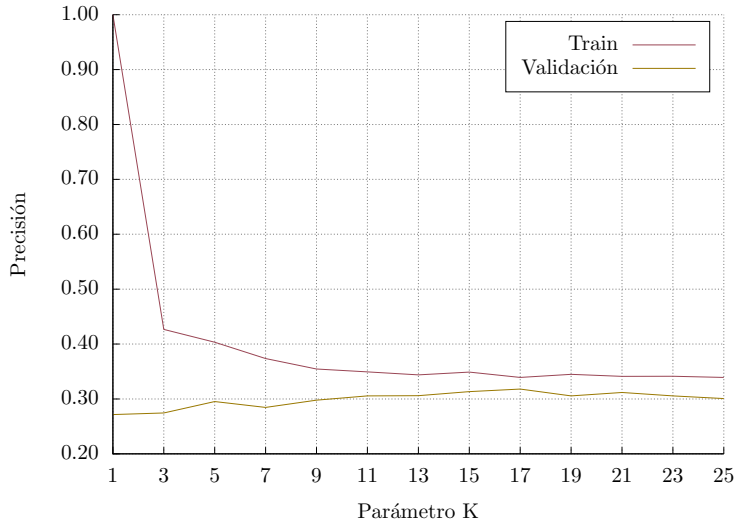


Figura 1: Evolución de KNN con el extractor *Nivel de gris*

El valor en test más alto conseguido con el extractor *Nivel de gris* se da cuando $K = 23$ siendo la precisión 0,362319.

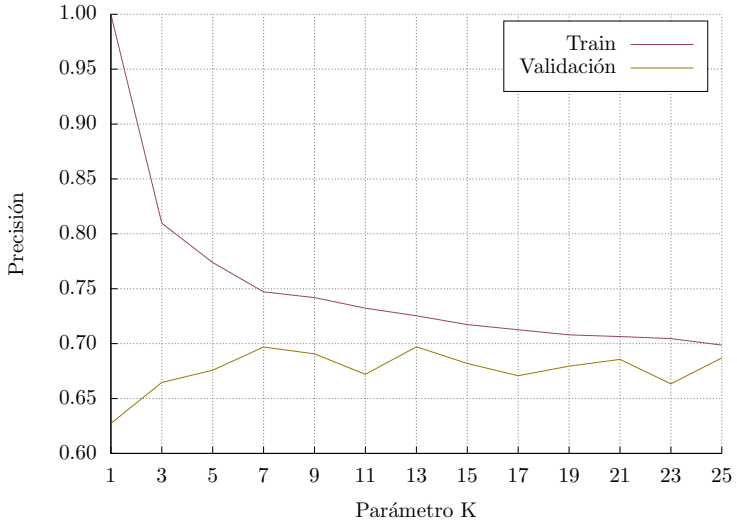


Figura 2: Evolución de KNN con el extractor *Histograma en HSV*

El valor en test más alto conseguido con el extractor *Histograma en HSV* se da cuando $K = 9$ siendo la precisión 0,70173.

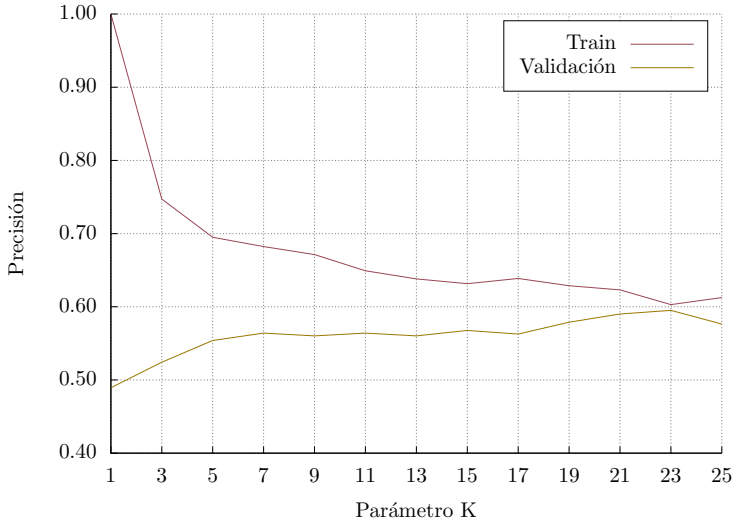


Figura 3: Evolución de KNN con el extractor *LBP*

El valor en test más alto conseguido con el extractor *LBP* se da cuando $K = 17$ siendo la precisión 0,592333.

4. Máquina de soporte vectorial (SVM)

Este algoritmo intenta separar los grupos de descriptores mediante una superficie limitadora. El hiperparámetro C indica la importancia que tiene cometer un error, a mayor valor mayor importancia se le da a un error.

Cuando los grupos de descriptores no pueden ser separados por una linea recta existen distintas formas de cambiar la linea limitadora. Para ello tenemos el hiperparametro *tipo de kernel*, el cual permite usar funciones no lineales para separar.

Los kernels que se usarán en esta práctica son el polinomial y el RBF. El primero tiene el hiperparametro *Grado* que determina el grado del polinomio. RBF tiene el hiperparametro *Gamma*, el cual cambia el valor de *gamma* en la formula $exp(\gamma(x^2 + y^2))$

4.1. Análisis de variación de hiperpámetros *C* y *tipo de kernel* (*Grado*, *Gamma*, etc)

En esta sección se recoge la evolución de la precisión según va cambiando los distintos hiperparámetros que SVM contempla. Se ha utilizado siempre el extractor *histograma en HSV* ya que es el que mejor resultado ha tenido. En todas esta gráficas se ha utilizado los valores $C = \{0{,}1, 0{,}5, 1, 20, 100\}$ (el eje de abcisas de las gráficas se muestra en una escala logarítmica).

La primera gráfica (figura 4) explica el desarrollo cuando el hiperparámetro $K = 0$ (función lineal). Se observa como el entrenamiento y la validación tienen unos valores similares a lo largo de la curva. Es importante también indicar que el parámetro C no tiene mucho peso en la precisión, principalmente porque la división entre el set de entrenamiento y validación influye mucho en el resultado. Esto se puede ver en todas las gráficas de este apartado.

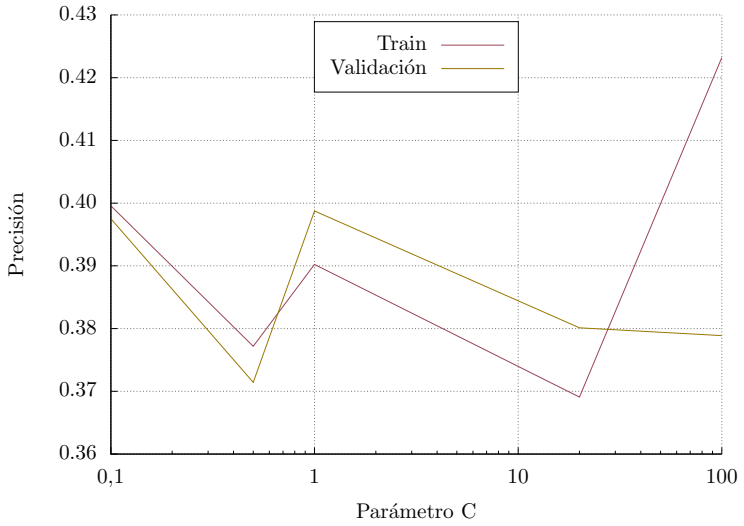
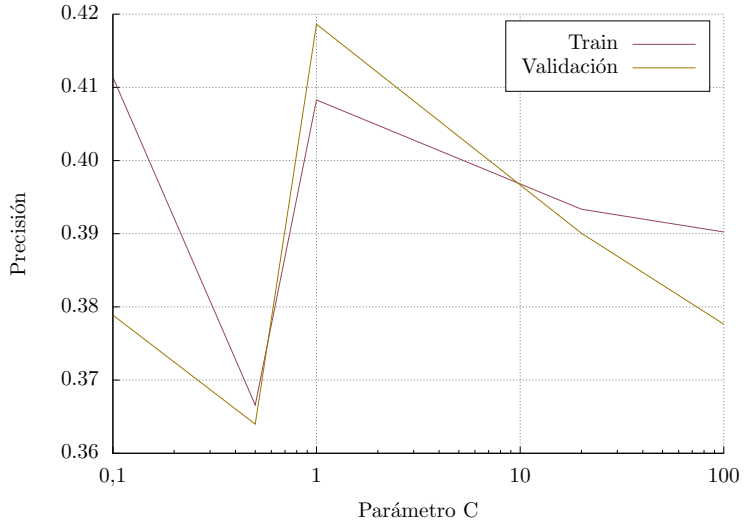


Figura 4: Evolucion de SVM con el hiperparámetro $K = 0$ (función lineal)

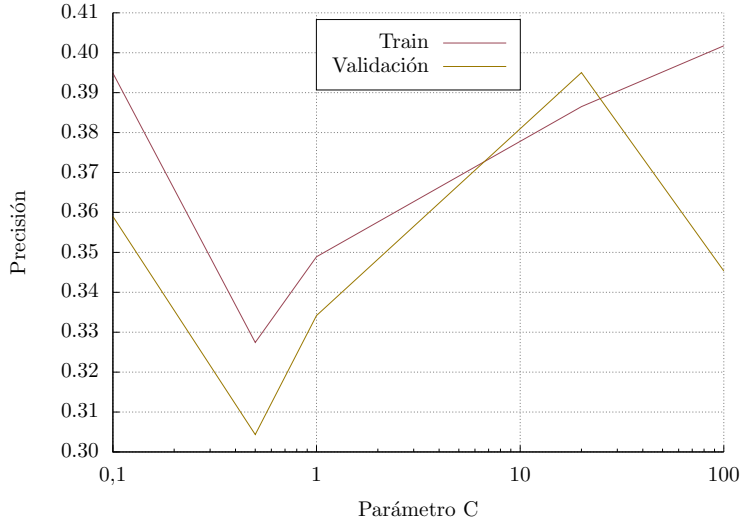
Cuando $K = 1$ (función polinómica) entra en juego el valor del hiperparámetro D. Se ha dividido los valores de dicho parámetro en tres gráficas distintas para una mejor comprensión (figuras 5a, 5b y 5c).

Cuando $K = 2$ (RBF) entra en juego el valor del hiperparámetro G. Se ha dividido los valores de dicho parámetro en tres gráficas distintas para una mejor comprensión (figuras 6a, 6b y 6c). Es intersante cómo se observa que al llegar al valor $C = 1$ el modelo sobreentrena, haciendo que todos los resultados cuando $C > 1$ también estén sobreentrenados.

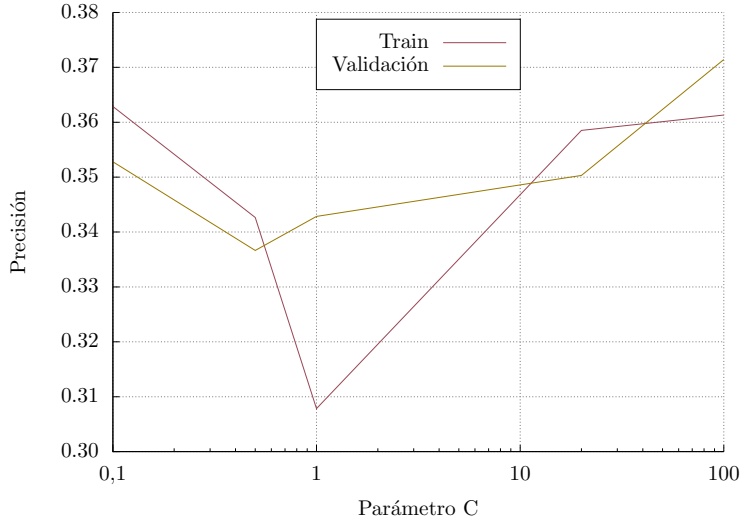
Este apartado tiene solo 5 valores por gráfica. Esto se debe a la cantidad de hiperparámetros que podemos cambiar además del tiempo que consume cada una de las ejecuciones posibles, repitiendo cada una de ellas para poder sacar un valor medio en cada caso.



(a) Hiperparámetro $D = 3$

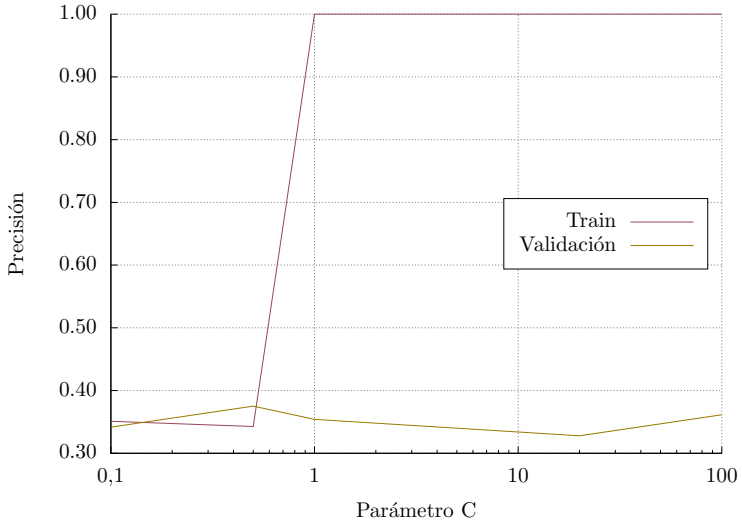


(b) Hiperparámetro $D = 4$

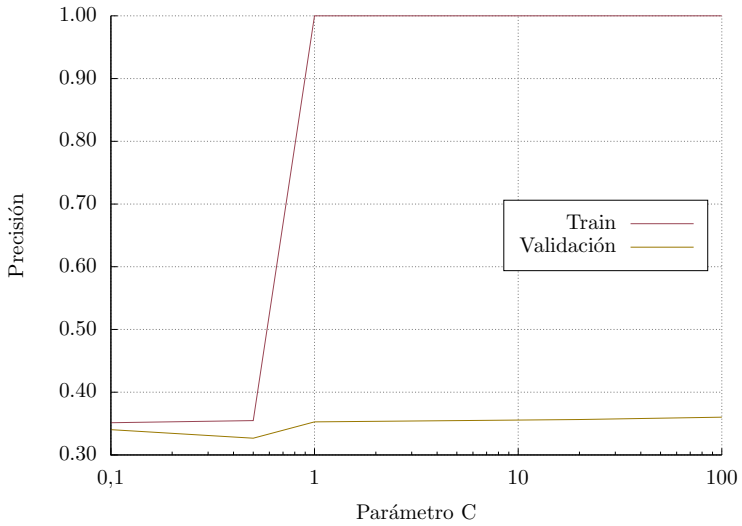


(c) Hiperparámetro $D = 5$

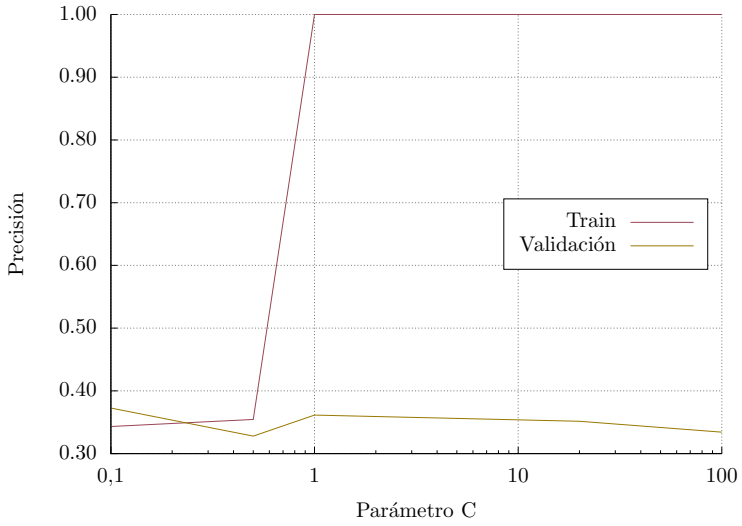
Figura 5: Evolución SVM con el hiperparámetro $K = 1$ (función polinómica)



(a) Hiperparámetro $G = 0{,}5$



(b) Hiperparámetro $G = 1$



(c) Hiperparámetro $G = 2$

Figura 6: Evolución con el hiperparámetro $K = 2$

5. Árboles de decisión (Random Forest)

Este algoritmo utiliza árboles de decisiones para realizar la predicción. Para hacer la división en cada nivel se selecciona una característica al azar de entre un grupo. El número de características que forman cada grupo se indica mediante el hiperparámetro *ActiveVarCount*. Por otro lado, el número de árboles máximo que se van a utilizar se indica mediante el hiperparámetro *MaxTree* y el error OOB se indica mediante el hiperparámetro homónimo.

5.1. Análisis de variación de hiperpámetros *ActiveVarCount*. *MaxTree* y *OOB*

Una vez más se ha utilizado únicamente el extractor *histograma en HSV* debido al buen resultado que proporciona. Es interesante ver cómo el cambio del error OOB o el incremento de árboles no affecta en gran cantidad a la precisión conseguida.

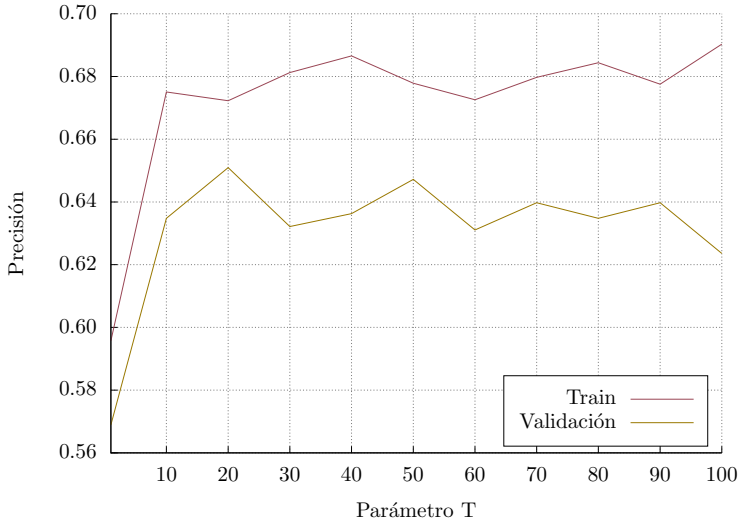


Figura 7: Evolucion de RF con el hiperparámetro $E = 0{,}1$

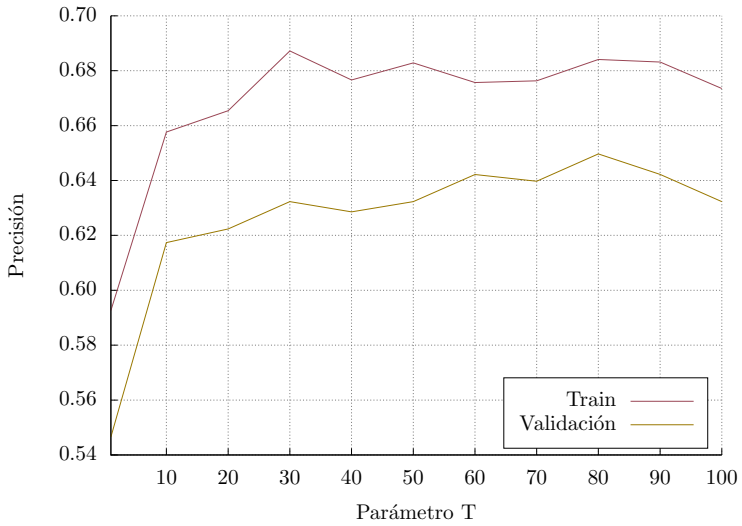


Figura 8: Evolucion de RF con el hiperparámetro $E = 0{,}01$