

Bootstrap y Modelos Ocultos de Markov (HMM) para Clasificación de Flechas 9x9

Mariana Belén Cruz Rodríguez

Análisis de Notebook

Septiembre 2025

Objetivo del Proyecto

El objetivo es clasificar imágenes binarias (blanco y negro) de 9x9 píxeles en cuatro categorías distintas de flechas:

- Arriba/up
- Abajo/down
- Izquierda/left
- Derecha/right



Figure: Ejemplo de una imagen de flecha con ruido.

Para esta tarea de clasificación, se utiliza un enfoque basado en **Modelos Ocultos de Markov (HMM)**.

Se crearon cuatro "imágenes perfectas" de 9x9 píxeles, representadas como matrices binarias, para cada dirección de flecha.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

El resto de las flechas se obtienen al rotar la anterior.

Dataset sintético

Para simular imágenes imperfectas, se generó un conjunto de datos sintético añadiendo "ruido" a los patrones perfectos.

- **Proceso:** Se invierte el valor de cada píxel (0 a 1, o 1 a 0) con una probabilidad p .
- **Parámetros del experimento:**
 - Nivel de ruido: $p = 0.05$
 - Muestras de entrenamiento por clase: $N_{train} = 600$
 - Muestras de prueba por clase: $N_{test} = 100$

Modelos Ocultos de Markov (HMM)

Se entrenó un HMM independiente para cada una de las cuatro clases de flechas.

- **Estructura:** Las matrices de 9×9 se aplanan en secuencias de 81 observaciones para alimentar los modelos.
- **Parámetros del HMM:**
 - $K = 8$ estados ocultos.
 - $M = 2$ símbolos de observación (0 o 1).
- **Entrenamiento (Estimación de Parámetros):** Se utilizó el algoritmo de **Baum-Welch** para aprender las matrices de transición de estados (A), emisión (B), y las probabilidades iniciales (π).
- **Clasificación (Evaluación):** Para una nueva imagen, se calcula la verosimilitud (log-likelihood) con el **algoritmo Forward** para cada uno de los 4 modelos. La clase del modelo con la verosimilitud más alta es la predicción final.

Rendimiento del Modelo Inicial

Se entrenaron los 4 modelos HMM con el conjunto de datos de entrenamiento y se evaluaron en el conjunto de prueba.

Resultados en el Conjunto de Prueba

Clase	Precisión (Accuracy)
1 (Arriba)	80.0%
2 (Abajo)	24.0%
3 (Izquierda)	87.0%
4 (Derecha)	97.0%
Total	72%

Table: Precisión por clase y global.

Observación

El rendimiento varía significativamente entre clases. El modelo puede confundirse por la simetría existente entre arriba/abajo y derecha/izquierda.

Bootstrap para Estimar la Incertidumbre

Para entender la estabilidad y variabilidad de los resultados, se aplicó la técnica de **bootstrap**.

- **Procedimiento:**

- ① Se realizaron $B = 200$ iteraciones.
- ② En cada iteración, se creó un nuevo conjunto de entrenamiento mediante remuestreo con reemplazo del original.
- ③ Se re-entrenaron los 4 modelos HMM con este nuevo conjunto.
- ④ Se evaluó el rendimiento en el conjunto de prueba **original (fijo)**.

- **Objetivo:** Generar una distribución de la precisión del modelo para calcular la media y los intervalos de confianza.

Resultados del Bootstrap

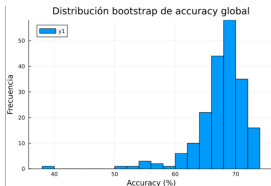


Figure: Distribución de la precisión global en 200 iteraciones de bootstrap.

Precisión Media:

- La precisión media obtenida tras las 200 ejecuciones fue de **67.38%**.

Intervalos de Confianza por Clase

El bootstrap también nos permite estimar la variabilidad del rendimiento para cada clase.

Intervalos de Confianza del 95%

Clase	IC 95% (Percentiles)
1 (Arriba)	[49.6%, 97%]
2 (Abajo)	[25%, 92.1%]
3 (Izquierda)	[74%, 97%]
4 (Derecha)	[77%, 92%]

Table: IC de la precisión para cada clase. La gran amplitud sugiere una alta variabilidad en el rendimiento dependiendo de la muestra de entrenamiento.

Resumen y Próximos Pasos

- Se implementó con éxito un clasificador de imágenes de flechas utilizando un HMM por clase, logrando una precisión promedio de 72%.
- El rendimiento es modesto y muestra una alta variabilidad, especialmente para la clase "Abajo" y "Arriba", lo que indica que el modelo es sensible a las muestras de entrenamiento.
- La técnica de bootstrap fue útil para cuantificar esta incertidumbre y proporcionar un rango más realista del rendimiento esperado del modelo.

¿Se puede mejorar?

Posibles Mejoras

- Explorar diferentes topologías de HMM o un número distinto de estados ocultos (K).
- Aumentar la cantidad y variabilidad de los datos de entrenamiento.
- Utilizar técnicas de extracción de características más sofisticadas en lugar de usar los píxeles en crudo.

Es importante considerar que el HMM no es muy bueno para esta clasificación ya que al "aplanar" las imágenes se pierde mucha información espacial, funciona significativamente mejor el uso de redes convolucionales 2D, que clasifican con mayor precisión