Método de evaluación para modelo integrante de un sistema clasificador (Análisis de peor caso)

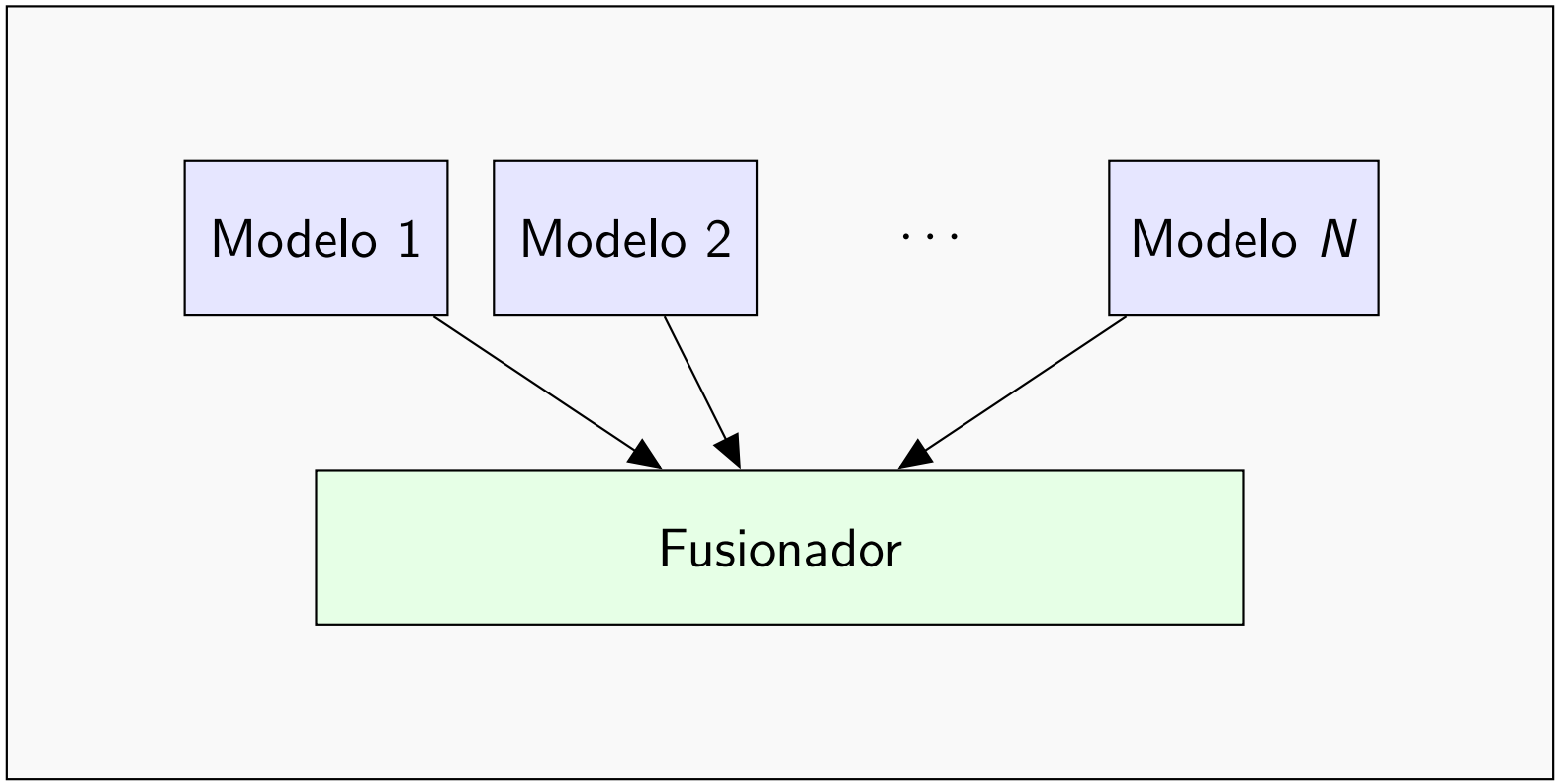
Introducción

Se tiene un sistema clasificador con distintos modelos en paralelo. Cada uno de esos modelos mira features de los datos y devuelve una clase. Luego se fusionan sus resultados mediante algún método, resultando en que el sistema determina la clase del dato de forma unívoca.

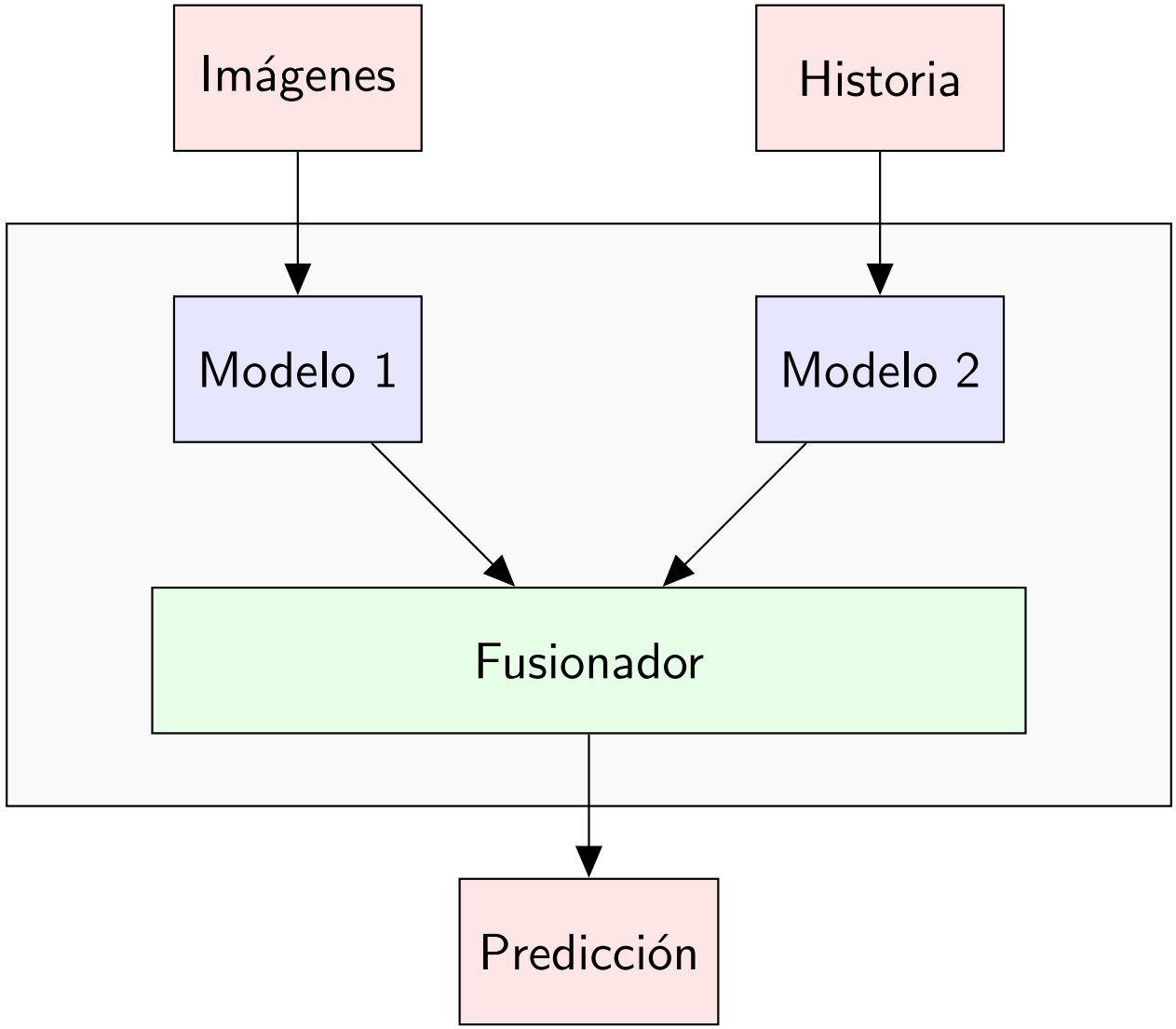
Particularmente, el sistema resuelve problemas de clasificación binaria.

Introducción

Sistema Clasificador



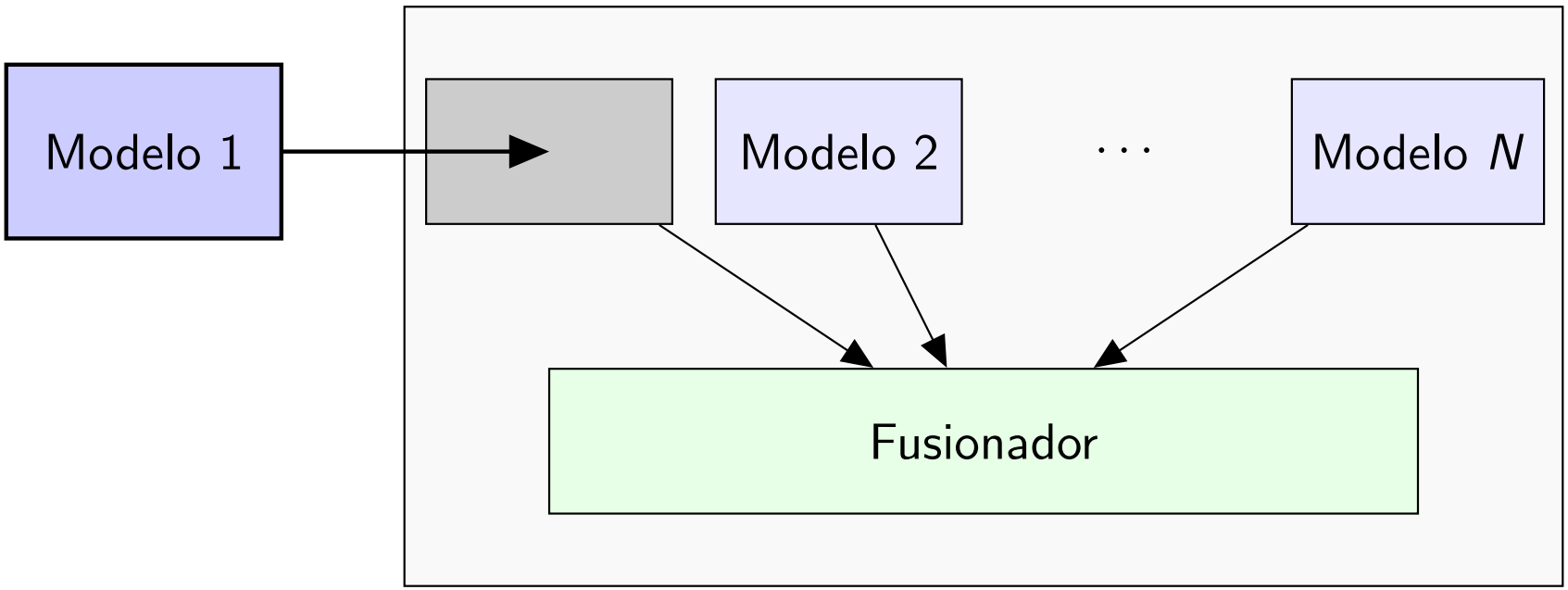
Sistema ejemplo



Objetivo

- Evaluar un modelo de forma independiente al sistema
- Obtener una estimación útil del comportamiento que tendrá el sistema con ese modelo

Objetivo



Información disponible

Datos de los que se dispone para el problema:

- El sistema incompleto, faltando un modelo
- Una función de costo para el sistema
- Datos para la evaluación del sistema
- El modelo a evaluar
- Datos para la evaluación del modelo

Información disponible

- Ambos conjuntos de datos pueden o no ser el mismo.
- La función de costo del modelo es la que se busca proponer, y es dependiente de la función de costo del sistema y de la influencia del modelo en el sistema.

Motivación

Una posible utilidad de una función de costo para el modelo que sea independiente del sistema puede ser realizar búsqueda de hiperparámetros.

Se puede querer reemplazar o añadir un modelo al sistema. Al momento de realizar la búsqueda de hiperparámetros en el modelo, con un costo para el modelo independiente al sistema se puede comparar entre modelos de forma mucho mas rápida y asi elegir a los más adecuados.

Intervención

Se va a utilizar una operación sobre el sistema llamada intervención, basada en la intervención de las redes bayesianas.

Al sistema sin el modelo se le inserta en ese lugar un modelo simple que hace siempre lo mismo a pesar del dato de entrada para poder analizar su comportamiento.

Intervención

Un ejemplo de intervención podría ser correr el sistema diciendole que el modelo devuelve siempre Positive. Luego se podría calcular la matriz de confusión asociada a sus resultados.

La otra intervención que se va a usar es siempre Negative.

Estimación del costo del sistema

Para poder comparar entre modelos se separa el proceso en dos partes:

- Primero es evaluado el sistema sin el modelo, usando la intervención. Se registran las probabilidades de acierto en el sistema intervenido con un modelo que da siempre positive, y uno que da siempre negative.
- Luego usando esos valores, se ejecuta el modelo de forma independiente para calcular alguna función de costo asociada.

Estimación del costo del sistema

- Idealmente, se buscaría estimar de forma precisa el resultado de la ejecución del sistema si se hubiera corrido con el modelo.
- Esto es imposible ya que dados modelos distintos la misma matriz de confusión pueden llevar a resultados muy distintos en el sistema segun su predicción en cada instancia.
- Por lo tanto la métrica a evaluar va a ser, dado un sistema y dada la matriz de confusión de un modelo, estimar el peor caso del sistema ejecutado con ese modelo.

Problema 1

El primer problema a resolver entonces va a ser encontrar una función de costo para el modelo con las siguientes condiciones:

- La función devuelve el peor caso del sistema dado que fuera ejecutado con el modelo
- Se tiene el resultado de ejecutar el sistema con intervención Positive y Negative en el modelo
- Los datasets para evaluar el modelo y el sistema son el mismo

Notación

Para definir el problema formalmente se introduce la notación $m_{ij}^{f,D}$ para denotar un valor de una matriz de confusión:

$$m_{ij}^{f,D} = |\{d \in D \mid f(d_1) = i \wedge d_2 = j\}|$$

- i es la clase real
- j es la clase que predice el modelo f sobre el dato
- D es el dataset utilizado, conteniendo un conjunto de pares d_1 instancia, d_2 clase real asociada, y su valor por defecto que representa todos los datos disponibles es D
- f es el modelo usado y puede valer:
  - M como el modelo a analizar
  - S como el sistema completo con el modelo
  - $S_{*\to x}$ como el sistema con un modelo intervenido que predice de forma constante la clase x para cualquier instancia

Especificación del problema

Se asume lo siguiente sobre el comportamiento del sistema para cualquier conjunto de instancias D, modelo M y sistema S:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = n\} \subseteq$
  $\{d \in D \mid d_2 = p \wedge S_{*\to n}(d_1) = n\}$

\\ Agregar evidencia solo puede mejorar la predicción

Especificación del problema

Se asume lo siguiente sobre el comportamiento del sistema para cualquier conjunto de instancias D, modelo M y sistema S:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = n\} \subseteq$
  $\{d \in D \mid d_2 = p \wedge S_{*\to n}(d_1) = n\}$

\\ Restar evidencia solo puede empeorar la predicción

Especificación del problema

Se asume lo siguiente sobre el comportamiento del sistema para cualquier conjunto de instancias D, modelo M y sistema S:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq \{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$
- $\{d \in D \mid d_2 = p \wedge S(d_1) = n\} \subseteq \{d \in D \mid d_2 = p \wedge S_{*\to n}(d_1) = n\}$
- $\{d \in D \mid d_2 = n \wedge S(d_1) = n\} \subseteq \{d \in D \mid d_2 = n \wedge S_{*\to n}(d_1) = n\}$
- $\{d \in D \mid d_2 = n \wedge S(d_1) = p\} \subseteq \{d \in D \mid d_2 = n \wedge S_{*\to p}(d_1) = p\}$

\\ Lo mismso para Clase Negative

Especificación del problema

Se espera lo siguiente sobre la función de costo asociada a las confusiones del sistema:

- $Costo^S_{TP} \leq Costo^S_{FN}$
- $Costo^S_{TN} \leq Costo^S_{FP}$

Los fallos tienen mayor costo que los aciertos.

Para este trabajo se va a asumir que los aciertos tienen costo 1 y los fallos costo 0.

Especificación del problema

Datos disponibles:

	Pred Pos	Pred Neg
Clase Pos	$m_{TP}^{S_{*\to p},D}$	$m_{FN}^{S_{*\to p},D}$
Clase Neg	$m_{FP}^{S_{*\to p},D}$	$m_{TN}^{S_{*\to p},D}$

Table: Matriz de confusión del sistema si Modelo = Positive

	Pred Pos	Pred Neg
Clase Pos	$m_{TP}^{S_{*\to n},D}$	$m_{FN}^{S_{*\to n},D}$
Clase Neg	$m_{FP}^{S_{*\to n},D}$	$m_{TN}^{S_{*\to n},D}$

Table: Matriz de confusión del sistema si Modelo = Negative

Especificación del problema

Otros datos disponibles:

	Positive	Negative
#Instancias	P	N

Table: Cantidad de instancias por clase

	Pred Pos	Pred Neg
Clase Pos	$m_{TP}^{M,D}$	$m_{FN}^{M,D}$
Clase Neg	$m_{FP}^{M,D}$	$m_{TN}^{M,D}$

Table: Matriz de confusión del modelo

Especificación del problema

Dados esos datos, se busca generar la peor matriz de confusión posible del sistema que sea válida (peor caso).

	Prediccion Positive	Prediccion Negative
Clase Positive	$m_{TP}^{S,D}$	$m_{FN}^{S,D}$
Clase Negative	$m_{FP}^{S,D}$	$m_{TN}^{S,D}$

Table: Matriz de confusión del sistema

Especificación del problema

La peor es la que maximiza:

- $m_{TP}^{S,D} * Costo_{TP}^{S} + m_{FN}^{S,D} * Costo_{FN}^{S} + m_{FP}^{S,D} * Costo_{FP}^{S} + m_{TN}^{S,D} * Costo_{TN}^{S}$

Como los costos asociados a las confusiones de aciertos valen 1 y las de los fallos 0, es equivalente a maximizar:

- $m_{FN}^{S,D} + m_{FP}^{S,D}$

El análisis también vale para otros casos.

Planteo inicial

Para lograr esto se va a analizar de forma cercana los comportamientos posibles de las instancias.

Una instancia se puede caracterizar de 4 formas distintas:

1. Sistema acierta independientemente del modelo.
2. Sistema acierta solo si el modelo acierta.
3. Sistema acierta solo si el modelo falla.
4. Sistema falla independientemente del modelo.

Planteo inicial

El problema se va a trabajar como un problema adversarial.

Se sabe que predice el modelo para cada instancia, pero no cuál será el resultado que tendría sistema para la instancia, ya que se desconoce de que tipo es.

El objetivo del adversario va a ser para el conjunto de instancias, y en base al resultado del modelo para cada instancia, asignarle un tipo a cada una de ellas de forma que la cantidad de fallos del sistema sea máxima.

Comportamiento de instancias

La siguiente tabla tiene los posibles casos en los que se puede encontrar una instancia con respecto al resultado del sistema en ella en base al resultado del modelo.

Comportamiento de instancias

Recordando la assumption 2 de la clase Positive:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq \{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$

	$S_{*\to p}(i)$	$S_{*\to n}(i)$	Es Válido
Instancia 1	p	p	
Instancia 2	n	p	
Instancia 3	p	n	
Instancia 4	n	n	

Table: Resultados posibles en instancias de clase Positive

Comportamiento de instancias

Recordando la assumption 2 de la clase Positive:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq$
  $\{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$

	$S_{*\to p}(i)$	$S_{*\to n}(i)$	Es Válido
Instancia 1	p	p	SI
Instancia 2	n	p	
Instancia 3	p	n	
Instancia 4	n	n	

Table: Resultados posibles en instancias de clase Positive

Comportamiento de instancias

Recordando la assumption 2 de la clase Positive:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq$
  $\{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$

	$S_{*\to p}(i)$	$S_{*\to n}(i)$	Es Válido
Instancia 1	p	p	SI
Instancia 2	n	p	NO
Instancia 3	p	n	
Instancia 4	n	n	

Table: Resultados posibles en instancias de clase Positive

Comportamiento de instancias

Recordando la assumption 2 de la clase Positive:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq$
  $\{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$

	$S_{*\to p}(i)$	$S_{*\to n}(i)$	Es Válido
Instancia 1	p	p	SI
Instancia 2	n	p	NO
Instancia 3	p	n	SI
Instancia 4	n	n	

Table: Resultados posibles en instancias de clase Positive

Comportamiento de instancias

Recordando la assumption 2 de la clase Positive:

- $\{d \in D \mid d_2 = p \wedge S(d_1) = p\} \subseteq \{d \in D \mid d_2 = p \wedge S_{*\to p}(d_1) = p\}$

	$S_{*\to p}(i)$	$S_{*\to n}(i)$	Es Válido
Instancia 1	p	p	SI
Instancia 2	n	p	NO
Instancia 3	p	n	SI
Instancia 4	n	n	SI

Table: Resultados posibles en instancias de clase Positive

Comportamiento de instancias

- Las instancias de tipo 1 nunca fallan independientemente del resultado del modelo
- No hay instancias de tipo 2
- Las instancias de tipo 3 solo fallan si el modelo falla
- Las instancias de tipo 4 siempre fallan independientemente del resultado del modelo

Solo las instancias de tipo 3 son en las que el adversario puede decidir para afectar el resultado del sistema.

Comportamiento de instancias

A su vez, el total de instancias de la clase Positive es equivalente a la suma de instancias tipo 1, 3 y 4, ya que de tipo 2 no pueden existir para ningún sistema que cumpla con la especificación.

Estrategia

La estrategia del adversario entonces va a consistir en asignar a las instancias de tipo 3 prioritariamente como casos en los que el modelo falla.

Las instancias de tipo 1 y 4 son irrelevantes para el adversario por lo que le conviene asignarlas a aciertos del modelo.

Fallos en peor caso

Para la clase Positive, la fórmula que da la cantidad de fallos totales basados en esa estrategia es la siguiente:

- InstanciasTipo4 + Min(FallosDelModelo, InstanciasTipo3)

Fallos en peor caso

Para la clase Positive, la fórmula que da la cantidad de fallos totales basados en esa estrategia es la siguiente:

- InstanciasTipo4 + Min($m_{FN}^{M,D}$, InstanciasTipo3)

\\ La cantidad de fallos del modelo es la confusion de Falsos Negativos

Fallos en peor caso

Para la clase Positive, la fórmula que da la cantidad de fallos totales basados en esa estrategia es la siguiente:

- InstanciasTipo4 + Min($m_{FN}^{M,D}$, InstanciasTipo3)
- El modelo falla para todas las instancias de tipo 3 que sea posible

Fallos en peor caso

Para la clase Positive, la fórmula que da la cantidad de fallos totales basados en esa estrategia es la siguiente:

- InstanciasTipo4 + Min($m_{FN}^{M,D}$, InstanciasTipo3)

- El modelo falla para todas las instancias de tipo 3 que sea posible
- Todas las instancias de tipo 4 suman un fallo
- Las instancias de tipo 1 no suman fallos

Fallos en peor caso

Si se pudiera saber la cantidad de instancias de cada tipo disponibles, se tiene la solución al peor caso.

Fallos en peor caso

Instancias de tipo 1:

- Tanto la intervención Positive como Negative del sistema aciertan
- Se saben las confusiones con intervención: $m_{TP}^{S_{*\to p},D}$ y $m_{TP}^{S_{*\to n},D}$
- A su vez, en base a las assumptions, se sabe que las instancias que suman en $m_{TP}^{S_{*\to n},D}$ también están en $m_{TP}^{S_{*\to p},D}$ ya que se está agregando evidencia
- $m_{TP}^{S_{*\to n},D}$ es la cantidad de instancias tipo 1

Fallos en peor caso

Instancias de tipo 4:

- Tanto la intervención Positive como Negative del sistema fallan
- Se saben las confusiones con intervención: $m_{FN}^{S_{*\to p},D}$ y $m_{FN}^{S_{*\to n},D}$
- A su vez, en base a las assumptions, se sabe que las instancias que suman en $m_{FN}^{S_{*\to p},D}$ también están en $m_{FN}^{S_{*\to n},D}$ ya que se está quitando evidencia
- $m_{FN}^{S_{*\to p},D}$ es la cantidad de instancias tipo 4

Fallos en peor caso

Instancias de tipo 3:

- Solo la intervención Positive del sistema acierta
- Son las instancias restantes de la clase Positive

Fallos en peor caso

La cantidad de instancias de tipo 3 es:

- CantidadPositive - $m_{FN}^{S_{*\to p},D}$ - $m_{TP}^{S_{*\to n},D}$
- $(m_{TP}^{S_{*\to p},D} + m_{FN}^{S_{*\to p},D})$ - $m_{FN}^{S_{*\to p},D}$ - $m_{TP}^{S_{*\to n},D}$
- $m_{TP}^{S_{*\to p},D}$ - $m_{TP}^{S_{*\to n},D}$

Esta resta en la cuenta representa las instancias en las que el sistema solo acierta si el modelo acierta, coincidiendo con las de tipo 3.

Fallos en peor caso

Recordando la fórmula de peor caso:

- InstanciasTipo4 + Min($m_{FN}^{M,D}$, InstanciasTipo3)

Reemplazando por las cantidades de instancias de cada tipo:

- $m_{FN}^{S_{*\to p},D}$ + Min($m_{FN}^{M,D}$, $m_{TP}^{S_{*\to p},D}$ - $m_{TP}^{S_{*\to n},D}$)

Fallos en peor caso

Se puede hacer el mismo análisis para la clase Negative, llegando a lo siguiente:

- Clase Positive: $m_{FN}^{S_{*\to p},D} + \text{Min}(m_{FN}^{M,D}, m_{TP}^{S_{*\to p},D} - m_{TP}^{S_{*\to n},D})$
- Clase Negative: $m_{FP}^{S_{*\to n},D} + \text{Min}(m_{FP}^{M,D}, m_{TN}^{S_{*\to n},D} - m_{TN}^{S_{*\to p},D})$

Resultando en el peor caso del sistema:

- $m_{FN}^{S_{*\to p},D} + \text{Min}(m_{FN}^{M,D}, m_{TP}^{S_{*\to p},D} - m_{TP}^{S_{*\to n},D}) +$
  $m_{FP}^{S_{*\to n},D} + \text{Min}(m_{FP}^{M,D}, m_{TN}^{S_{*\to n},D} - m_{TN}^{S_{*\to p},D})$

Problema 2

El segundo problema a resolver entonces va a ser analizar la función de costo para el modelo cuando los datasets sean distintos:

- La función de costo de peor caso del modelo en el sistema es la analizada en el problema anterior
- Los datasets para evaluar el modelo y el sistema son distintos
- Los datos de ambos datasets provienen de la misma distribución

Datasets distintos

Veamos como interpretar la función anterior para poderla usar en caso que ambos datasets sean distintos:

- Las confusiones del modelo se ejecutan con el dataset del modelo sin problemas
- Las confusiones del sistema intervenido se ejecutan con un dataset distinto
- Se van a estimar las confusiones que hubiera tenido el sistema intervenido en el dataset del modelo

Datasets distintos

Teniendo la confusión de un sistema intervenido para un dataset, se estima la confusión en otro de forma simple:

- Primero se divide cada confusión por la cantidad de instancias de su clase para saber el accuracy
- Luego se multiplica cada accuracy por la cantidad de instancias de su clase correspondiente en el dataset del modelo

Datasets distintos

Por ley de los grandes números, a medida que aumenta la cantidad de instancias de cada dataset se puede saber lo siguiente:

- Cada accuracy calculado para el dataset del sistema va a tender al accuracy de la población
- Si se ejecutaba el sistema intervenido con los datos del modelo su accuracy hubiera tenidodo a la media de la población
- Como en ambos casos el accuracy tiende a lo mismo, la estimación de la confusión no va a ser muy lejana a la real

Experimentación

Para comprobar que estimar la confusión lleva a una función de costo que es de utilidad en el problema, se van a realizar dos experimentos.

Para ambos experimentos se ejecuta primero el sistema intervenido para un dataset del sistema, se estima su confusión en el dataset del modelo y se ejecuta el modelo.

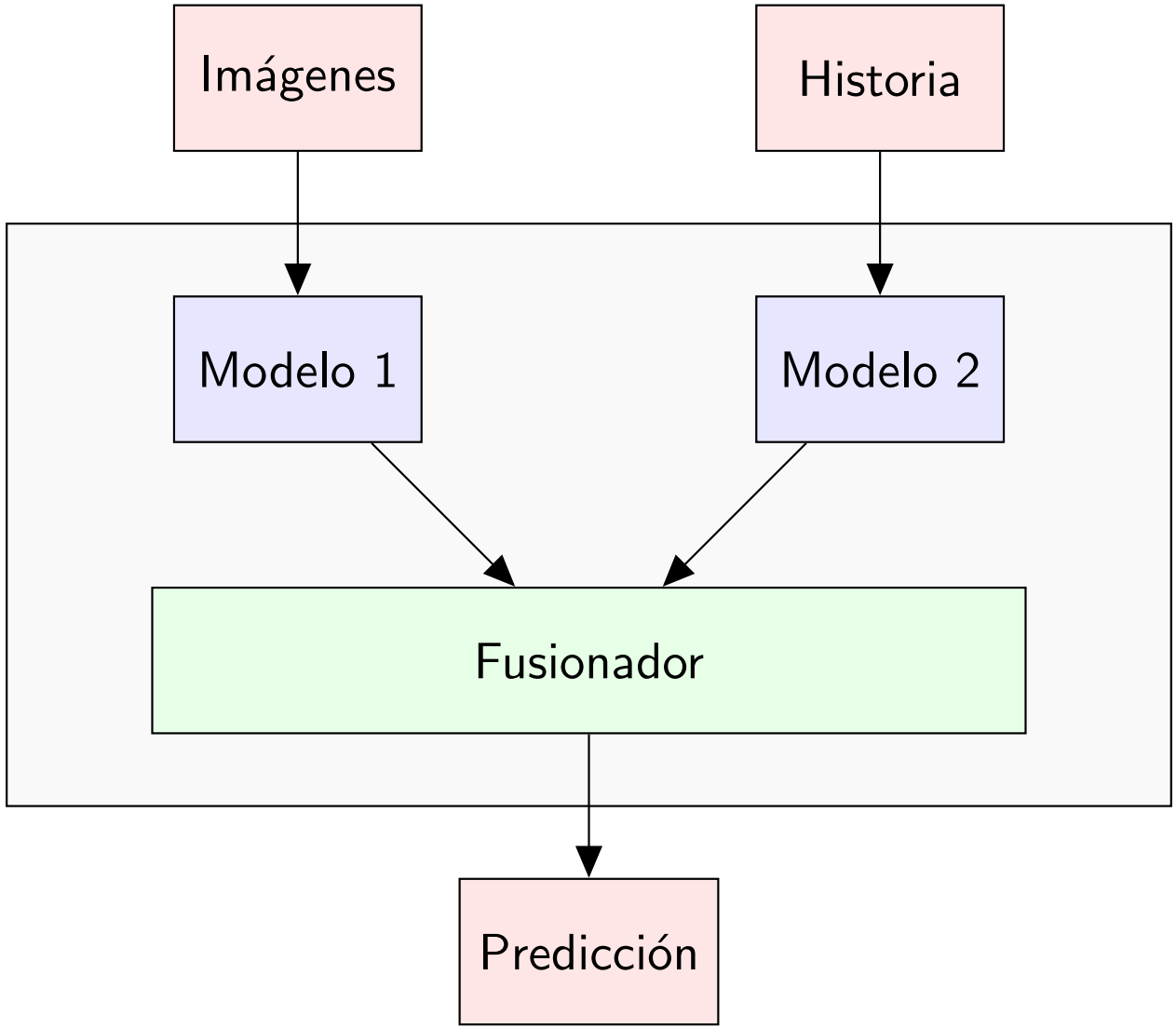
A su vez se ejecuta realmente el sistema con el dataset del modelo para comparar los resultados reales con los que provee la función de peor caso.

Experimentación

El sistema utilizado tiene 2 modelos, que clasifican como clase Positiva o Negativa. El fusionador puede ser un OR/AND lógico.

Los resultados de ambos modelos fueron simulados con variables aleatorias Bernoulli para cada instancia y cada clase. Las variables de una misma clase pueden no ser independientes entre sí.

Experimentación



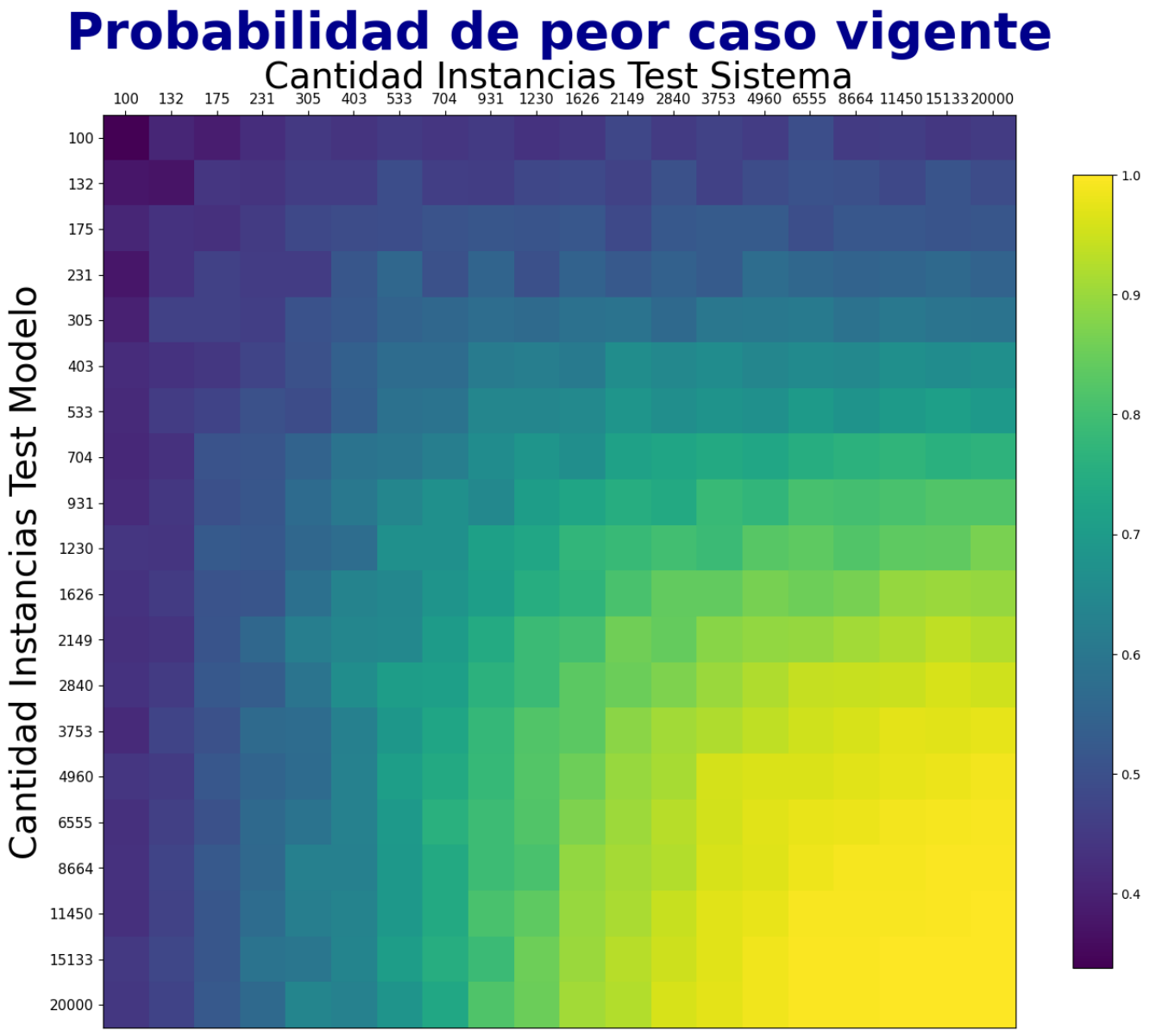
Experimento 1

- Los accuracies de ambos modelos 0.5 ambas clases
- Fusionador de tipo OR
- Correlación clase Positive de 0.95
- Correlación clase Negative de -0.95

Se asignan distintos tamaños en cada dataset y para cada combinación se ejecuta 1000 veces y se reporta:

- $P(PeorCaso \geq EvalReal)$

Experimento 1



Experimento 1

La probabilidad se acerca rápidamente a 1 a medida que ambas cantidades de instancias aumentan.

Para el caso de un sistema con dos modelos, a tamaños no tan grandes, el peor caso usando datasets distintos provee casi la misma información que con un solo dataset.

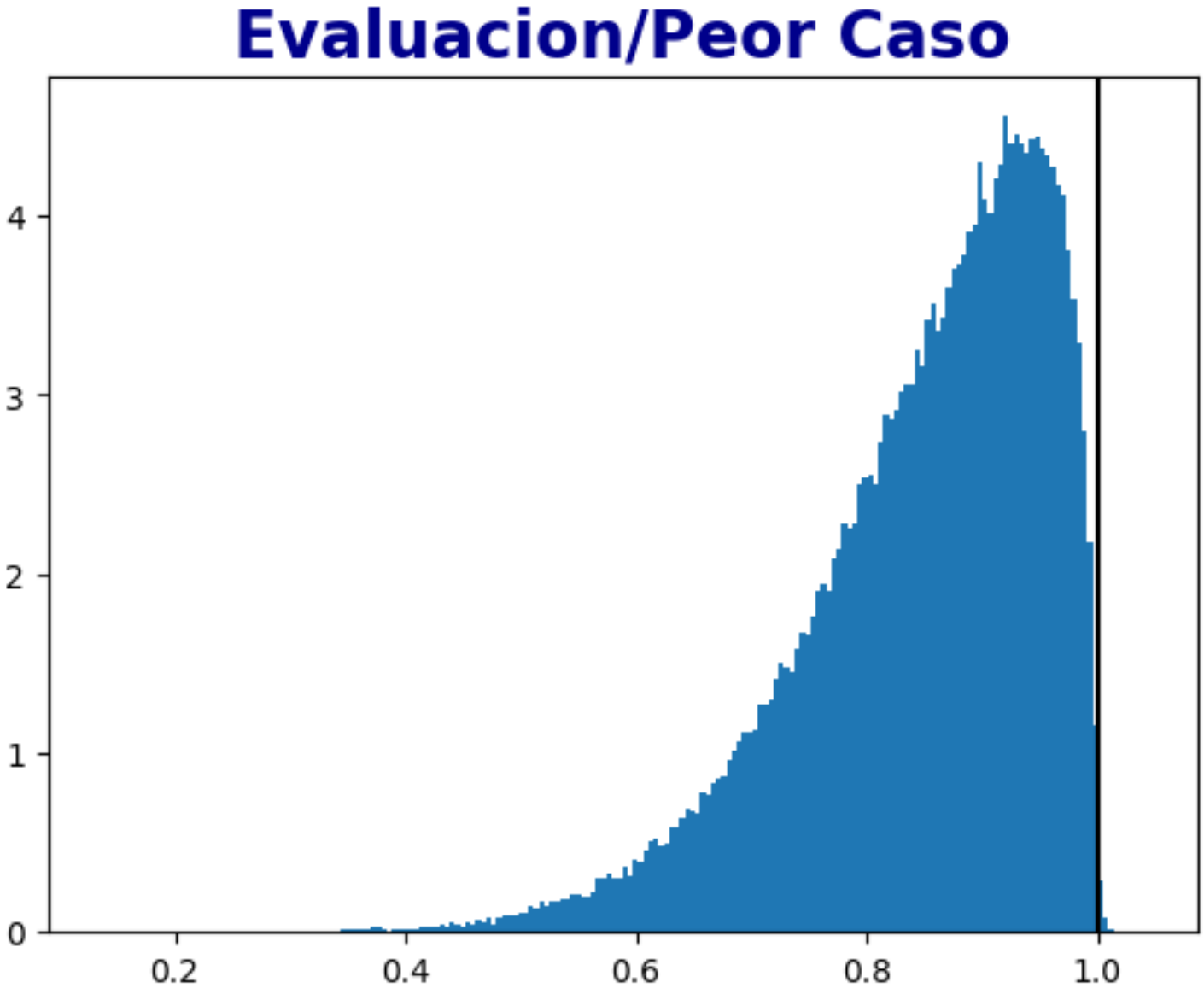
Experimento 2

- Accuracies de ambos modelos aleatorias
- Fusionador aleatorio entre OR y AND
- Correlacións aleatorias válidas para los accuracies

Los tamaños de ambos datasets tienen 10000 instancias por clase, se ejecuta 100000 veces y se reporta:

- $\frac{EvalReal}{PeorCaso}$

Experimento 2



Experimento 2

El cociente entre ambos valores no suele pasar 1, siendo esto el resultado buscado. Incluso en los raros casos en los que si lo supera, no lo hace por un márgen muy grande.

Preguntas?