Análisis de costo de clasificadores

2025 Tesis de Licenciatura

Gianfranco Lopetrone
gian2000franco@gmail.com

Director
Victor Braberman
victor.braberman@gmail.com

Co-Director
Sebastian Uchitel
sebastian.uchitel@gmail.com

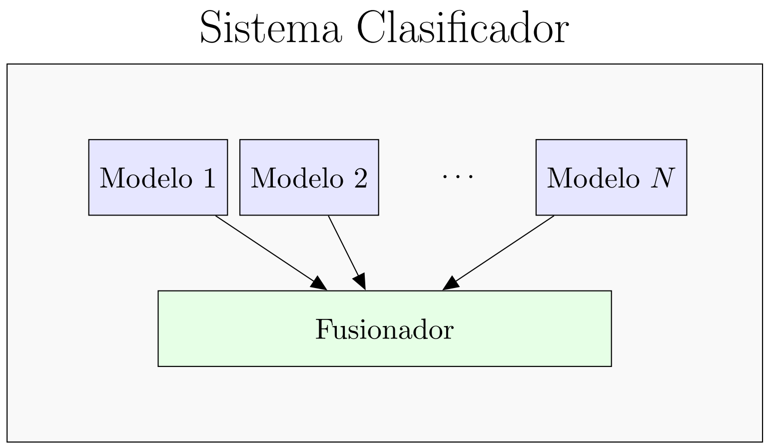
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón Cero + Infinito)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Conmutador: (+54 11) 5285-9721 / 5285-7400
https://dc.uba.ar

1 Introducción

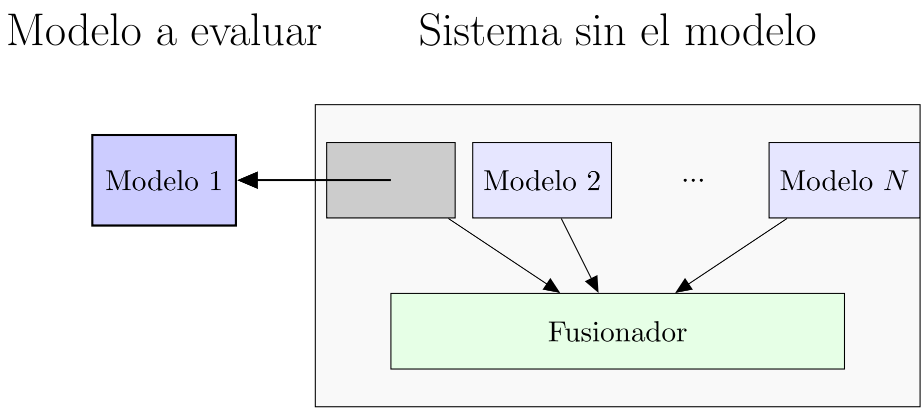
2 Problema a trabajar

El problema a trabajar va a ser el de un sistema clasificador, que recibe datos de distintos tipos y luego devuelve una clase asociada a esos datos. Por ahora las clases asociadas a los datos pueden dos, Positiva y Negativa, siendo un problema de clasificación binaria. Los datos que recibe el sistema son luego de forma parcial o total observados por distintos modelos clasificadores que dan cada uno su resultado individual. Luego esos resultados son puestos en común por un fusionador, que devuelve una única clase para cada dato asociado.

Los modelos corren en paralelo, y de forma independiente donde cada uno mira parte del dato de entrada y devuelve su resultado, sin depender de otros modelos.



El objetivo es encontrar alguna función con la cual evaluar uno de los modelos del sistema, sin la necesidad de correr el sistema completo. Al evaluar el modelo con esa función, se debe obtener información relevante del sistema solo al ejecutarla en el modelo. Esto no significa que para definir la función no se pueda correr el sistema completo. Por lo tanto el planteo es encontrar una función que de información relevante, construida en base al sistema en el que el modelo se va a ejecutar, pero con un hueco en el lugar del modelo. Esta función debe tener al modelo como parámetro. Luego se utiliza independiente al sistema en cualquier posible modelo que vaya a ocupar ese lugar.

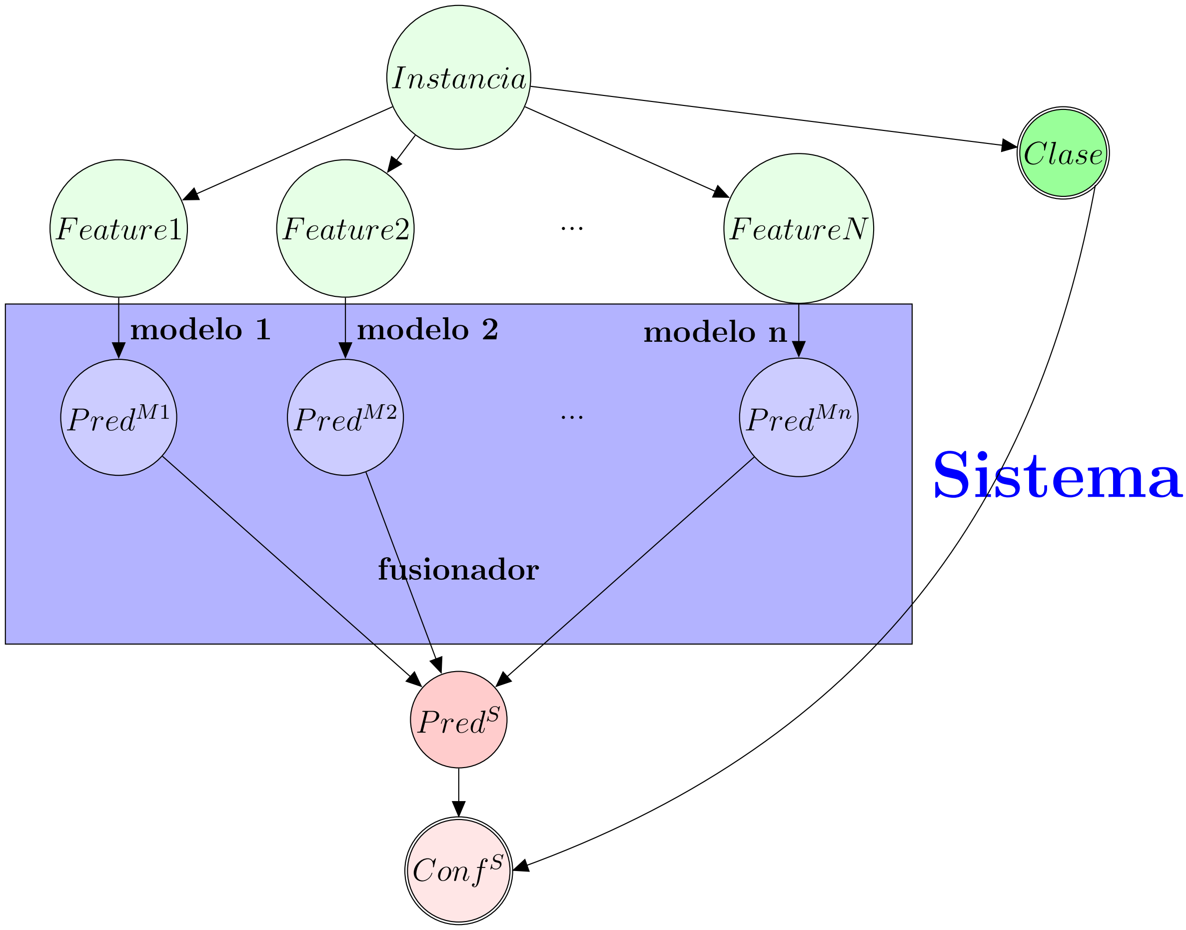


Para representar el funcionamiento del sistema en cierta población de datos, voy a usar Redes Bayesianas Causales.

2.1 Redes bayesianas causales

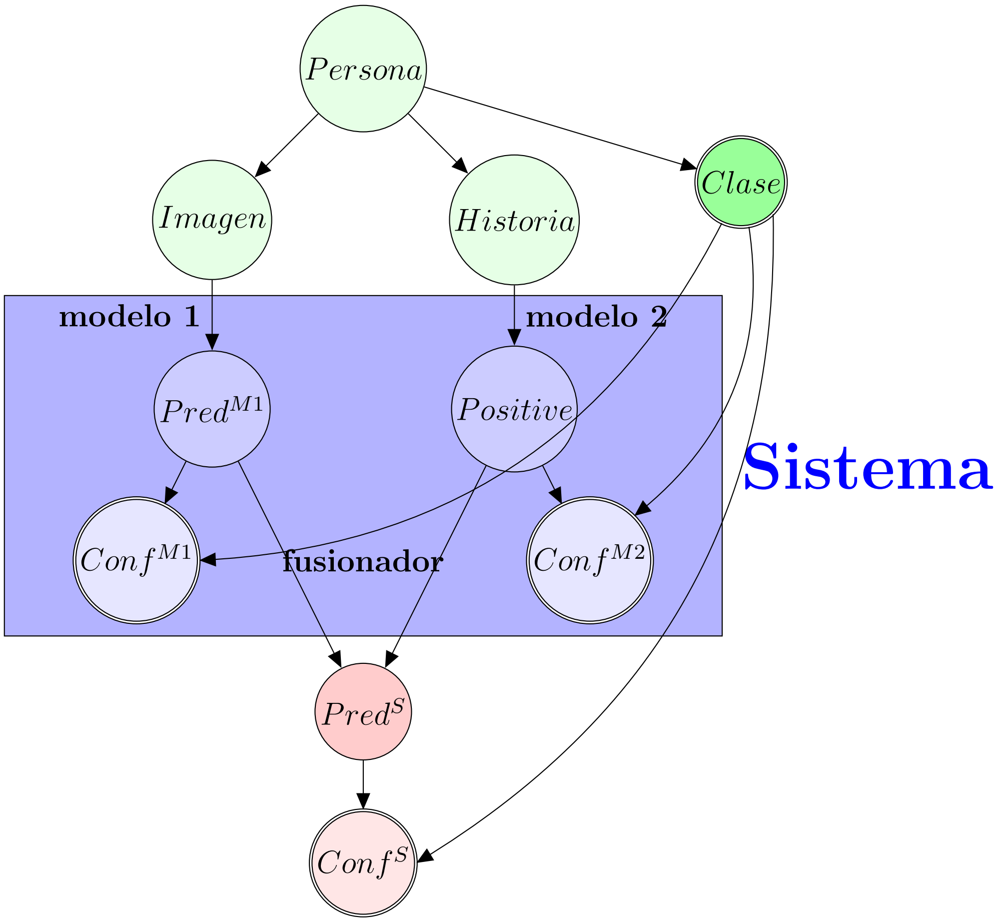
Una Red Bayesiana es un Grafo Acíclico Dirigido, donde se representan distribuciones de probabilidad conjuntas. Se separan variables aleatorias en nodos y se vinculan a otras con aristas, representando probabilidades condicionales, asi definiendo la distribución completa de forma eficiente. Una Red Bayesiana Causal es una Red Bayesiana cuyas aristas agregan un significado de causalidad de nodos padres a nodos hijos sobre el significado probabilistico de las Redes Bayesianas[1]. Los nodos con doble círculo representan que son deterministicos dados sus padres.

Para el problema a tratar, voy a modelar con un nodo raiz la población de datos, del cual depende la Clase a la que pertenecen y las distinas features que se observan sobre ese dato. Luego esas features son analizadas por el sistema y por último se consigue una predicción y su confusión asociada dependiendo de la clase a la que pertenece ese dato.



2.2 Ejemplo a analizar

Empiezo usando un sistema en particular para analizar y intentar generalizar a otros casos. Tengo un sistema con dos modelos en paralelo, cada uno con datos de entrada distintos. Uno por ejemplo podría recibir imágenes médicas y el otro la historia clínica de un paciente, cada uno clasificando el estado del paciente (Positivo o Negativo) de forma independiente. Luego sus resultados son pasados a un fusionador que se encarga de unificarlos y devolver un único resultado para todo el sistema.

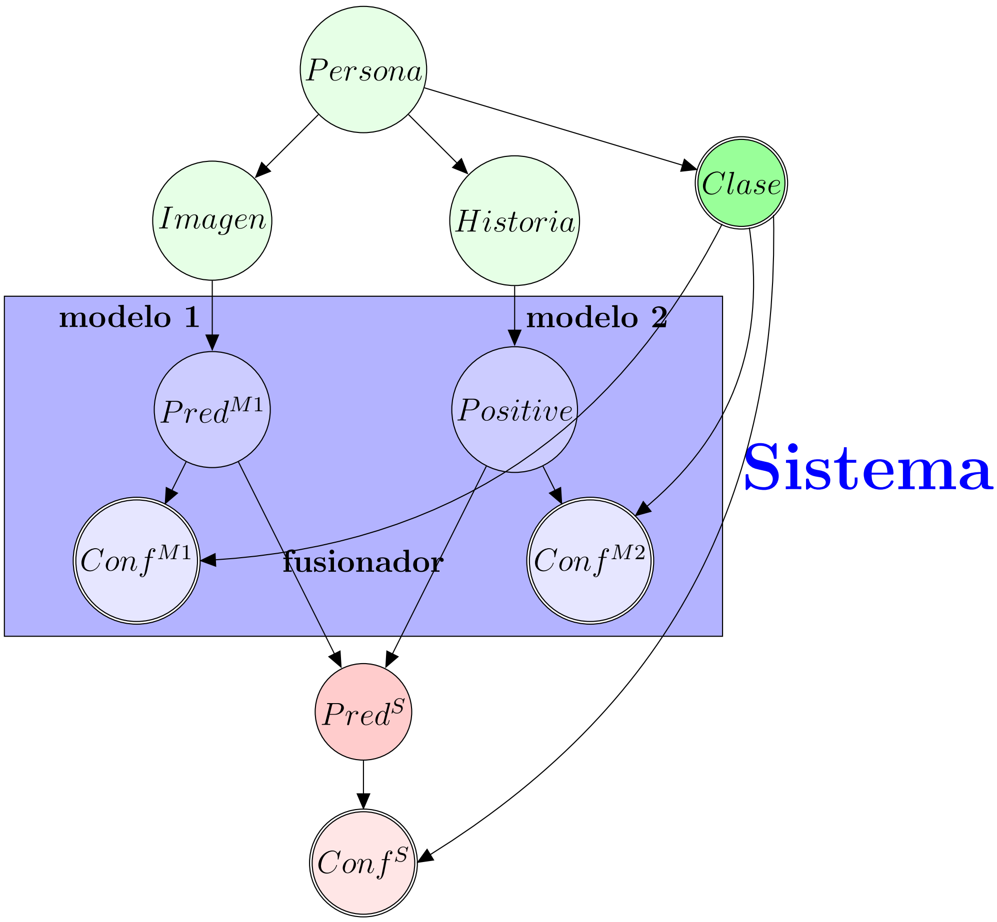


2.3 Intervención

Una operación importante en las Redes Bayesianas Causales es la operación de la intervención. Las consultas que se suelen hacer a las Redes Bayesianas son probabilidades condicionales, de la pinta $P(X = x|Y = y)$, donde el dominio se restringe solo a los casos donde el nodo $Y = y$ para luego mirar la probabilidad de X.

La intervención por otro lado, también es una consulta probabilistica sobre la red, pero donde se le asigna a cierto nodo un valor con probabilidad 1. La operación de la intervención se ve reflejado en una consulta con el operador **do**. Un ejemplo se vería de la forma $P(X = x|do(Y = y))$. La diferencia con la consulta anterior se da en que la primera acota el unvierso de datos solo a los que $Y = y$, pero la segunda con intervención no. Esta consulta mira todos los valores de X, para todos los datos, pero asumiendo que $Y = y$ en todos ellos e ignorando el verdadero valor de Y.

En una Red Causal, la intervención se puede ver reflejada en una red simplemente sacando las aristas de los padres del nodo intervenido hacia él. Si quisiera realizar intervención en la red del sistema de ejemplo anterior, donde la predicción del modelo 2 de histórias clínicas fuera siempre que la persona pertenece a clase Positiva ($Pred^{M2} = Positive$), el modelo gráfico resultante se vería de la siguiente manera:



La operación de invervención es muy importante para analizar como se comporta el sistema sin necesidad de evaluarlo con un modelo. Se pueden asumir distintos comportamientos sencillos del modelo y analizar el rendimiento del sistema para esos distintos modelos por medio de la intervención. Luego teniendo la evaluación real de un posible modelo, mediante alguna función de costo, asemejar el comportamiento del sistema con el modelo real a los distintos comportamientos observados del sistema con las intervenciones en el modelo.

3 Caso Promedio

El modelo que buscamos evaluar es el modelo 2 que recibe histórias clínicas. Buscamos darle un costo a los errores que pueda cometer el modelo 2, dandole a cada clase un costo particular según cuanto afecta al sistema en su totalidad.

Para definir esta idea más formalmente usamos la matriz de confusión y su matriz de costo asociada.

La matriz de confusión es una matriz que tiene como columnas las clases reales a las que pertenecen los datos que evaluamos, y como filas a las clases que nuestro clasificador predijo. Luego en cada celda de la matriz se encuentra un número entero que representa la cantidad de ocurrencias de esa combinación en particular luego de ejecutar al clasificador sobre nuestros datos de evaluación.

La matriz de costo asociada a la matriz de confusión es una forma de darle un costo a cada tipo de error de la matriz de confusión. Es una matriz con las mismas dimensiónes, pero en lugar de contar cantidad de ocurrencias, le da un costo particular a cada tipo de error, siendo que la predicción difiera de la clase real. A los datos para los cuales nuestra predicción acertó se le suele dar un costo de 0.

	Predicción Positive	Predicción Negative
Clase Positive	$Conf_{TP}$	$Conf_{FN}$
Clase Negative	$Conf_{FP}$	$Conf_{TN}$

Matriz de confusión

	Predicción Positive	Predicción Negative
Clase Positive	$Costo_{TP}$	$Costo_{FN}$
Clase Negative	$Costo_{FP}$	$Costo_{TN}$

Matriz de costo

Esa matriz nos permite a la hora de evaluar al clasificador, dar un costo realista a cada error con el objetivo de que refleje los costos reales de cometerlos en la realidad. De esta forma elegir un clasificador sobre otro cuando la suma para todas las instancias de los costos asociados a la predicción que hagamos sea mínima, consiguiendo un clasificador que se adapte mejor a nuestra situación de uso.

La forma de evaluar el costo de un clasificador es correrlo para muchas instancias y contar la cantidad de veces que ocurre cada tipo de resultado en la matriz de confusión. Luego multiplicarla elemento a elemento con la matriz de costo asociada y sumar todos los resultados. A mayor costo, peor es el funcionamiento del clasificador para esas instancias.

Para cualquier sistema clasificador, sucede que la matriz de costo está definida para la clasificación final del sistema, pero nosotros queremos evaluar que tan bien funciona un modelo integrante de ese sistema. En particular el modelo 2 de historias clínicas en este caso.

Por lo tanto debemos derivar una matriz de costo asociada al modelo en base la matriz de costo del sistema y algunas otras características según el objetivo que buscamos para nuestra matriz de costo.

Podemos querer una matriz de costo para distintos objetivos y vamos a analizar algunos de ellos:

3.1 Estimación del costo del sistema para los resultados del modelo

Una primera idea es para cada resultado del modelo 2, sumar la probabilidad de que ese resultado cause un fallo en el sistema y pesarlo por el costo de ese fallo en particular. Por ejemplo si nuestro modelo de histórias clínicas clasifica a una persona que se encontraba sana como alguien que tiene un problema de salud, sumar la probabilidad de que esa mala clasificación se translade al sistema y resulte en que el sistema también se equivoque en la clasificación. Luego lo pesamos por el costo del error del sistema para un False Negative.

- Probabilidad de error del sistema cuando el modelo 2 falla:

  Clase Positive: $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = N))$

  Clase Negative: $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = P))$

- Probabilidad de error del sistema cuando el modelo 2 acierta:

  Clase Positive: $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P))$

  Clase Negative: $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = N))$

Ahora sabiendo la probabilidad de que cada resulado del modelo 2 cause un error en el sistema, pesando esas probabilidades por los costos reales de los fallos podemos calcular los costos asociados al modelo 2.

- Costo de una instancia para el modelo 2:

- $Costo^{M2}_{TP} = P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P))$ x $Costo^S_{FN}$

  $Costo^{M2}_{TN} = P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = N))$ x $Costo^S_{FN}$

  $Costo^{M2}_{FN} = P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = N))$ x $Costo^S_{FP}$

  $Costo^{M2}_{FP} = P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = P))$ x $Costo^S_{FP}$

  Tenemos una función de costo asociada a cada resultado posible del modelo 2.

  Para evaluar el modelo 2, lo corremos para un conjunto de instancias conocidas y multiplicamos cada una de las confusiónes obtenidas por el costo asociado a ellas que acabamos de estimar. Luego sumamos esos valores.

- Evaluación para el modelo 2:

  $Eval^{M2} = Costo^{M2}_{TP}$ x $Conf^{M2}_{TP} + Costo^{M2}_{TN}$ x $Conf^{M2}_{TN} + Costo^{M2}_{FP}$ x $Conf^{M2}_{FP}$ + $Costo^{M2}_{FN}$ x $Conf^{M2}_{FN}$

  Donde $Conf^{M2}_{FP}$ se refiere a la cantidad de instancias que clasificamos de forma Positive y eran de clase Negative al correr el modelo 2 individualmente.

  Ahora tenemos una forma de evaluar el costo de cada tipo de error para el modelo 2 de forma individual, estimando el costo al sistema sin necesidad de ejecutarlo. Más adelante veremos algunos casos de ejemplo.

3.2 Representación del costo de los fallos del modelo en el sistema

Como vimos anteriormente, la matriz de costo asociada a un problema de clasificación suele tener un costo de 0 para los aciertos (elementos de la diagonal). Nuestra función de costo anterior no cumple esa propiedad. Con esta función buscamos que se cumpla eso, sin perder la influencia que tienen los resultados de nuestro modelo a nivel sistema.

A su vez solo penalizamos los errores del modelo que efectivamente causen una perdida de rendimiento del sistema, y reducimos el costo de los errores que no tienen impacto en el resultado para mantener la relación entre los distintos resultados de una clase.

Esto lo logramos restando a la probabilidad de fallo del sistema cuando el modelo falla, la probabilidad de fallo del sistema cuando el modelo acierta que llamamos Tasa de transición de error. Este valor resultante nos da la probabilidad que fallar introduzca un nuevo error que acertar hubiera podido evitar.

- Tasa de transición de error:

  Clase Positive: $P_{\mathcal{G}}(ConfF = FN|Clase = P, \mathrm{do}(Pred2 = N)) - P_{\mathcal{G}}(ConfF = FN|Clase = P, \mathrm{do}(Pred2 = P))$

  Clase Negative: $P_{\mathcal{G}}(ConfF = FP|Clase = N, \mathrm{do}(Pred2 = P)) - P_{\mathcal{G}}(ConfF = FP|Clase = N, \mathrm{do}(Pred2 = N))$

  Estas probabilidades solo tiene sentido definirlas para los fallos del modelo. Para los aciertos del modelo el primer y segundo término son idénticos y se cancelan mutuamente consiguiendo una probabilidad de 0 como buscabamos.

  Para conseguir la función de costo buscada solo queda multiplicar por el costo real del error en el sistema como habiamos hecho antes.

- Costo de una instancia para el modelo 2:

  $Costo^{M2}_{TP} = 0 =$

  $(P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P))$ - $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P)))$ x $Costo^S_{FN}$

  $Costo^{M2}_{TN} = 0 =$

$(P_{\mathcal{G}}(Conf^S = FP|Clase = N, \text{do}(Pred^{M2} = N))$ - $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \text{do}(Pred^{M2} = N)))$ x $Costo^S_{FN}$

$Costo^{M2}_{FN} =$

$(P_{\mathcal{G}}(Conf^S = FN|Clase = P, \text{do}(Pred^{M2} = N))$ - $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \text{do}(Pred^{M2} = P)))$ x $Costo^S_{FP}$

$Costo^{M2}_{FP} =$

$(P_{\mathcal{G}}(Conf^S = FP|Clase = N, \text{do}(Pred^{M2} = P))$ - $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \text{do}(Pred^{M2} = N)))$ x $Costo^S_{FP}$

Aca podemos observar como en el caso de True Positive y True Negative se resta y suma el mismo término. Esto nos da una probabilidad de 0 que luego se transifiere a que el costo asociado también sea 0.

Evaluar el modelo 2 en base a esta nueva función de costo es igual que antes. Luego de correr nuestro modelo para un conjunto de instancias, multiplicamos la cantidad de apariciones de cada tipo de resultado por el costo asociado y los sumamos.

- Evaluación para el modelo 2:

  $Eval^{M2} = Costo^{M2}_{TP}$ x $Conf^{M2}_{TP} + Costo^{M2}_{TN}$ x $Conf^{M2}_{TN} + Costo^{M2}_{FP}$ x $Conf^{M2}_{FP} + Costo^{M2}_{FN}$ x $Conf^{M2}_{FN}$

3.3 Costo de peor caso del sistema

En los casos anteriores usamos la función de costo para que frente a un modelo 2, al evaluarlo con alguna de esas funciónes podamos ver cuanto se ve afectada la evaluación final del sistema. Esto no es del todo acertado para el caso general ya que la evaluación solo va a ser exacta cuando los modelos que componen el sistema sean independientes uno de otro frente a sus instancias. En cuanto se rompa esa independencia la evaluación del modelo 2 comienza a diferir del resultado real del sistema al correrlo con ese modelo.

Como ejemplo veamos el caso donde tenemos un fusionador AND lógico que predice positivo si al menos una de los 2 modelos predicen positivo. A su vez tenemos un modelo 1 con un 50% de accuracy para la clase positiva y un costo de 1 el error de False Negative asociado. Al calcular el costo con nuestra primera función obtenemos que para la clase positiva el costo de un True Positive es de 1/2 y de un False Negative es de 1. Al correrlo para 100 instancias con un modelo 2 de 50% de accuracy observamos que el costo total del sistema es de 100 para esa clase en vez de 75, que nuestro modelo 2 estima en base a los resultados. Al observar mas cuidadosamente como funciona el sistema vemos que los errores de los modelos suelen ser en lugares distintos por lo que aunque ambos tengan 50% de accuracy para la clase positiva, como fallan siempre en instancias distintas nuestro fusionador nunca acierta correctamente para esa clase.

En estas situaciones definir previamente una función de costo para cualquier modelo 2 y esperar que funcione correctamente en el sistema no es posible, por lo que buscamos analizar como se comporta el peor caso. El objetivo de esta tercera función es determinar cual es el peor resultado posible que nuestro sistema pueda llegar a tener dada una matriz de confusión del modelo que buscamos evaluar.

- (1) Errores máximos del sistema cuando modelo = TP:
  Min($Conf^{M2}_{TP}$, $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P))$ x $CantPositive$)
- (2) Errores máximos del sistema cuando modelo = TN:
  Min($Conf^{M2}_{TN}$, $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = N))$ x $CantNegative$)

  Estos dos valores dicen la cantidad máxima de errores posibles del sistema dado que nuestro modelo acierte, uno para cada clase. Dado que la probabilidad dentro del mínimo es la cantidad maxima de fallos que puede haber si el modelo acierta siempre, este peor caso considera que cada acierto nuestro justo cae en un caso en los que el sistema falla igual incluso con el acierto del modelo.
- (3) Errores máximos del sistema cuando modelo = FN:
  Min($Conf^{M2}_{FN}$,
  $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = N))$ x $CantPositive$ -
  Min($Conf^{M2}_{TP}$, $P_{\mathcal{G}}(Conf^S = FN|Clase = P, \mathrm{do}(Pred^{M2} = P))$ x $CantPositive$))
- (4) Errores máximos del sistema cuando modelo = FP:
  Min($Conf^{M2}_{FP}$,
  $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = P))$ x $CantNegative$ -
  Min($Conf^{M2}_{TN}$, $P_{\mathcal{G}}(Conf^S = FP|Clase = N, \mathrm{do}(Pred^{M2} = N))$ x $CantNegative$))

  Estos otros dos valores nos dicen lo mismo pero para los casos donde el modelo falla. La cantidad máxima de fallos posibles en este caso es la primera probabilidad que se encuentra dentro del mímimo externo. Pero a esa cantidad máxima de fallos debemos quitarle la cantidad que ya le asignamos a los aciertos anteriormente.

  Esto se debe a que ambas probabilidades son calculadas sobre el total de instancias, y asumimos que las instancias para las cuales acertó el modelo pero en cambio el sistema falló, si el modelo hubiera fallado entonces el sistema seguiría fallando. A su vez podemos asumir que si el modelo falló y el sistema acertó en la clasificación, si el modelo hubiera acertado entonces se mantiene el acierto del sistema para cualquier instancia.

  En base a esto es que a las instancias donde tanto el sistema como el modelo fallan le restamos los fallos del sistema donde el modelo acertó porque asumimos que las instancias esas ya fueron contempladas.

  Luego nos queda construir la función de costo en base a estos valores. Como cada uno representa el peor caso del sistema para distintos resultados nuestro modelo, sumamos los errores de la misma clase y los multiplicamos por su costo asociado. el resto de las instancias de cada clase las asignamos a predicciones correctas del sistema.

	Predicción Positive	Predicción Negative
Clase Positive	($Conf_{TP}^{M2}$-(1) + $Conf_{FN}^{M2}$-(3)) x 0	((1) + (3)) x $Costo_{FN}^{S}$
Clase Negative	((2) + (4)) x $Costo_{FP}^{S}$	($Conf_{TP}^{M2}$-(2) + $Conf_{FN}^{M2}$-(4)) x 0

4 Experimentación

Ejemplos para distintos casos:
FUNCIÓN DE COSTO NUMERO 1

1. Para un modelo de imágenes con un 80% de accuracy para ambas clases usando un fusionador AND lógico y el costo para el sistema de un False Positive es 1 y False Negative 5:

   Probabilidad de error del sistema cuando el modelo 2 falla:

   Clase Positive = 100%, Clase Negative = 20%

   Probabilidad de error del sistema cuando el modelo 2 acierta:

   Clase Positive = 20%, Clase Negative = 0%

   Costo de una instancia para el modelo 2:

   $Costo_{TP}^{M2}$ = 20% x 5 = 1

   $Costo_{TN}^{M2}$ = 0% x 1 = 0

   $Costo_{FP}^{M2}$ = 20% x 1 = 0.2

   $Costo_{FN}^{M2}$ = 100% x 5 = 5

   Luego en caso de correr al sistema con 100 instancias de la clase Positiva y 200 instancias de la clase Negativa, y un modelo 2 de 50% accuracy, asumiendo que los errores cometidos por ambos modelos fueran independientes, esperaríamos obtener los siguientes resultados:

   $P_{\mathcal{G}}(Conf^S = FN|Clase = P)$ = 100% - (80% x 50%) = 60%

   $P_{\mathcal{G}}(Conf^S = FP|Clase = N)$ = (100% - 80%) x (100% - 50%) = 10%

   $Costo_{Total}^{S}$ = 60% x 100 x 5 + 10% x 200 x 1 = 300 + 20 = 320

   $Costo_{Total}^{M1}$ = 50% x 100 x 1 + 50% x 200 x 0 + 50% x 200 x 0.2 + 50% x 100 x 5 = 300 + 20 = 320

   Podemos ver como los costos son los mismos para tanto el modelo como el sistema en este caso. También esta igualdad se mantiene dentro de cada clase.

2. Para un modelo de imágenes con un 25% de accuracy para la clase Positiva y 50% para la Negativa usando un fusionador OR lógico y el costo para el sistema de un False Positive es 1 y False Negative 5:

   Probabilidad de error del sistema cuando el modelo 2 falla:

Clase Positiva = 75%, Clase Negativa = 100%

Probabilida de error del sistema cuando el modelo 2 acierta:

Clase Positiva = 0%, Clase Negativa = 50%

Costo de una instancia para el modelo 2:

$Costo_{TP}^{M2}$ = 0% x 5 = 0

$Costo_{TN}^{M2}$ = 50% x 1 = 0.5

$Costo_{FP}^{M2}$ = 100% x 1 = 1

$Costo_{FN}^{M2}$ = 75% x 5 = 3.75

Si corremos el sistema para 100 instancias de la clase Positiva y 200 de Negativa, con un modelo 2 con 40% accuracy para clase Positiva y 50% Negativa asumiendo que los errores de ambos modelos son independientes esperaríamos obtener los siguientes resultados:

$P_{\mathcal{G}}(Conf^S = FN|Clase = P)$ = (100% - 25%) x (100% - 40%) = 45%

$P_{\mathcal{G}}(Conf^S = FP|Clase = N)$ = 100% - (100% - 50%) x (100% - 50%) = 75%

$Costo_{Total}^{S}$ = 45% x 100 x 5 + 75% x 200 x 1 = 225 + 150 = 375

$Costo_{Total}^{M1}$ = 40% x 100 x 0 + 50% x 200 x 0.5 + 50% x 200 x 1 + 60% x 100 x 3.75 = 225 + 150 = 375

En este otro caso sucede lo mismo que en el anterior. Mientras los errores sean independientes, el costo del modelo 2 y del sistema son idénticos para cada clase.

FUNCIÓN DE COSTO NUMERO 2

1. Para un modelo de imágenes con un 80% de accuracy para ambas clases usando un fusionador AND lógico y el costo para el sistema de un False Positive es 1 y False Negative 5:

   Probabilidad de error del sistema cuando el modelo 2 falla:

   Clase Positive = 100%, Clase Negative = 20%

   Probabilidad de error del sistema cuando el modelo 2 acierta:

   Clase Positive = 20%, Clase Negative = 0%

   Costo de una instancia para el modelo 2:

   $Costo_{TP}^{M2}$ = (20% - 20%) x 5 = 0

   $Costo_{TN}^{M2}$ = (0% - 0%) x 1 = 0

   $Costo_{FP}^{M2}$ = (20% - 0%) x 1 = 0.2

   $Costo_{FN}^{M2}$ = (100% - 20%) x 5 = 4

   Luego en caso de correr al sistema con 100 instancias de la clase Positiva y 200 instancias de la clase Negativa, y un modelo 2 de 50% accuracy, asumiendo que los errores cometidos por ambos modelos fueran independientes, esperaríamos obtener los siguientes resultados:

$P_{\mathcal{G}}(Conf^S = FN|Clase = P)$ = 100% - (80% x 50%) = 60%

$P_{\mathcal{G}}(Conf^S = FP|Clase = N)$ = (100% - 80%) x (100% - 50%) = 10%

$Costo^S_{Total}$ = 60% x 100 x 5 + 10% x 200 x 1 = 300 + 20 = 320

$Costo^{M1}_{Total}$ = 50% x 100 x 0 + 50% x 200 x 0 + 50% x 200 x 0.2 + 50% x 100 x 4 = 200 + 20 = 220

2. Para un modelo de imágenes con un 25% de accuracy para la clase Positiva y 50% para la Negativa usando un fusionador or lógico y el costo para el sistema de un False Positive es 1 y False Negative 5:

   Probabilidad de error del sistema cuando el modelo 2 falla:

   Clase Positiva = 75%, Clase Negativa = 100%

   Probabilida de error del sistema cuando el modelo 2 acierta:

   Clase Positiva = 0%, Clase Negativa = 50%

   Costo de una instancia para el modelo 2:

   $Costo^{M2}_{TP}$ = (0% - 0%) x 5 = 0

   $Costo^{M2}_{TN}$ = (50% - 50%) x 1 = 0

   $Costo^{M2}_{FP}$ = (100% - 50%) x 1 = 0.5

   $Costo^{M2}_{FN}$ = (75% - 0%) x 5 = 3.75

   Si corremos el sistema para 100 instancias de la clase Positiva y 200 de Negativa, con un modelo 2 con 40% accuracy para clase Positiva y 50% Negativa asumiendo que los errores de ambos modelos son independientes esperaríamos obtener los siguientes resultados:

   $P_{\mathcal{G}}(Conf^S = FN|Clase = P)$ = (100% - 25%) x (100% - 40%) = 45%

   $P_{\mathcal{G}}(Conf^S = FP|Clase = N)$ = 100% - (100% - 50%) x (100% - 50%) = 75%

   $Costo^S_{Total}$ = 45% x 100 x 5 + 75% x 200 x 1 = 225 + 150 = 375

   $Costo^{M1}_{Total}$ = 40% x 100 x 0 + 50% x 200 x 0 + 50% x 200 x 0.5 + 60% x 100 x 3.75 = 225 + 50 = 275

Para la función de costo número 2, el fusionador con un AND lógico da costos iguales al sistema para la clase Negative y distintos para Positive. El caso del fusionador con un OR, la clase Positive es la que da igual y Negative distinta.

Se puede ver que para cada clase en particular, si el sistema no introduce ningún error en los casos donde el modelo 2 clasifica correctamente, tanto el costo del sistema como el del modelo 2 serán iguales. Si el sistema ya introducía errores incluso acertando en el modelo 2, el costo de la clase se verá reducido proporcionalmente.

References

[1] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press, 2009. URL: https://books.google.co.in/books?id=7dzpHCHzNQ4C.