



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

Diseño y análisis de algoritmos para encaminamiento en redes de comunicación con fallos

Tesis presentada a la Facultad de Ciencias Físicas y Matemáticas de la
Universidad de Concepción para optar al título profesional de Ingeniero
Civil Matemático

Por
Jorge Eduardo Ávila Cartes

Profesores guía: Dr. Christopher Thraves y Dr. Sebastián Niklitschek

Enero, 2018
Concepción, Chile



©Jorge Eduardo Ávila Cartes
Se autoriza la reproducción total o parcial, con fines
académicos, por cualquier medio o procedimiento,
incluyendo la cita bibliográfica del documento.



A Yasmin y Floricio.

Agradecimientos

La mayoría de las historias empiezan con un *érase una vez*. Hoy quiero partir hablando desde el presente. No muchos imaginan la satisfacción que se siente lograr dar un salto como este, lograr ser parte, para toda la vida, de este pequeño pero maravilloso grupo de ICM's de la UdeC. No siempre fue mi sueño serlo, de hecho lo descubrí pocos meses antes de entrar a la universidad, pero debo decir que volvería a elegir el mismo camino si me enfrentase en otra vida a esta decisión.

Hoy en día una de las primeras preguntas que se le hace a una persona es ¿qué estudió?, y créanme que desde donde yo vengo la mayoría de las personas no tiene respuesta a eso. No puedo expresar en palabras ni actos lo agradecido que estoy de mis padres por haber hecho sacrificios inmensos; como alejarnos del lugar que nos vio nacer y de nuestra familia, por duro que fuese en su momento, para tener oportunidades universitarias en nuestra vida. Siempre me dijeron que estudiara porque querían que fuera más que ellos, hoy les digo que si logro ser la mitad de lo que ellos son, estoy seguro que estarían satisfechos de lo que han logrado conmigo.

Agradezco profundamente a quienes me formaron en la carrera. Al profesor Julio Aracena (a quien ahora puedo llamar simplemente Julio), quien me ayudó a tener varias de mis mejores vacaciones pagadas a distintas escuelas de verano e invierno, además de compartir muchas instancias y buenos momentos, y a quien debo en parte la iniciación en el área que me gusta. A mis profesores guía, Christopher Thraves y Sebastián Niklitschek, quienes me dieron la confianza y apoyo en todo este camino, y a quienes les di una lección de paciencia, pero no se preocupen, que todo estaba premeditado. Mención especial para Sebastián, quien aparte de guiarme académicamente se transformó en un excelente amigo, y un apoyo incondicional en todo momento en el que lo necesité. No puedo olvidar a mis amigos Felipe y Jorge, con quienes logramos salir victoriosos de este largo camino, y compartí un montón de aventuras matemáticas y ñoñas en mi vida. Finalmente, agradezco a la profesora Mónica Selva por ser la más apañadora jefa de carrera, quien me dejaba hacer asados, tocatas y cuanta actividad se me ocurriera. A ella y a la sra. María Eugenia, les estaré eternamente agradecido porque hicieron que el DIM, más allá de ser un lugar de trabajo, se sintiera como un segundo hogar.

Toda esta aventura quiero dedicarla a quienes estuvieron conmigo desde el principio. A aquellos que me dieron la vida, mis padres, quienes sufrieron y se alegraron conmigo en todo este proceso. Mis hermanos, uno ayudándome en lo que pudo y al otro enseñándole lo que puedo. A toda mi familia que me apoyó desde que era pequeño. En especial, agradezco a quien me acompaña hace más de tres años, Catalina, porque compartimos alegrías y penurias, y te has convertido en mi cómplice eterno hace más de tres años, te amo.

Índice general

1	Resumen	11
2	Introducción	13
2.1	Estructura de esta memoria	14
3	Encaminamiento en redes	15
3.1	Definición del problema	15
3.2	Aprendizaje y Predicción con Ayuda de Expertos	16
3.2.1	Modelo Original	16
3.2.2	Modelación al problema de encaminamiento	17
3.3	Objetivos	19
3.4	Estado del arte	19
4	Ambientes	21
4.1	Ambiente Probabilista	21
4.2	Ambiente Markoviano	23
4.2.1	Modelo probabilista de fallos como un caso particular del modelo markoviano	28
4.3	Ambiente Adversarial	30
5	Predictores	31
5.1	Preliminares	31

5.2	Predictor Uniforme (PUNIF)	34
5.3	Follow the perturbed leader (FTPL)	36
5.3.1	Probabilidad de elección de un experto	36
5.4	Follow the perturbed leader modificado (MFTPL)	41
5.4.1	Probabilidad de elección de un experto	46
5.5	Resumen de cotas	50
6	Experimentos	51
6.1	Métodos de Monte-Carlo	52
6.2	Ambiente Adversarial	54
6.2.1	Fallos	54
6.2.2	Pérdida acumulada	56
6.2.3	Arrepentimiento acumulado	57
6.3	Ambiente Probabilista	58
6.3.1	Fallos	59
6.3.2	Pérdida acumulada	60
6.3.3	Arrepentimiento acumulado	61
6.4	Ambiente Markoviano	62
6.4.1	Fallos	63
6.4.2	Pérdida acumulada	64
6.4.3	Arrepentimiento acumulado	65
6.5	Datos Reales	67
6.5.1	F_{ME}^P usando valor de fallo mayor o igual a 1400.	69
6.5.2	F_{ME}^P usando valor de fallo mayor o igual a 1000.	70
6.5.3	F_{ME}^P usando valor de fallo mayor o igual a 600.	71
7	Conclusiones	73

Índice de cuadros

3.1	Ejemplo de tabla de costos para 3 expertos y 4 instantes de tiempo.	18
3.2	Pérdidas acumuladas de cada experto en el ejemplo de 3 expertos y 4 instantes de tiempo.	18
3.3	Elecciones y su predicción del predictor I_t para el ejemplo de 3 expertos y 4 instantes de tiempo.	19
5.1	Tasas de fallo para un ambiente probabilista.	35
5.2	Ejemplo de costos arbitrarios y fallos en un ambiente probabilista.	35
5.3	Tasas de fallo estimadas para un ambiente probabilista.	36
5.4	Costos de cada experto para el ejemplo 5.3.	39
5.5	Pérdidas acumuladas de cada experto para el ejemplo 5.3.	39
5.6	Perturbación aleatoria $Unif[0, 0.1]$	40
5.7	Tabla resumen de la ejecución de FTPL.	40
5.8	Probabilidades de transición estimadas para el ejemplo de la escalera que utiliza MFTPL markoviano.	45
5.9	Tasas de fallo estimadas para el ejemplo de la escalera que utiliza MFTPL probabilista.	45
5.10	Perturbación $Unif[0, \sqrt{nN}]$ para ejemplo de la escalera.	46
5.11	Resumen de cotas para $\mathbb{E}[X]$	50
6.1	Extracto de la tabla de costos para el ejemplo de un ambiente adversarial.	54
6.2	Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente adversarial.	55
6.3	Valores esperados y desviación estándar de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente adversarial.	56
6.4	Valores esperados y desviación estándar del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente adversarial.	57

6.5	Tasas de fallo de cada experto en el ejemplo de un ambiente probabilista.	58
6.6	Extracto de la tabla de costos para el ejemplo de un ambiente probabilista.	58
6.7	Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente probabilista.	60
6.8	Valores esperados y desviación estándar de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente probabilista.	61
6.9	Valores esperados y desviación estándar del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente probabilista.	62
6.10	Probabilidades de transición de la cadena de Markov de fallos de cada experto en el ejemplo de un ambiente markoviano.	62
6.11	Extracto de la tabla de costos para el ejemplo de un ambiente adversarial.	63
6.12	Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente markoviano.	64
6.13	Valores esperados de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente markoviano.	65
6.14	Valores esperados del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente markoviano	66
6.15	Nodos que componen la red de latencias <i>NLNog</i>	67
6.16	Extracto de la tabla de datos reales a utilizar.	68
6.17	Expertos restantes considerados en los experimentos con datos reales que conectan Nurita con Singapore.	68
6.18	Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 1400.	70
6.19	Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 1000.	71
6.20	Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 600.	72

Índice de figuras

3.1	Gráfica de una red con N caminos entre el nodo fuente S y el nodo destino T.	16
5.1	Pérdida acumulada de FTPL y el mejor experto para ejemplo de la escalera	41
5.2	Pérdida acumulada de los predictores para una ejecución de estos sobre el ejemplo de la escalera usando una perturbación aleatoria $Unif[0, 0.1]$	43
5.3	Pérdida acumulada de los predictores para el ejemplo de la escalera bajo una perturbación aleatoria $Unif[0, \sqrt{nN}]$, la cual minimiza la pérdida acumulada para el caso de FTPL.	44
6.1	Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Adversarial)	55
6.2	Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Adversarial).	56
6.3	Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Adversarial)	57
6.4	Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Probabilista). . . .	59
6.5	Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Probabilista).	60
6.6	Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Probabilista).	61
6.7	Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Markoviano). . . .	63
6.8	Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Markoviano).	65
6.9	Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Markoviano).	66
6.10	Razón F_{ME}^P para un valor de fallos fijado en 1400 (Datos Reales).	69
6.11	Razón F_{ME}^P para un valor de fallos fijado en 1000 (Datos Reales).	70
6.12	Razón F_{ME}^P para un valor de fallos fijado en 600 (Datos Reales).	71

1. Resumen

En el presente trabajo estudiamos el problema de encaminamiento en redes de comunicación utilizando el modelo de *Aprendizaje y Predicción con Ayuda de Expertos*, incorporando el supuesto de que los caminos presentes puedan fallar. Efectuamos una presentación del modelo y su adaptación al problema considerado. Posteriormente introducimos una nueva noción a la cual denominamos *ambientes*, la cual nos permite modelar la aparición de los fallos en los caminos (llamados también expertos). Estudiamos qué ocurre con los predictores (cualquiera sea) cuando actúan bajo estos supuestos de fallo. En particular, mostramos cotas para el valor esperado de fallos de predictores que actúan bajo los ambientes probabilista y markoviano, además de mostrar, en casos específicos donde las probabilidades de fallo se mantienen constantes en el tiempo para cada experto, que el valor esperado de fallos se obtiene explícitamente y es independiente del predictor considerado.

En una siguiente etapa, estudiamos los predictores PUNIF y FTPL, y mostramos cotas para los valores esperado de fallo en los distintos ambientes, para luego dar paso a nuevos predictores definidos por nosotros, MFTPL probabilista y MFTPL markoviano, que corresponden a una generalización del predictor FTPL, pero que buscan utilizar la información de la aparición de los fallos bajo un ambiente probabilista y markoviano, respectivamente. Finalizamos esta etapa resumiendo todas las cotas encontradas para los predictores mencionados.

Para finalizar el trabajo, sometemos a todos los predictores estudiados a simulaciones numéricas bajo ambientes adversarial, probabilista y markoviano, además de datos reales facilitados por el proyecto PANACEA. En estos experimentos se evidencia que el mejor predictor es MFTPL markoviano, quien generaliza el comportamiento de MFTPL probabilista en todos los ambientes.

2. Introducción

El encaminamiento de paquetes en redes de comunicación, como Internet, es una tarea fundamental para el buen funcionamiento de las mismas. Dicha tarea consiste en decidir qué camino usar para enviar un flujo de paquetes de datos desde un origen a un destino. Usualmente, tal decisión es estática, es decir, una vez escogida la ruta, esta será utilizada de manera fija. Tal elección es buena cuando los costos de envío no cambian. En redes de comunicación como Internet, este fenómeno no ocurre, y la calidad de las rutas disponibles, cuyos costos pueden ser considerados como su latencia o tiempo de envío de un paquete, cambian periódicamente, pudiendo incluso fallar, lo que se traduce en un corte en la comunicación y el no envío de los datos. Uno de los problemas centrales en Estadística y Machine Learning es el problema de predecir eventos futuros basados en observaciones pasadas, como bien dice Cesa-Bianchi *et al.* en [6]. Este comportamiento ha abierto el estudio y uso del encaminamiento dinámico. En dicho tipo de encaminamiento, se permite que, dentro de un mismo flujo de datos, el camino se adapte según la calidad de la comunicación, pudiendo así cambiar el flujo a un mejor camino (de menor costo) o a uno que no tenga fallos. Una forma de estudiar este problema es mediante el modelo de *Aprendizaje y Predicción con Ayuda de Expertos* [4], el cual utiliza la información pasada de cada camino para decidir la mejor elección futura.

Podemos encontrar diversos tipos de redes en nuestro diario vivir, cada una con comportamientos particulares. Por ejemplo, una red de caminos en una ciudad muy poblada como Concepción, donde existen horarios punta en los que el tráfico vehicular es muy elevado, como los ingresos y salidas a los trabajos diurnos y los horarios de almuerzo. El lector puede notar fácilmente que tales eventos en el tráfico ocurren de manera periódica en los puntos más concurridos de su ciudad, y de acuerdo a su experiencia, puede tomar decisiones sobre qué ruta es más conveniente a una hora específica.

Imagine ahora que se encuentra en una embotelladora que consta de una línea de producción en la cual se lava la botella a utilizar, posteriormente es llenada, etiquetada y almacenada. El objetivo es obtener un producto terminado en perfectas condiciones, y para ello todos los puntos que componen el proceso deben funcionar de manera óptima. Si se encontrase una botella mal etiquetada, lo más probable es que las botellas que pasaron antes y después de esa hayan sufrido el mismo error producto de una falla en la maquinaria del proceso de etiquetado.

Piense ahora en el caso de una empresa multinacional que constantemente está enviando infor-

mación confidencial y valiosa de sus clientes entre sus sucursales mediante correos electrónicos. Es de suma importancia que toda información llegue a destino y no sea intervenida, pero esta red está expuesta a ataques maliciosos que buscan cortar la comunicación y robar la información. Desafortunadamente para la empresa, tales ataques ocurren de manera irregular y en los momentos menos esperados, por lo que deben estar alertas ante cualquier falla en la comunicación.

En cada una de las redes expuestas anteriormente existe un comportamiento marcado de ciertos eventos. En el primer caso vimos que nos podemos encontrar con congestión vehicular periódicamente, en la segunda vimos que es muy probable que cuando ocurra una falla en un producto esta se repita, y por último, ejemplificamos un comportamiento de una red que puede fallar de manera inesperada. Estos comportamientos son la base para nuestro objetivo en esta memoria.

*Mejor que de nuestro juicio, debemos fiarnos del cálculo algebraico.
Leonhard Euler*

Con el fin de orientar al lector respecto a la estructura de esta memoria y dar algunas pistas importantes a considerar de cada capítulo, presentamos a continuación una breve información de cada uno de ellos y cómo interactúan entre sí.

2.1 Estructura de esta memoria

Esta memoria se compone de 6 capítulos. El primero y sobre el cuál se encuentra en este momento, corresponde a la *Introducción*, en la que se da una pincelada del problema que estamos interesados en abordar y algunos ejemplos que tratan de mostrar ciertos comportamientos que nos interesan en las redes de comunicación.

El segundo capítulo, llamado *Encaminamiento en redes* presenta la definición del problema que abordaremos en esta memoria, el modelo de *aprendizaje y predicción con ayuda de expertos* que utilizamos para atacarlo, los objetivos que hemos propuesto alcanzar al desarrollar la memoria y el estado del arte referente al problema de encaminamiento en redes con fallos. Además, se introduce la noción de *predictor* y los elementos básicos que nos permitirán medir el desempeño de estos.

En el tercer capítulo definimos tres tipos de modelos para determinar fallos en una red de comunicación, a los que definimos como *Ambientes*. Recomendamos que lea la introducción previo a este capítulo para tener una idea más intuitiva del por qué considerar los modelos de fallo que se presentan. Además, encontrará algunos resultados para acotar el valor esperado de fallos que un predictor puede cometer bajo los distintos modelos de fallo.

En el cuarto capítulo entramos en tierra derecha en el estudio y definición de predictores para el problema de encaminamiento en redes con fallos. Presentamos dos predictores de la literatura y proponemos otros dos. Para cada uno de ellos presentamos resultados teóricos para acotar el valor esperado de fallos y algunos ejemplos prácticos que muestran las medidas de cada predictor y su funcionamiento. Al final del capítulo presentamos un resumen con las cotas que se obtienen hasta ese punto.

El quinto capítulo trata de *Experimentos*. Iniciamos con una revisión de los métodos de Monte-Carlo, herramienta que usaremos para estimar experimentalmente valores esperados de fallo y otras medidas sobre los predictores. Presentamos experimentos para cada uno de los modelos de fallo presentados en el capítulo 3 y al final sometemos a los predictores a un problema con datos reales. Finalmente, en el capítulo 6, realizamos comentarios y conclusiones del trabajo realizado.

3. Encaminamiento en redes

Los problemas de encaminamiento consisten en encontrar una ruta para enviar un paquete (físico o de información) desde un punto de origen a uno de destino. Podemos encontrar este tipo de problemas en transporte, abastecimiento de productos, redes de internet, entre otros. En todos y cada uno de ellos el objetivo es enviar un paquete en el menor tiempo posible o al menor costo. Las rutas disponibles para tales envíos están definidas por una red de comunicación, la cual se modela matemáticamente por un grafo (dirigido en la mayoría de los casos). Este capítulo introduce al lector en el problema a estudiar, formalizando matemáticamente el modelo a utilizar y entregando las nociones básicas que utilizaremos a lo largo de esta memoria. Además, presentamos los objetivos y repasamos lo que se ha hecho en relación al problema que deseamos abordar.

3.1 Definición del problema

El problema a estudiar es conocido como *Online shortest path problem* [17], donde contamos con una red dirigida y sin ciclos, con un nodo fuente S y un nodo sumidero T , y deseamos enviar paquetes de información desde S a T al menor costo posible (usualmente considerado como tiempo o latencia). Existen N caminos a nuestra disposición que unen S y T , no necesariamente disjuntos, donde nuestro objetivo es, en cada instante de tiempo t , escoger uno de esos caminos para enviar la información basados en alguna política de decisión. Esta forma de modelar el problema es una variación del clásico *Multi-Armed Bandit problem* (Liu y Zhao [14]). Estos caminos tienen la particularidad de que pueden fallar en ciertos instantes, lo cual se traduce en un no envío de la información y una alta penalización en su costo. Con esto, nuestro problema no sólo consiste en minimizar los costos de envío, sino que también en minimizar el número de veces que se escoge un camino que falla.

Nuestra elección estará determinada por un **predictor**, quien elige uno y sólo un camino por cada instante de tiempo t , basando su elección en el historial de pérdidas y fallos hasta el instante anterior ($t - 1$) de cada experto. Esta técnica es la utilizada en el modelo de *aprendizaje y predicción con ayuda de expertos*, y es sobre la cual se desarrollará esta memoria.

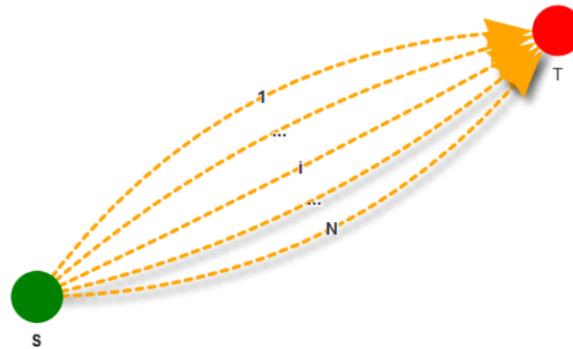


Figura 3.1: Gráfica de una red con N caminos entre el nodo fuente S y el nodo destino T .

Se consideran tres modelos que determinan los fallos de cada camino: adversarial (arbitrariamente), probabilista (definidas por una tasa de fallos) y markoviano (mediante una cadena de markov de dos estados: fallo y no fallo).

La noción de predictor es la clave en esta memoria, es la base de todos nuestros resultados y sobre lo que trabajaremos. Esto nos motiva a formalizarla en la siguiente definición:

Definición 3.1.1 — Predictor. Sea $\{1, \dots, N\}$ un conjunto finito a cuyos elementos denotaremos por expertos. Llamaremos predictor a la función I_t que en cada instante de tiempo **escoge** un experto

$$I_t: \mathbb{N} \rightarrow \{1, \dots, N\}$$

$$t \mapsto I_t = i$$

donde la **predicción** de I_t coincide con la del experto i elegido.

3.2 Aprendizaje y Predicción con Ayuda de Expertos

Aprendizaje y predicción con ayuda de expertos es un modelo matemático enmarcado en la teoría de predicción de secuencias de valores individuales. Primero enunciaremos su modelo original, y luego daremos los respectivos detalles que sumergirán a nuestro problema en este modelo.

3.2.1 Modelo Original

La predicción con ayuda de expertos está basada en el siguiente protocolo de decisiones secuenciales ([4], [6]):

- Quien toma la decisión es un **predictor** cuyo objetivo es predecir una secuencia desconocida y_1, y_2, \dots de elementos de un espacio de salida \mathcal{Y} .
- Las predicciones del predictor, denotadas por la secuencia $\hat{p}_1, \hat{p}_2, \dots$ pertenecen al espacio de decisión \mathcal{D} , el cual se asume como un subconjunto convexo de un espacio vectorial. En

algunos casos se asume que $\mathcal{D} = \mathcal{Y}$, pero en general pueden ser diferentes.

- El **predicador calcula su predicción de manera secuencial**, y su desempeño es comparado con un conjunto de **expertos**. Más precisamente, en cada instante t , el predictor tiene acceso al conjunto $\{f_{i,t-1} : i \in \{1, \dots, N\}\}$, donde $f_{i,t-1} \in \mathcal{D}$ corresponde al valor propuesto por el experto i en el instante $(t-1)$, y N es la cantidad total de expertos.
- Basándose en los valores propuestos por los expertos, el predictor **elige** creerle a uno de los expertos y entrega su **predicción** \hat{p}_t (correspondiente al valor que arroja el experto en el que cree el predictor, es decir, $\hat{p}_t = f_{i,t}$, para algún i elegido) para la salida y_t . Una vez que \hat{p}_t es calculada, el valor a predecir, y_t , es revelado.
- Las predicciones del predictor y los valores arrojados por los expertos reciben una puntuación usando una función no negativa $\ell : \mathcal{D} \times \mathcal{Y} \rightarrow \mathbb{R}^+$, a la cual llamaremos **función de pérdida**. De esta forma, en cada instante de tiempo, el predictor incurre en una pérdida $\ell(\hat{p}_t, y_t)$, mientras que cada experto i tiene una pérdida $\ell(f_{i,t}, y_t)$.

La función de pérdida definida anteriormente nos entrega un dato para cada instante de tiempo t . Es en base a esta función que definiremos dos medidas de interés para nuestro predictor: la pérdida acumulada y el arrepentimiento acumulado (hasta un horizonte de tiempo n).

El objetivo del predictor es mantener lo más pequeño posible el arrepentimiento acumulado con respecto a cada experto i hasta un horizonte de tiempo n , $R_{i,n}$. Esta cantidad está definida por

$$R_{i,n} = \sum_{t=1}^n \left(\ell(\hat{p}_t, y_t) - \ell(f_{i,t}, y_t) \right) = L_n - L_{i,n}$$

donde $L_n = \sum_{t=1}^n \ell(\hat{p}_t, y_t)$ denota la **pérdida acumulada** del predictor, mientras que $L_{i,t} = \sum_{t=1}^n \ell(f_{i,t}, y_t)$ denota la pérdida acumulada del experto i . De esta forma, podemos resumir lo anterior en mantener lo más pequeño posible el **arrepentimiento acumulado** R_n :

$$R_n = \max_{i \in \{1, \dots, n\}} R_{i,n} = L_n - \min_{i \in \{1, \dots, N\}} L_{i,n}$$

Adicionalmente, definimos por $r_{i,t} = \ell(\hat{p}_t, y_t) - \ell(f_{i,t}, y_t)$ el **arrepentimiento instantáneo**. Uno puede interpretar a $r_{i,t}$ como el arrepentimiento que sufre el predictor de no haber escuchado el consejo del experto i justo después de que la t -ésima salida y_t ha sido revelada.

3.2.2 Modelación al problema de encaminamiento

La adaptación del modelo de aprendizaje y predicción con ayuda de expertos aplicado al problema de encaminamiento consiste en hacer una correcta lectura de cada uno de los conjuntos y valores del modelo original. Para ser precisos, consideraremos lo siguiente:

- Cada camino que une S y T será considerado un **experto**. De aquí en adelante se usará indistintamente *experto* y *camino*.
- El espacio de decisión \mathcal{D} y el espacio de salida \mathcal{Y} coinciden y corresponden al intervalo $[0, 1]$ (usado en [12, 17]), los cuales representan los valores posibles para los costos de envío, que para nosotros quedará determinado por la latencia del camino, donde una **latencia igual a 1 representa un fallo**.
- El valor propuesto por cada experto i en el instante t , $f_{i,t}$, corresponderá a la latencia del camino y es un valor desconocido que sólo es revelado hasta que se entrega la predicción del predictor.

- y_t corresponde al menor valor entre las latencias de los expertos en el instante t , es decir, $y_t = \min_{i \in \{1, \dots, N\}} f_{i,t}$.
- Para cada experto, la función de pérdida ℓ , estará definida por

$$\ell(f_{i,t}, y_t) = \begin{cases} f_{i,t} - y_t & f_{i,t} \text{ no es un fallo} \\ 1 & \text{e.o.c} \end{cases}$$

Para el caso del predictor, su pérdida en el instante t corresponderá a aquella asociada al experto i elegido en dicho instante.

Como ha quedado explícito en el último punto, la misión de nuestro predictor es escoger, en cada instante t , a uno de los expertos, a esta elección la denotaremos por I_t .

Ejemplo

Para efectos de análisis de problemas numéricos, expresaremos los datos de los costos en una tabla como la que se muestra en el Cuadro 3.1:

t	E_1	E_2	E_3
1	0.10	0.50	1.00
2	0.20	1.00	0.10
3	0.30	0.20	1.00
4	1.00	0.80	0.10

Cuadro 3.1: Ejemplo de tabla de costos para 3 expertos y 4 instantes de tiempo.

donde la primera columna representa el instante de tiempo considerado. E_1, E_2, E_3 aluden a los tres expertos en juego, y los valores restantes de la tabla corresponden a las predicciones de cada experto E_i en el instante t respectivo (conocidas como $f_{i,t}$).

Observamos que en este experimento, el horizonte de tiempo considerado es $n = 4$, la cantidad de expertos es $N = 3$, el experto E_1 falla en el instante $t = 4$, E_2 falla en el instante $t = 2$, mientras que el experto E_3 falla en los instantes $t = 1$ y $t = 3$. Además, podemos observar las pérdidas acumuladas de cada experto en el Cuadro 3.2:

$L_{E_1,4}$	$L_{E_2,4}$	$L_{E_3,4}$
1.60	2.50	2.20

Cuadro 3.2: Pérdidas acumuladas de cada experto en el ejemplo de 3 expertos y 4 instantes de tiempo.

Supongamos ahora que nuestro predictor realiza las elecciones $I_1 = E_1, I_2 = E_2, I_3 = E_2, I_4 = E_2$, sus predicciones serán las mostradas en el Cuadro 3.3:

t	I_t	\hat{p}_t
1	E_1	0.10
2	E_2	1.00
3	E_2	0.20
4	E_3	0.80

Cuadro 3.3: Elecciones y su predicción del predictor I_t para el ejemplo de 3 expertos y 4 instantes de tiempo.

en tal caso, tenemos que

- $L_4 = 2.10$
- $R_4 = L_4 - L_{E_1,4} = 2.10 - 1.60 = 0.50 > 0$

del arrepentimiento acumulado (R_n) podemos concluir que nuestro predictor no tomó tan buenas decisiones como el mejor de los expertos, en este caso E_1 , quien tuvo la menor pérdida acumulada.

Dentro de nuestros objetivos estará definir y estudiar predictores que se asemejen lo más posible al mejor experto, lo cual se traduce en un arrepentimiento acumulado cercano a 0. La clave en los predictores está en cómo utilizar la información pasada de los expertos para tomar la mejor decisión en el presente, cosa que nuestro predictor utilizado en el ejemplo anterior no hizo.

3.3 Objetivos

En esta memoria se propone diseñar y analizar algoritmos de encaminamiento dinámico usando como base el modelo de *Aprendizaje y Predicción con Ayuda de Expertos*, que nos ayudará a predecir el comportamiento de la red, sus fallos y la calidad de sus enlaces. Los objetivos específicos de este proyecto son:

- O1:** Diseñar algoritmos de encaminamiento dinámicos usando el modelo de *Aprendizaje y Predicción con Ayuda de Expertos*. Es decir, diseñar predictores que sean capaces de aprender y predecir fallos en base a la información histórica en una red de comunicaciones para luego decidir por qué camino encaminar paquetes de datos en cada instante de tiempo.
- O2:** Evaluar los algoritmos propuestos empíricamente para determinar el arrepentimiento acumulado que producen y el número de caminos fallidos que escogen.
- O3:** Evaluar los algoritmos propuestos analíticamente. Es decir, demostrar cotas superiores para su arrepentimiento acumulado y el número de caminos fallidos que escoge.

3.4 Estado del arte

Las investigaciones a este problema comenzaron la década del 50 (ver, por ejemplo, Blackwell [3] y Hannan [11] para algunos resultados básicos) y siguieron con fuerza en la década del 90 con los trabajos de Vovk [20], Littlestone y Warmuth [13] y Cesa-Bianchi *et al.* [6]. Estos resultados muestran que para cualquier función de pérdida acotada, si el predictor tiene acceso a las pérdidas pasadas de cada experto, entonces es posible construir algoritmos que, para cualquier comportamiento del ambiente, se comportan tan bien como el mejor de los N expertos. Más precisamente, la pérdida acumulada de estos algoritmos es, a lo más tan grande como aquella del mejor experto más una cantidad proporcional a $\sqrt{\ln N/n}$ para cualquier función de pérdida, donde n es la cantidad de instantes de tiempo total a considerar.

Este problema ha sido modelado de distintas formas. Zou *et al* [21] formulan el problema como un *combinatorial bandit optimization problem*, y consideran varios escenarios que difieren en dónde se realiza la decisión de ruteo (nodo fuente o en cada nodo interior) y en la cantidad de información disponible para tomar dicha decisión. Para ellos los costos o latencias son dependientes del tiempo y modeladas por variables aleatorias i.i.d. Tehrani y Zhao en [18] también consideran este enfoque respecto a los costos y proponen una versión distribuida del algoritmo Bellman-Ford, DBFL (*Distributed Bellman Ford with Learning*), donde cada nodo conoce la información de sus vecinos y el camino para llegar desde el nodo fuente al destino se va construyendo de nodo en nodo.

En otros trabajos, como [5] y [9] se estudia este mismo problema pero desde un enfoque adversarial, donde las latencias de cada conexión en la red puede cambiar de manera arbitraria.

Otro enfoque es ver a cada camino disponible como un ente total, y conocer sólo el costo total de este. Tal modelo proviene del clásico *Multi-Armed Bandit problem (MAB)*. En el clásico MAB hay N brazos independientes. En cada instante, el jugador elige un brazo para jugar. Un brazo, cuando es jugado, incurre en un costo variable i.i.d. de una distribución desconocida, que sólo es revelado una vez que es jugado. El paso natural al problema del camino más corto con costo variable es considerar cada camino desde el nodo fuente al de destino como un brazo. Liu y Zhao en [14] consideran este caso, al igual que Kalai y Vempala [12], György *et al.* [8] y Takimoto y Warmuth [17].

En [10] György, Linder y Lugosi abordan variantes del MAB, una de ellas considera que el predictor tiene acceso a las pérdidas correspondientes al camino que escogió en tal instante, pero a nada más. La segunda variante es más restrictiva, y consiste en que se conoce la pérdida total del camino elegido, pero no las pérdidas individuales de cada arista que compone el camino.

En todas, el objetivo es minimizar el *regret* o arrepentimiento acumulado (la diferencia de la latencia acumulada hasta un horizonte de tiempo del predictor a usar respecto al mejor camino).

4. Ambientes

Un ambiente corresponde al escenario en el cual se expresan los fallos de cada uno de los expertos. La principal contribución de este trabajo consiste en la incorporación de fallos al modelo en estudio. Nosotros modelaremos la aparición de estos fallos de diversas maneras; apareciendo a razón de una tasa de fallo (usado por Anantharam en [1]), en cuyo caso diremos que el problema está bajo un *ambiente probabilista*, determinados por una cadena de Markov (usado por Tekin [19] en el *Multi Armed Bandit Problem* y previamente por Anantharam *et al.* en [2]) en un *ambiente markoviano*, o simplemente determinados por un adversario (Cesa-Bianchi y Lugosi en [5], y György *et al.* en [9]), bajo un *ambiente adversarial*.

Cada ambiente presenta características que marcan diferencias entre uno y otro, motivando el interés por abordarlos. El ambiente adversarial se puede entender como aquel del peor caso, donde incluso los costos en nuestra red de comunicación son definidos por un adversario, quien tiene por objetivo proponer el peor caso posible a nuestros predictores. En el caso probabilista, los fallos son definidos mediante tasas de fallo, mientras que en el ambiente markoviano, los fallos son determinados por una cadena de Markov de dos estados, uno de fallo y otro de no fallo.

En lo que sigue, cuando hablemos de modelo probabilista, markoviano o adversarial, siempre estaremos haciendo énfasis en el modelo utilizado para describir la ocurrencia de fallos.

Definición 4.0.1 — Ambiente. Entenderemos por ambiente al modelo que determina la manera en que falla cada experto.

4.1 Ambiente Probabilista

Un ambiente probabilista es aquel donde se asume que cada experto falla a una cierta tasa. Esto se formaliza en la siguiente definición.

Definición 4.1.1 — Modelo probabilista de fallos. En un *modelo probabilista de fallos*, la probabilidad de fallo de cada experto i en un instante de tiempo t , está definida por una *tasa de*

fallo $p_{i,t}$, es decir,

$$\mathbb{P}(i \text{ falla en } t) = p_{i,t}$$

donde $\forall t_1, t_2, \forall i, j$, p_{i,t_1} es independiente de p_{j,t_2}

Intuitivamente, podemos pensar que en un ambiente probabilista, en cada instante de tiempo t , un experto lanza una moneda con dos caras, *fallo* y *no fallo*, donde la cara de *fallo* posee probabilidad $p_{i,t}$ de salir sorteada en el instante t .

La definición anterior nos plantea el caso más general que podemos encontrar en un ambiente probabilista, aquel donde las tasas de fallo pueden cambiar en cada instante de tiempo.

Nuestro objetivo es contar la cantidad de fallos que cometen nuestros predictores en estudio. Para este efecto, de ahora en más definiremos la variable aleatoria X como la cantidad total de fallos que comete un predictor.

Definición 4.1.2 — Cantidad de fallos que comete un predictor. Sea X_t la variable aleatoria que determina si el predictor falla en un instante t , entonces

$$X_t = \begin{cases} 1, & \text{Si la elección del predictor en el instante } t, I_t, \text{ es un fallo} \\ 0, & \text{En caso contrario} \end{cases}$$

y la cantidad de fallos que comete un predictor hasta un horizonte de tiempo n está definida por

$$X = \sum_{t=1}^n X_t.$$

Nota Los predictores que usamos son probabilistas, por lo tanto tiene sentido definir la variable aleatoria X .

El siguiente teorema nos entrega una cota para el caso en el que tales tasas son constantes en el tiempo.

Teorema 4.1.1 Bajo un ambiente probabilista, donde cada experto i tiene una tasa de fallos $p_i, \forall t$, el valor esperado de fallos $\mathbb{E}[X]$ de cualquier predictor hasta el tiempo n satisface

$$n \cdot \min_j \{p_j\} \leq \mathbb{E}[X] \leq n \cdot \max_j \{p_j\}.$$

Demostración. El valor esperado de fallo del predictor I_t en el instante t está determinado por,

$$\begin{aligned} \mathbb{E}[X_t] &= 0 \cdot \mathbb{P}(X_t = 0) + 1 \cdot \mathbb{P}(X_t = 1) \\ &= \mathbb{P}(X_t = 1) \\ &= \mathbb{P}(I_t \text{ es un fallo}) \\ &= \sum_j \mathbb{P}(I_t = j \wedge j \text{ falla}) \\ &= \sum_j \mathbb{P}(I_t = j) \cdot \mathbb{P}(j \text{ falla en } t), \text{ por independencia de los eventos} \\ &= \sum_j \mathbb{P}(I_t = j) \cdot p_j \end{aligned}$$

Como $X = \sum_{t=1}^n X_t$, su valor esperado es

$$\begin{aligned}
 \mathbb{E}[X] &= \mathbb{E}\left[\sum_{t=1}^n X_t\right] \\
 &= \sum_{t=1}^n \mathbb{E}[X_t] \\
 &= \sum_{t=1}^n \sum_j \mathbb{P}(I_t = j) \cdot p_j \\
 &\leq \max_j p_j \cdot \sum_{t=1}^n \sum_j \mathbb{P}(I_t = j) \overset{=1}{=} \\
 &= \max_j p_j \cdot \sum_{t=1}^n 1 \\
 &= n \cdot \max_j p_j
 \end{aligned}$$

y de manera análoga, se concluye que $\mathbb{E}[X] \geq n \cdot \min_j p_j$, con lo que concluye la demostración. ■

De este resultado se desprende el siguiente corolario, el cual nos da un valor exacto para el valor esperado de fallos en el caso particular de un modelo probabilista.

Corolario 4.1.2 Si todos los expertos comparten la misma tasa de fallo p , entonces, $\mathbb{E}[X] = n \cdot p$.

Demostración. Se obtiene directo del teorema 4.1.1. ■

Nota

Es interesante notar que los resultados vistos anteriormente no dependen del predictor. En particular, el corolario es un resultado muy fuerte, pues nos entrega el valor exacto del valor esperado de fallos, independiente de los costos de la tabla y de la información que use nuestro predictor para tomar sus decisiones.

4.2 Ambiente Markoviano

Un proceso estocástico $\mathbf{X} = \{X_t : t \in T\}$ es una colección de variables aleatorias. La variable t a menudo representa tiempo (para nuestro caso, será más bien un *instante de tiempo*), y en tal caso, el proceso \mathbf{X} modela el valor de una variable aleatoria X que cambia en el tiempo.

Llamaremos X_t al estado del proceso en el tiempo t y diremos que X_t es un proceso de tiempo (o a tiempo) discreto si la variable que indexa el proceso, t , toma valores en un conjunto discreto.

Definición 4.2.1 — Cadena de Markov, (15). Un proceso estocástico a tiempo discreto X_0, X_1, X_2, \dots es una cadena de Markov para todos a_0, \dots, a_t en el espacio en el que el proceso toma valores, si

$$\mathbb{P}(X_t = a_t | X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \dots, X_0 = a_0) = \mathbb{P}(X_t = a_t | X_{t-1} = a_{t-1}).$$

La definición anterior indica que el estado X_t depende del estado previo X_{t-1} , pero es independiente de la historia particular de cómo el proceso llegó hasta el estado X_{t-1} . Esto se conoce como *la propiedad de Markov*. Es importante notar que la propiedad de Markov no implica que X_t es independiente de las variables aleatorias X_0, \dots, X_{t-2} , esto sólo implica que cualquier dependencia de X_t sobre el pasado, está capturada en el valor de X_{t-1} . Sin pérdida de generalidad, podemos asumir que el espacio de estados discretos de una cadena de Markov es $\{0, 1, 2, \dots, n\}$ (o $\{0, 1, 2, \dots\}$ si es infinito numerable). La probabilidad de que el proceso se mueva desde un estado i a un estado j en un paso (o un instante de tiempo) está definida por

$$P_{i,j} := \mathbb{P}(X_t = j | X_{t-1} = i)$$

y se conoce como *probabilidad de transición en un paso*.

La propiedad de Markov implica que la cadena de Markov está únicamente definida por la *Matriz de transición en un paso* \mathbf{P}

$$\mathbf{P} = \begin{pmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,j} & \dots \\ P_{1,0} & P_{1,1} & \dots & P_{1,j} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ P_{i,0} & P_{i,1} & \dots & P_{i,j} & \dots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}$$

donde cada componente de la matriz representa una probabilidad de transición en un paso desde un estado a otro, los cuales están definidos por la posición de las columnas y filas de \mathbf{P} . Además, se satisface que

$$\sum_{j \geq 0} P_{i,j} = 1$$

donde los sumandos corresponden a todos los posibles movimientos que se pueden realizar estando en el estado i , por tanto, tal probabilidad debe ser efectivamente igual a 1.

Denotamos por $p_i(t)$ a la probabilidad de que el proceso esté en el estado i en el tiempo t . Asimismo, $\vec{p}(t) = (p_0(t), p_1(t), p_2(t), \dots)$ denota el vector que da la distribución de los estados de la cadena en el instante t . De esta forma, si sumamos sobre todos los posibles estados en el instante $t - 1$, obtenemos que

$$p_i(t) = \sum_{j \geq 0} p_j(t-1)P_{j,i}$$

o visto de otra forma,

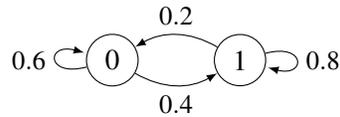
$$\vec{p}(t) = \vec{p}(t-1)\mathbf{P}$$

Las probabilidades de transición en un paso nos entregan la información inmediata de lo que ocurrirá en un instante de tiempo siguiente. A su vez, se mencionó que X_t es dependiente de las variables X_0, X_1, \dots, X_{t-2} , lo cual se ve reflejado en la *probabilidad de transición en m -pasos*:

$$\forall m \geq 0, P_{i,j}^m = \mathbb{P}(X_{t+m} = j | X_t = i)$$

la cual nos entrega la probabilidad de que la cadena se mueva desde el estado i al estado j en exactamente m -pasos.

■ **Ejemplo 4.1** Considere la siguiente cadena de Markov de dos estados:



donde el vector de probabilidades para el instante $t = 0$ es $\bar{p}(0) = (0.5, 0.5)$.

Las *probabilidades de transición* en dicha cadena están dadas por

$$P_{0,1} = 0.4, P_{0,0} = 0.6, P_{1,0} = 0.2, P_{1,1} = 0.8.$$

Por tanto, la *matriz de transición en un paso* es

$$\mathbf{P} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix}$$

Es fácil notar además que la suma de cada fila de la matriz anterior es igual a 1, es decir,

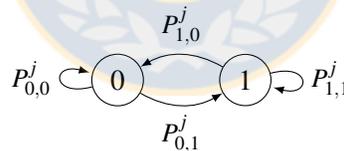
$$P_{0,0} + P_{0,1} = P_{1,0} + P_{1,1} = 1,$$

y las probabilidades del proceso en el instante $t = 1$ están dadas por el vector

$$\bar{p}(1) = \bar{p}(0)\mathbf{P} = (0.4, 0.6).$$

Las cadenas de Markov son importantes en esta memoria debido a que nos sirven para modelar la naturaleza de los fallos cometidos por cada experto, lo cual se formaliza en la siguiente definición.

Definición 4.2.2 — Cadena de Markov de fallos de un experto. Dado un experto $j \in \{1, \dots, N\}$, denotamos por $\mathbf{X}^j = \{X_t^j\}_t$ a la cadena de Markov de dos estados (fallo 1 y no fallo 0) del experto j



donde $P_{n,m}^j$, con $n, m \in \{0, 1\}$ denota las probabilidades de transición entre un estado n y otro m para el experto j .

Para nuestro propósito de estimar los fallos, nos interesa conocer la probabilidad de fallo de cada experto bajo un ambiente Markoviano. Esto lo vemos en el siguiente lema:

Lema 4.2.1 — Probabilidad de fallo de un experto bajo un ambiente Markoviano. Bajo un ambiente Markoviano, la probabilidad de que un experto $j \in \{1, \dots, N\}$ caiga en el estado de fallo en el instante t , está dada por la siguiente expresión:

$$\mathbb{P}(X_t^j = 1) = a_0^j (K_j)^t + \frac{((K_j)^t - 1)P_{0,1}^j}{K_j - 1},$$

donde, $K_j = P_{1,1}^j - P_{0,1}^j$ y a_0^j denota la probabilidad inicial de caer en el estado de fallo para el

experto j .

Demostración.

$$\begin{aligned}
 \mathbb{P}(X_t^j = 1) &= \mathbb{P}(X_t^j = 1 | X_{t-1}^j = 1) \cdot \mathbb{P}(X_{t-1}^j = 1) + \mathbb{P}(X_t^j = 1 | X_{t-1}^j = 0) \cdot \mathbb{P}(X_{t-1}^j = 0) \\
 &= P_{1,1}^j \cdot \mathbb{P}(X_{t-1}^j = 1) + P_{0,1}^j \cdot \mathbb{P}(X_{t-1}^j = 0) \\
 &= P_{1,1}^j \cdot \mathbb{P}(X_{t-1}^j = 1) + P_{0,1}^j \cdot (1 - \mathbb{P}(X_{t-1}^j = 1)) \\
 &= (P_{1,1}^j - P_{0,1}^j) \cdot \mathbb{P}(X_{t-1}^j = 1) + P_{0,1}^j \\
 &= K_j \cdot \mathbb{P}(X_{t-1}^j = 1) + P_{0,1}^j
 \end{aligned}$$

De lo anterior, obtenemos una ecuación para $a_t^j := \mathbb{P}(X_t^j = 1)$ de la forma

$$a_t^j = a_{t-1}^j K_j + P_{0,1}^j$$

de donde, razonando recursivamente, se obtiene que

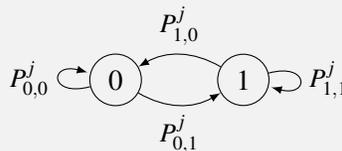
$$a_t^j = a_0^j (K_j)^t + \frac{((K_j)^t - 1) P_{0,1}^j}{K_j - 1}$$

■ **Ejemplo 4.2** Siguiendo con el Ejemplo 4.1, donde $K = P_{1,1} - P_{0,1} = 0.8 - 0.4 = 0.4$ y $a_0 = 0.5$, tenemos que, para $t = 2$

$$\begin{aligned}
 \mathbb{P}(X_t = 1) &= a_0^j (K_j)^t + \frac{((K_j)^t - 1) P_{0,1}^j}{K_j - 1} \\
 &= 0.5 \cdot 0.4 + \frac{((0.4)^2 - 1) 0.4}{0.4 - 1} \\
 &= 0.76
 \end{aligned}$$

■ **Definición 4.2.3 — Modelo markoviano de fallos.** Llamaremos *modelo markoviano de fallos* a aquel donde los fallos son determinados por una cadena de Markov de dos estados (como en la definición 4.2.2), uno de fallo y otro de no fallo. Para cada experto se tendrá una cadena de markov donde el estado 0 denotará el estado de no fallo, mientras que el estado 1 denotará lo contrario.

■ **Teorema 4.2.2** Bajo un ambiente markoviano, donde cada experto i sigue un modelo markoviano de fallos,



el valor esperado de fallos $\mathbb{E}[X]$ de cualquier predictor, satisface

$$n \min_t \min_j \left\{ a_0^j + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \leq \mathbb{E}[X] \leq n \max_t \max_j \left\{ a_0^j + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\}$$

donde a_0^j es la probabilidad de fallo inicial del experto j , n es la cantidad de instantes de tiempo total y $K_j = P_{1,1}^j - P_{0,1}^j$.

Demostración. Sea $\mathbf{X}^j = \{X_t^j\}_t$ la cadena de Markov que define los estados del experto j . De manera análoga a la técnica usada en el Teorema 4.1.1, calculamos el valor esperado de fallos en el instante t para el predictor,

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_j \mathbb{P}(I_t = j) \cdot \mathbb{P}(j \text{ falla en } t) \\ &= \sum_j \mathbb{P}(I_t = j) \cdot \mathbb{P}(X_t^j = 1) \end{aligned}$$

donde, del lema 4.2.1

$$\mathbb{P}(X_t^j = 1) = a_0^j (K_j)^t + \frac{((K_j)^t - 1) P_{0,1}^j}{K_j - 1}$$

En consecuencia,

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_j \mathbb{P}(I_t = j) \cdot \left(a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right) \\ &\leq \max_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \sum_j \mathbb{P}(I_t = j) \stackrel{=1}{=} \\ &= \max_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \end{aligned}$$

y dado que

$$E[X] = \sum_{t=1}^n \mathbb{E}[X_t],$$

entonces,

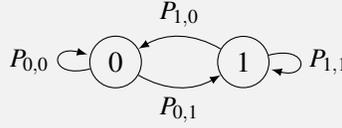
$$\begin{aligned} \mathbb{E}[X] &\leq \sum_{t=1}^n \max_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \\ &\leq \sum_{t=1}^n \max_t \max_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \\ &= n \max_t \max_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\} \end{aligned}$$

de manera análoga se concluye que

$$\mathbb{E}[X] \geq n \min_t \min_j \left\{ a_0^j (K_j)^t + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\}$$

con lo cual se demuestra el teorema. ■

Teorema 4.2.3 Bajo un ambiente markoviano, donde todos los expertos siguen el mismo modelo markoviano de fallos,



el valor esperado de fallos $\mathbb{E}[X]$ de cualquier predictor es igual a

$$\mathbb{E}[X] = \left(a_0 + \frac{P_{0,1}}{K-1} \right) \left(\frac{K^{n+1} - 1}{K-1} \right) - \frac{P_{0,1}}{K-1} n$$

donde a_0 es la probabilidad de fallo inicial, n la cantidad de instantes de tiempo total, $K = P_{1,1} - P_{0,1}$ y $|K| < 1$.

Demostración. Siguiendo la demostración del teorema 4.2.2, tenemos que

$$\mathbb{E}[X_t] = a_0 K^t + \frac{K^t - 1}{K-1} P_{0,1}$$

y por tanto,

$$\begin{aligned} \mathbb{E}[X] &= \sum_{t=1}^n \left(a_0 K^t + \frac{K^t - 1}{K-1} P_{0,1} \right) \\ &= \left(a_0 + \frac{P_{0,1}}{K-1} \right) \sum_{t=1}^n K^t - \frac{P_{0,1}}{K-1} n \end{aligned}$$

■ **Ejemplo 4.3** Supongamos que tenemos N expertos cuyos fallos están modelados por la cadena de Markov del Ejemplo 4.1, donde $a_0 = 0.5$, $P_{0,1} = 0.4$ y $K = 0.4$. Entonces, el valor esperado de fallos de cualquier predictor en $n = 10$ instantes de tiempo será:

$$\begin{aligned} \mathbb{E}[X] &= \left(a_0 + \frac{P_{0,1}}{K-1} \right) \left(\frac{K^{n+1} - 1}{K-1} \right) - \frac{P_{0,1}}{K-1} n \\ &= \left(0.5 + \frac{0.4}{0.4-1} \right) \left(\frac{0.4^{11} - 1}{0.4-1} \right) - \frac{0.4}{0.4-1} 11 \\ &= 6.34 \end{aligned}$$

Este resultado cobra sentido, pues en la cadena de Markov del Ejemplo 4.1 se aprecia que la probabilidad de salir del estado de fallo no es muy alta. Puede revisar el Ejemplo 4.2 donde la probabilidad de caer en el estado de fallo en el instante $t = 2$ fue de 0.76. ■

4.2.1 Modelo probabilista de fallos como un caso particular del modelo markoviano

Es interesante notar que el modelo markoviano de fallos generaliza al modelo probabilista, en el sentido de que, escogiendo valores adecuados de las probabilidades de transición en las cadenas de Markov que definen el comportamiento de los fallos de cada experto, podemos encontrar tasas de fallo para cada uno. Este resultado queda formalizado en la siguiente proposición.

Proposición 4.2.4 Un *modelo markoviano de fallos*, en el que para cada experto se cumple que $P_{0,0}^j - P_{1,0}^j = 0$, es equivalente a un *modelo probabilista de fallos* de tasa $P_{1,1}^j$

Demostración. Dado un experto i y X_t su estado en el tiempo t , tenemos que

$$\begin{aligned}\mathbb{P}(X_t^j = 0) &= \mathbb{P}(X_t^j = 0 | X_{t-1}^j = 0) \cdot \mathbb{P}(X_{t-1}^j = 0) + \mathbb{P}(X_t^j = 0 | X_{t-1}^j = 1) \cdot \mathbb{P}(X_{t-1}^j = 1) \\ &= P_{0,0}^j \cdot \mathbb{P}(X_{t-1}^j = 0) + P_{1,0}^j \cdot \mathbb{P}(X_{t-1}^j = 1) \\ &= P_{0,0}^j \cdot \mathbb{P}(X_{t-1}^j = 0) + P_{1,0}^j \cdot (1 - \mathbb{P}(X_{t-1}^j = 1)) \\ &= (P_{0,0}^j - P_{1,0}^j) \cdot \mathbb{P}(X_{t-1}^j = 0) + P_{1,0}^j \\ &= P_{1,0}^j\end{aligned}$$

lo cual implica que $\mathbb{P}(X_t^j = 1) = P_{1,1}^j$ ■

Nota Respecto a la hipótesis de la proposición 4.2.4 tenemos

$$P_{0,0}^j - P_{1,0}^j = 0 \Leftrightarrow P_{1,1}^j - P_{0,1}^j = 0 \Leftrightarrow P_{1,1}^j + P_{0,0}^j = 1$$

En efecto,

$$\begin{aligned}P_{0,0}^j - P_{1,0}^j = 0 &\Leftrightarrow (1 - P_{0,1}^j) - (1 - P_{1,1}^j) = 0 \\ &\Leftrightarrow P_{1,1}^j - P_{0,1}^j = 0 \\ &\Leftrightarrow P_{1,1}^j - (1 - P_{0,0}^j) = 0 \\ &\Leftrightarrow P_{1,1}^j + P_{0,0}^j = 1\end{aligned}$$

Para hacer notar la validez y concordancia entre los resultados del modelo probabilista y el markoviano, analizaremos un caso particular, usando la proposición 4.2.4 y el teorema 4.2.3.

Nota La proposición 4.2.4 nos indica que hay una relación entre el modelo probabilista y el markoviano. Tal relación debe verse reflejada en los resultados que se tienen para el ambiente markoviano, obteniendo a partir de estos, resultados para el ambiente probabilista. En efecto, bajo los supuestos del teorema 4.2.3, si $K = 0$ obtenemos que

$$\begin{aligned}\mathbb{E}[X] &= a_0 + \frac{P_{0,1}}{-1} - \frac{P_{0,1}}{-1}n \\ &= a_0 + P_{0,1}(n - 1) \\ &= a_0 + P_{1,1}(n - 1), \text{ por la nota anterior}\end{aligned}$$

por otro lado, de la proposición 4.2.4 tenemos que $a_0 = P_{1,1}$ es la tasa de fallo, con lo cual

$$\mathbb{E}[X] = n \cdot P_{0,1}$$

lo cual concuerda con el corolario 4.1.2 del ambiente probabilista.

4.3 Ambiente Adversarial

Ya hemos visto dos ambientes que modelan los fallos de los expertos: probabilista y markoviano. Note que en ambos se asume la existencia de una medida de probabilidad para los fallos, esto nos permite tener cierta información sobre la aparición de los fallos para cada experto. El ambiente adversarial es un caso opuesto, donde la aparición de fallos es totalmente arbitraria y definida por un adversario (de allí el nombre). Este *adversario* juega el rol de posicionar los fallos de cada experto en los instantes de tiempo que estime conveniente para su propio beneficio, que para nuestro caso será el de hacernos fallar lo máximo que pueda.

Definición 4.3.1 — Modelo adversarial de fallos. Llamaremos *modelo adversarial de fallos* a aquel donde los fallos son determinados por un adversario, el cual no se rige por una medida de probabilidad y las apariciones de los fallos son totalmente arbitrarias.

Los tres ambientes ya definidos serán los modelos de fallo que utilizaremos en esta memoria. Para el modelo probabilista y markoviano, mostraremos resultados de predictores en el capítulo siguiente, y luego, en los experimentos, utilizaremos los tres ambientes con el fin de comparar varios predictores que serán definidos en el próximo capítulo.



5. Predictores

El objetivo principal de esta memoria es diseñar y analizar algoritmos para encaminamiento en redes de comunicación con fallos. Hay muchos predictores que han sido utilizados en la literatura y que nos dan un piso y meta para proponer nuevos algoritmos que nos ayuden a resolver nuestra nueva problemática; evitar la mayor cantidad de fallos posibles en nuestras decisiones. Es lógico pensar que como parámetro básico de comparación está la elección al azar. Esta metodología de decisión está adoptada por el predictor uniforme, quien basa sus elecciones en escoger con la misma probabilidad a cada experto.

Este capítulo comprende el estudio y análisis de predictores que han sido estudiados en la literatura, entre los que encontramos: predictor uniforme (PUNIF) y follow the perturbed leader (FTPL, ver Kalai y Vempala [12] y [4, pg. 74]).

Por otra parte, proponemos un nuevo predictor que extiende a FTPL, ya que considera, además de las pérdidas acumuladas, la información que proporciona el ambiente respecto a la ocurrencia de fallos en el sistema. A este predictor le llamamos modified follow the perturbed leader (MFTPL) y del cual realizaremos el análisis respectivo para acotar el valor esperado de fallos en los distintos ambientes.

5.1 Preliminares

Algunos resultados que nos servirán para entender de mejor manera las cotas propuestas para cada predictor serán presentados en esta sección.

Lema 5.1.1 Para todo $S \subseteq \{1, \dots, N\}$ y $\mathbf{Z} = (Z_1, \dots, Z_{|S|})$ vector de v.a. continuas i.i.d., se tiene la siguiente identidad

$$\sum_{i \in S} \mathbb{P}(L_i + Z_i \leq L_j + Z_j (\forall j \in S \setminus \{i\})) = 1$$

donde $(L_1, \dots, L_{|S|}) \in \mathbb{R}^{|S|}$.

Demostración. Por inducción sobre la cardinalidad de S .

Para el caso en que $S = \{j, k\}$, tenemos que

$$\begin{aligned} \sum_{i \in S} \mathbb{P}(L_i + Z_i \leq L_j + Z_j (\forall j \in S \setminus \{i\})) &= \mathbb{P}(L_j + Z_j \leq L_k + Z_k) + \mathbb{P}(L_k + Z_k \leq L_j + Z_j) \\ &= 1 \end{aligned}$$

con lo cual se cumple la identidad. Notemos además que el conjunto $\{L_j + Z_j = L_k + Z_k\}$ tiene medida igual a 0 debido a la continuidad de las variables aleatorias.

Supongamos ahora que el resultado es válido para $|S| = n < N$. Consideremos \hat{S} tal que $|\hat{S}| = n + 1$, en tal caso, escribimos $\hat{S} = \{i_1, \dots, i_n, i_{n+1}\} := S \cup \{i_{n+1}\} \subseteq \{1, \dots, N\}$,

$$\begin{aligned} &\sum_{i \in \hat{S}} \mathbb{P}(L_i + Z_i \leq L_j + Z_j (\forall j \in \hat{S} \setminus \{i\})) \\ &= \mathbb{P}(L_{i_{n+1}} + Z_{i_{n+1}} \leq L_j + Z_j (\forall j \in \hat{S} \setminus \{i_{n+1}\})) + \sum_{i \in S} \mathbb{P}(L_i + Z_i \leq L_j + Z_j (\forall j \in \hat{S} \setminus \{i\})) \\ &= \mathbb{P}(L_{i_{n+1}} + Z_{i_{n+1}} \leq L_j + Z_j (\forall j \in \{i_1, \dots, i_n\})) + \sum_{k=1}^n \mathbb{P}(L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in \hat{S} \setminus \{i_k\})) \\ &= \mathbb{P}(L_{i_{n+1}} + Z_{i_{n+1}} \leq L_j + Z_j (\forall j \in \{i_1, \dots, i_n\})) \\ &\quad + \sum_{k=1}^n \mathbb{P}(L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\}) \wedge L_{i_k} + Z_{i_k} \leq L_{i_{n+1}} + Z_{i_{n+1}}) \\ &= \sum_{k=1}^n \mathbb{P}(L_{i_{n+1}} + Z_{i_{n+1}} \leq L_{i_k} + Z_{i_k} \wedge L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\})) \\ &\quad + \sum_{k=1}^n \mathbb{P}(L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\}) \wedge L_{i_k} + Z_{i_k} \leq L_{i_{n+1}} + Z_{i_{n+1}}) \\ &= \sum_{k=1}^n \left(\underbrace{\mathbb{P}(L_{i_{n+1}} + Z_{i_{n+1}} \leq L_{i_k} + Z_{i_k})}_{:=A} \underbrace{\mathbb{P}(L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\}))}_{:=B} \right. \\ &\quad \left. + \mathbb{P}(\underbrace{L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\})}_{:=B} \wedge \underbrace{L_{i_k} + Z_{i_k} \leq L_{i_{n+1}} + Z_{i_{n+1}}}_{:=A^c}) \right) \\ &= \sum_{k=1}^n \mathbb{P}(L_{i_k} + Z_{i_k} \leq L_j + Z_j (\forall j \in S \setminus \{i_k\})) \\ &= 1, \text{ por hipótesis de inducción} \end{aligned}$$

■

Este lema nos permitirá encontrar cotas para el valor esperado de fallos en los predictores estudiados.

También tenemos un resultado general que aplica para cualquier predictor sobre los ambientes probabilista y markoviano. Este resultado lo veremos en la siguiente proposición.

Proposición 5.1.2 Bajo los ambientes probabilista y markoviano, con N expertos, el valor esperado de fallos de **cualquier predictor** en un instante t , $\mathbb{E}[X_t]$, está acotado por

$$\mathbb{E}[X_t] \leq \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N \min_{i \in S} \mathbb{P}(i \text{ falla en } t) \right) \max_{i \in S \setminus \{i^*\}} \mathbb{P}(I_t = i) + \min_{i \in S} \mathbb{P}(i \text{ falla en } t) \quad (5.1)$$

$$\mathbb{E}[X_t] \leq \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N \max_{i \in S} \mathbb{P}(i \text{ falla en } t) \right) \min_{i \in S \setminus \{k^*\}} \mathbb{P}(I_t = i) + \max_{i \in S} \mathbb{P}(i \text{ falla en } t) \quad (5.2)$$

donde $S = \{1, \dots, N\}$, $i^* = \operatorname{argmin}_{i \in S} \mathbb{P}(i \text{ falla en } t)$ y $k^* = \operatorname{argmax}_{i \in S} \mathbb{P}(i \text{ falla en } t)$.

En esta proposición se entregan dos cotas muy parecidas pero con un sutil detalle que podremos ver en la demostración.

Demostración. Siguiendo con la técnica usada en el teorema 4.1.1, notemos que

$$\mathbb{E}[X_t] = \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \mathbb{P}(I_t = i)$$

además,

$$\mathbb{P}(I_t = i^*) = 1 - \sum_{\substack{i=1 \\ i \neq i^*}}^N \mathbb{P}(I_t = i)$$

en consecuencia,

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_{\substack{i=1 \\ i \neq i^*}}^N \mathbb{P}(i \text{ falla en } t) \cdot \mathbb{P}(I_t = i) + \mathbb{P}(i^* \text{ falla en } t) \cdot \left[1 - \sum_{\substack{i=1 \\ i \neq i^*}}^N \mathbb{P}(I_t = i) \right] \\ &= \sum_{\substack{i=1 \\ i \neq i^*}}^N \underbrace{\left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(i^* \text{ falla en } t) \right)}_{\geq 0} \cdot \mathbb{P}(I_t = i) + \mathbb{P}(i^* \text{ falla en } t) \\ &\leq \max_{i \in S \setminus \{i^*\}} \mathbb{P}(I_t = i) \sum_{\substack{i=1 \\ i \neq i^*}}^N \left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(i^* \text{ falla en } t) \right) + \mathbb{P}(i^* \text{ falla en } t) \\ &= \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N \mathbb{P}(i^* \text{ falla en } t) \right) \max_{i \in S \setminus \{i^*\}} \mathbb{P}(I_t = i) + \mathbb{P}(i^* \text{ falla en } t) \end{aligned}$$

con lo cual se demuestra la cota 5.1.

De manera análoga, usando k^* , tenemos que

$$\begin{aligned}
\mathbb{E}[X_t] &= \sum_{\substack{i=1 \\ i \neq k^*}}^N \underbrace{\left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(k^* \text{ falla en } t) \right)}_{\leq 0} \cdot \mathbb{P}(I_t = i) + \mathbb{P}(k^* \text{ falla en } t) \\
&\leq \min_{i \in S \setminus \{k^*\}} \mathbb{P}(I_t = i) \sum_{\substack{i=1 \\ i \neq k^*}}^N \left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(k^* \text{ falla en } t) \right) + \mathbb{P}(k^* \text{ falla en } t) \\
&= \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N\mathbb{P}(k^* \text{ falla en } t) \right) \max_{i \in S \setminus \{k^*\}} \mathbb{P}(I_t = i) + \mathbb{P}(k^* \text{ falla en } t)
\end{aligned}$$

con lo cual se demuestra la cota 5.2. ■

Nota Para la Proposición 5.1.2, recordamos que:

- bajo un ambiente probabilista, $\mathbb{P}(i \text{ falla en } t) = p_{i,t}$,
- y bajo un ambiente markoviano, $\mathbb{P}(i \text{ falla en } t) = \mathbb{P}(X_t^i = 1)$ (ver definición 4.2.1)

5.2 Predictor Uniforme (PUNIF)

El primer predictor que presentaremos es el predictor uniforme, al que denotaremos PUNIF. Este predictor entrega una performance que nos servirá como base para comparar con los demás predictores que utilizan mayor información que él en sus elecciones.

Definición 5.2.1 — PUNIF. Dados $\{1, \dots, N\}$ expertos, la elección del predictor PUNIF en el instante t , está dada por

$$I_t = i, \text{ con probabilidad } \frac{1}{N}$$

Proposición 5.2.1 Bajo los ambientes probabilista y markoviano, con N expertos, el valor esperado de fallos del predictor PUNIF en un instante t , $\mathbb{E}[X_t]$ está dado exactamente por

$$\mathbb{E}[X_t] = \frac{\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t)}{N}$$

Demostración. Siguiendo la técnica usada en el teorema 4.1.1,

$$\begin{aligned}
\mathbb{E}[X_t] &= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot P(I_t = i) \\
&= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \frac{1}{N} \\
&= \frac{1}{N} \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t)
\end{aligned}$$

demostrando la proposición. ■



- Para el caso del **ambiente probabilista** (ver definición 4.1.1),

$$\mathbb{P}(i \text{ falla en } t) = p_{i,t}$$

- Para el caso del **ambiente markoviano** (ver definición 4.2.3), por lema 4.2.1

$$\mathbb{P}(i \text{ falla en } t) = a_0^i (K_i)^t + \frac{((K_i)^t - 1)P_{0,1}^i}{K_i - 1},$$

con $K_i = P_{1,1}^i - P_{0,0}^i$ y a_0^i la probabilidad inicial de que i caiga en el estado de fallo.

- **Ejemplo 5.1** Si tenemos $N = 5$ expertos, con tasas de fallo dadas en el Cuadro 5.1

	E_1	E_2	E_3	E_4	E_5
p_i	0.10	0.20	0.25	0.50	1.00

Cuadro 5.1: Tasas de fallo para un ambiente probabilista.

entonces el valor esperado de fallos en un instante de tiempo t es

$$\mathbb{E}[X_t] = \frac{0.10 + 0.20 + 0.25 + 0.50 + 1.00}{5} = 0.41$$

además, considerando que las tasas de fallo son fijas en el tiempo, el valor esperado de fallos hasta un instante $n = 20$ está dado por

$$\mathbb{E}[X] = \sum_{t=1}^{20} \mathbb{E}[X_t] = 20\mathbb{E}[X_t] = 20 \cdot 0.41 = 8.1$$

Notemos que lo anterior fue posible gracias a la independencia respecto al tiempo de las tasas de fallo. ■

- **Ejemplo 5.2** En el caso en que sólo conocemos los costos de envío y fallos de los expertos, como se muestra en el Cuadro 5.2,

t	E_1	E_2	E_3	E_4	E_5
1	1.00	1.00	0.20	0.60	1.00
2	0.50	1.00	0.60	1.00	1.00
3	0.10	0.60	0.70	0.40	1.00
4	0.20	0.90	0.10	1.00	1.00
5	0.40	0.50	1.00	0.00	1.00
6	1.00	0.60	0.40	0.10	1.00
7	0.80	0.80	1.00	1.00	1.00
8	0.60	1.00	1.00	0.20	1.00
9	0.90	0.50	0.20	1.00	1.00
10	0.30	1.00	0.20	0.50	1.00

Cuadro 5.2: Ejemplo de costos arbitrarios y fallos en un ambiente probabilista.

podemos estimar las tasas de fallo tomando un enfoque frecuentista, vale decir, $\hat{p}_i = \frac{\text{cant de fallos de } i}{n}$, obteniendo las tasas que se muestran en el Cuadro 5.3:

	E_1	E_2	E_3	E_4	E_5
\hat{p}_i	0.20	0.40	0.30	0.40	1.00

Cuadro 5.3: Tasas de fallo estimadas para un ambiente probabilista.

puediendo así estimar el valor esperado de fallos con las tasas anteriormente expuestas,

$$\bar{X}_t = \frac{0.20 + 0.40 + 0.30 + 0.40 + 1.00}{5} = 0.46$$

Es bueno recalcar que la tabla anterior fue generada con las tasas de fallo del ejemplo 5.1, y debido a que la muestra es pequeña, nuestras estimaciones no son del todo cercanas a las tasas reales, esto debiera compensarse para n grande.

5.3 Follow the perturbed leader (FTPL)

Dentro del modelo de *Aprendizaje y Predicción con Ayuda de Expertos* se presenta un predictor en particular, llamado *Follow the perturbed leader* (ver Kalai y Vempala [12] y [4, pg. 74]), el cual tiene por objetivo seguir al mejor experto sujeto a una perturbación aleatoria. Tal perturbación fue pensada para desenvolverse en un ambiente adversarial, cuyo objetivo es confundir al adversario.

Definición 5.3.1 — FTPL. Dados $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n$ N -vectores aleatorios independientes e idénticamente distribuidos (i.i.d) absolutamente continuos con respecto a la medida de Lebesgue, con componentes $Z_{i,t}, i = 1, \dots, N$. La elección del predictor FTPL está dada por

$$I_t = \operatorname{argmin}_i \{L_{i,t-1} + Z_{i,t}\}$$

Se puede apreciar que FTPL no utiliza la información de los fallos que ocurren para cada experto, sólo se encarga de perturbar la pérdida acumulada.

5.3.1 Probabilidad de elección de un experto

Recordamos que para el cálculo del valor esperado de fallos, específicamente en 4.1.1, utilizamos una técnica que involucra conocer la probabilidad de elección de un experto en un instante de tiempo determinado. Esta técnica seguirá siendo utilizada para analizar qué ocurre con los predictores en estudio.

Dado un experto $i \in \{1, \dots, N\}$, notamos que la probabilidad de escoger tal experto está sujeta a que

$$L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \neq i)$$

y en consecuencia,

$$\mathbb{P}(I_t = i) = \mathbb{P}\left(L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \neq i)\right)$$

Cotas para el valor esperado de fallos de este predictor se pueden obtener directamente de la Proposición 5.1.2.

Nota La Proposición 5.1.2 aplica directamente al **predicador FTPL**, donde

$$\mathbb{P}(I_t = i) = \mathbb{P}\left(L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \neq i)\right)$$

Las cotas de la proposición 5.1.2 son ajustadas. Para ver esto, evidenciaremos dos casos en los cuales las cotas se alcanzan. El primero y más general, es cuando las probabilidades de fallo de cada experto coinciden. El segundo caso corresponde a aquel donde las probabilidades de elección de cada experto coinciden. Hay predicadores para los cuales esto se tiene para todo instante t , como es el caso del predicador PUNIF, mientras que otros, como FTPL, que necesitan condiciones sobre las pérdidas acumuladas de los expertos.

- **Caso 1** Las probabilidades de fallo de cada experto coinciden, es decir, $\mathbb{P}(i \text{ falla en } t) = c, \forall i \in S = \{1, \dots, N\}$

En tal caso,

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \mathbb{P}(I_t = i) \\ \mathbb{E}[X_t] &= \sum_{i=1}^N c \cdot \mathbb{P}(I_t = i) \\ &= c \underbrace{\sum_{i=1}^N \mathbb{P}(I_t = i)}_{=1} \\ &= c \end{aligned}$$

Mientras que

$$(5.1) \Rightarrow \mathbb{E}[X_t] \leq \min_{i \in S} \mathbb{P}(i \text{ falla en } t) = c$$

$$(5.2) \Rightarrow \mathbb{E}[X_t] \leq \max_{i \in S} \mathbb{P}(i \text{ falla en } t) = c$$

Más aún, en este caso, el valor esperado de fallos total $\mathbb{E}[X] = \sum_{t=1}^n \mathbb{E}[X_t]$ es ajustado, pues el análisis anterior es válido independiente del instante t , cosa que no siempre ocurre en el caso que veremos a continuación.

- **Caso 2** Cuando la probabilidad de que el predicador elija un experto es la misma, es decir, $\mathbb{P}(I_t = i) = \alpha, \forall i \in S$, tenemos que

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \mathbb{P}(I_t = i) \\ &= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \alpha \\ &= \alpha \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \end{aligned}$$

es bueno notar que en realidad el valor de α es $\frac{1}{N}$, lo cual nos indica que en tales casos, el predicador (cualquiera sea), se comporta como el predicador PUNIF en ese paso. Notemos lo que ocurre con las cotas de la proposición 5.1.2

De 5.1, tenemos

$$\begin{aligned}\mathbb{E}[X_t] &\leq \alpha \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - \alpha N \min_{i \in S} \mathbb{P}(i \text{ falla en } t) + \min_{i \in S} \mathbb{P}(i \text{ falla en } t) \\ &= \alpha \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - \min_{i \in S} \mathbb{P}(i \text{ falla en } t) + \min_{i \in S} \mathbb{P}(i \text{ falla en } t) \\ &= \alpha \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t)\end{aligned}$$

Para 5.2 el análisis es el mismo.

En el análisis anterior no se utilizó un predictor en particular. Aunque sabemos que el predictor PUNIF cumple con esta condición, también FTPL lo hace, cuando las pérdidas acumuladas de cada experto i coinciden en un instante t , vale decir, $L_{i,t-1} = L_{j,t-1}$, ($\forall i, j \in \{1, \dots, N\}$), y en tal caso

$$\begin{aligned}\mathbb{P}(I_t = i) &= \mathbb{P}(L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \neq i)) \\ &= \mathbb{P}(Z_{i,t} \leq Z_{j,t} (\forall j \neq i))\end{aligned}$$

y como las variables $Z_{i,t}$ son idénticamente distribuidas, tales probabilidades de elección coinciden $\forall i$ y son iguales a $\frac{1}{N}$.

Nota Para el caso de **FTPL** bajo un ambiente probabilista donde las tasas de fallo son fijas para cada experto (independientes del tiempo), usando la cota 5.1, podemos eliminar la contribución de la elección de los expertos acotando $\mathbb{P}(I_t = i)$ por 1, en cuyo caso obtenemos que

$$\mathbb{E}[X_t] \leq \sum_{i=1}^N p_i - (N-1) \min_i p_i$$

lo cual nos entrega una cota que no siempre aporta información, basta ver el caso $N = 4$ con $(p_1, p_2, p_3, p_4) = (0.9, 0.1, 0.1, 0.9)$ y se tiene que

$$\mathbb{E}[X_t] \leq 2 - (4-1) \cdot 0.1 = 1.7$$

y sabemos que el valor máximo para $\mathbb{E}[X_t]$ es 1.

Por otro lado, realizando el mismo análisis anterior en la ecuación 5.2, pero esta vez acotando la probabilidad inferiormente por 0 (debido al signo de la diferencia que lo acompaña), concluimos que

$$\mathbb{E}[X_t] \leq \max_i p_i$$

mismo resultado que se obtuvo en el teorema 4.1.1.

Nota Cuando las pérdidas acumuladas de cada experto coinciden en un instante t , el predictor FTPL se comporta como el predictor PUNIF en ese paso.

■ **Ejemplo 5.3 — Peor caso para FTPL: La escalera.** Hemos encontrado un ejemplo particular en el que FTPL comete muchos fallos. El nombre de *escalera* viene de la configuración de los fallos en la tabla de costos (que puede ver en el Cuadro 5.3), podemos observar que se repite la misma configuración diagonal de fallos desde el instante $t = 2$ hasta el final.

En el Cuadro 5.4 se muestra la tabla de costos para este ejemplo, la cual contiene 3 expertos y 16 instantes de tiempo a considerar. En el Cuadro 5.5, puede apreciar la tabla con las pérdidas acumuladas de cada experto en este ejemplo.

t	E_1	E_2	E_3
1	0.1	0.5	0.8
2	1.0	0.0	0.0
3	0.0	1.0	0.0
4	0.0	0.0	1.0
5	1.0	0.0	0.0
6	0.0	1.0	0.0
7	0.0	0.0	1.0
8	1.0	0.0	0.0
9	0.0	1.0	0.0
10	0.0	0.0	1.0
11	1.0	0.0	0.0
12	0.0	1.0	0.0
13	0.0	0.0	1.0
14	1.0	0.0	0.0
15	0.0	1.0	0.0
16	0.0	0.0	1.0



t	$L_{E1,t}$	$L_{E2,t}$	$L_{E3,t}$
1	0.10	0.50	0.80
2	1.10	0.50	0.80
3	1.10	1.50	0.80
4	1.10	1.50	1.80
5	2.10	1.50	1.80
6	2.10	2.50	1.80
7	2.10	2.50	2.80
8	3.10	2.50	2.80
9	3.10	3.50	2.80
10	3.10	3.50	3.80
11	4.10	3.50	3.80
12	4.10	4.50	3.80
13	4.10	4.50	4.80
14	5.10	4.50	4.80
15	5.10	5.50	4.80
16	5.10	5.50	5.80

Cuadro 5.4: Costos de cada experto para el ejemplo 5.3.

Cuadro 5.5: Pérdidas acumuladas de cada experto para el ejemplo 5.3.

Recordamos al lector que la pérdida del experto i en el instante t es igual a $f_{i,t} - y_t$ en caso de no haber fallo, y 1 en caso contrario. Para efectos prácticos de estudiar los fallos es lo mismo obviar el aporte de y_t en la pérdida, dado que esto conlleva simplemente una traslación en el valor de la pérdida instantánea, y por ende no altera la elección del predictor.

La elección de FTPL corresponde al $\operatorname{argmin}_i \{L_{i,t-1} + Z_{i,t}\}$, y cuando $t = 1$, $L_{i,t-1}$ se asume como 0, por lo que la elección de FTPL queda en manos del valor que la variable aleatoria tome en el instante $t = 1$, en este caso, tal perturbación será tomada como una uniforme en el intervalo $[0, \Delta]$, con $\Delta = 0.1$.

Una vez que ejecutamos nuestro predictor sobre este escenario, bajo las perturbaciones mostradas en el Cuadro 5.6

t	E_1	E_2	E_3
1	0.01	0.05	0.06
2	0.01	0.03	0.07
3	0.06	0.03	0.09
4	0.07	0.09	0.03
5	0.10	0.02	0.08
6	0.06	0.07	0.10
7	0.01	0.09	0.09
8	0.09	0.03	0.08
9	0.01	0.03	0.08
10	0.08	0.09	0.01
11	0.09	0.09	0.01
12	0.07	0.03	0.05
13	0.02	0.04	0.08
14	0.08	0.04	0.06
15	0.06	0.02	0.01
16	0.02	0.05	0.01

Cuadro 5.6: Perturbación aleatoria $Unif[0, 0.1]$.

obtenemos, para cada instante t , el *Mejor Experto* y su pérdida acumulada, la elección del predictor FTPL, su pérdida acumulada y arrepentimiento acumulado, y un indicador de fallo. Esta información se muestra en el Cuadro 5.7:

t	Mejor Experto	$L_{i,t}$	Mejor Experto	I_t	L_t	R_t	Falla I_t
1	E1		0.10	E1	0.10	0.00	No
2	E2		0.50	E1	1.10	0.60	Si
3	E3		0.80	E2	2.10	1.30	Si
4	E1		1.10	E3	3.10	2.00	Si
5	E2		1.50	E1	4.10	2.60	Si
6	E3		1.80	E2	5.10	3.30	Si
7	E1		2.10	E3	6.10	4.00	Si
8	E2		2.50	E1	7.10	4.60	Si
9	E3		2.80	E2	8.10	5.30	Si
10	E1		3.10	E3	9.10	6.00	Si
11	E2		3.50	E1	10.10	6.60	Si
12	E3		3.80	E2	11.10	7.30	Si
13	E1		4.10	E3	12.10	8.00	Si
14	E2		4.50	E1	13.10	8.60	Si
15	E3		4.80	E2	14.10	9.30	Si
16	E1		5.10	E3	15.10	10.00	Si

Cuadro 5.7: Tabla resumen de la ejecución de FTPL.

Pérdida Acumulada

Ejemplo escalera

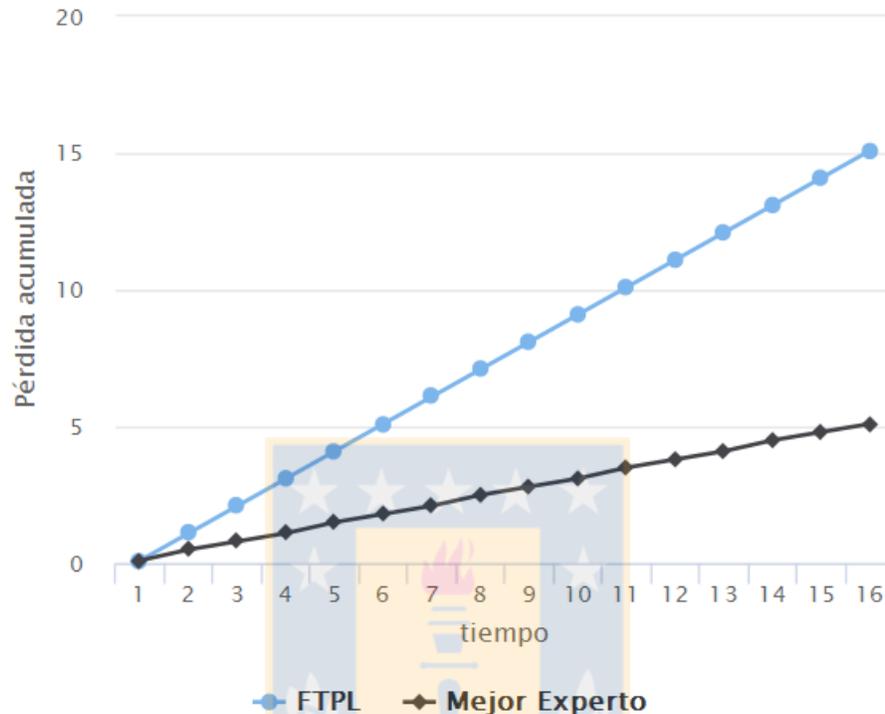


Figura 5.1: Pérdida acumulada de FTPL y el mejor experto para ejemplo de la escalera

Podemos observar que el predictor FTPL escoge absolutamente todos los fallos que se presentan. Esto motiva a preguntarse el por qué de este fenómeno y cómo remediarlo. Pues bien, podemos notar que cada camino, luego de fallar, no falla en dos instantes consecutivos y luego de eso, vuelve a fallar. Este comportamiento sigue claramente un patrón de fallos, pero FTPL no está construido para usar esta información. Esto nos ha motivado a definir un nuevo predictor, basado en FTPL, pero que utilice la información de los fallos en cada camino, el cual presentamos y analizamos a continuación.

5.4 Follow the perturbed leader modificado (MFTPL)

La cantidad de veces que falla cada experto es una variable importante a la hora de tomar decisiones sobre si es conveniente elegirlo o no. Lo anterior claramente no lo sabemos con exactitud en la realidad debido a la aleatoriedad del ambiente en el que estamos inmersos. Sin embargo, cada ambiente tiene asociado un modelo que nos permite describir ese comportamiento. El tener una idea sobre qué predictores fallarán con alta probabilidad nos invita a no considerarlos en nuestra elección. En base a lo anterior, surge el predictor modificado de FTPL, al cual denominamos MFTPL.

Definición 5.4.1 — MFTPL Probabilista. Sean $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n$ N -vectores aleatorios independientes e idénticamente distribuidos (i.i.d) y $\mathbf{M}_1, \mathbf{M}_2, \dots \subset \{0, 1\}^N$ también i.i.d. Ambas distribuciones absolutamente continuas con respecto a la medida de Lebesgue, tal que

$$\mathbb{P}(M_{i,t} = 1) = q_{i,t},$$

donde $q_{i,t} \in [0, 1]$ representa la tasa de fallo que el predictor MFTPL cree que tendrá el experto i en el tiempo t .

La elección del predictor MFTPL está dada por

$$I_t = \operatorname{argmin}_{i: M_{i,t}=0} \left\{ L_{i,t-1} + Z_{i,t} \right\}$$

Nota Note que el predictor anterior necesita creer que al menos un experto no falla en un instante t para efectuar su elección. En caso contrario, lanzará nuevamente las variables aleatorias $M_{i,t}$ para obtener un conjunto no vacío de expertos sobre el cual realizar su elección.

Definición 5.4.2 — MFTPL Markoviano. Sean $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n$ N -vectores aleatorios independientes e idénticamente distribuidos (i.i.d) y $\mathbf{M}_1, \mathbf{M}_2, \dots \subset \{0, 1\}^N$ también i.i.d. Ambas distribuciones absolutamente continuas con respecto a la medida de Lebesgue, tal que

$$M_{i,t} = 1 \Leftrightarrow X_t^i = 1,$$

donde, X_t^j denota el estado de la cadena de Markov de fallos, $\{\mathbf{X}^i\}_t$, que el predictor MFTPL cree que tendrá el experto i en el instante t .

La elección del predictor MFTPL está dada por

$$I_t = \operatorname{argmin}_{i: M_{i,t}=0} \left\{ L_{i,t-1} + Z_{i,t} \right\}$$

Nota En las definiciones anteriores, el predictor entiende a $\mathbb{P}(M_{i,t} = 1)$ como la probabilidad que tiene el experto i en el instante t de fallar. Tal probabilidad es definida por el predictor y no necesariamente coincide con el modelo de fallos intrínseco de cada experto.

Para el caso de MFTPL markoviano, la expresión que define dicha probabilidad está dada por el lema 4.2.1.

■ **Ejemplo 5.4 — Peor caso para FTPL: la escalera, contrastando con MFTPL y PUNIF.** Tomando nuevamente como referencia el ejemplo 5.3, esta vez contrastaremos el desempeño, bajo las mismas condiciones vistas en 5.3, ejecuciones particulares de los predictores MFTPL probabilista, MFTPL markoviano FTPL y PUNIF.

En la Figura 5.2 se presenta una ejecución particular de la escalera, en la que la perturbación aleatoria Z de los predictores FTPL, MFTPL probabilista y MFTPL markoviano sigue una distribución $Unif[0, 0.1]$

Pérdida Acumulada

Ejemplo escalera

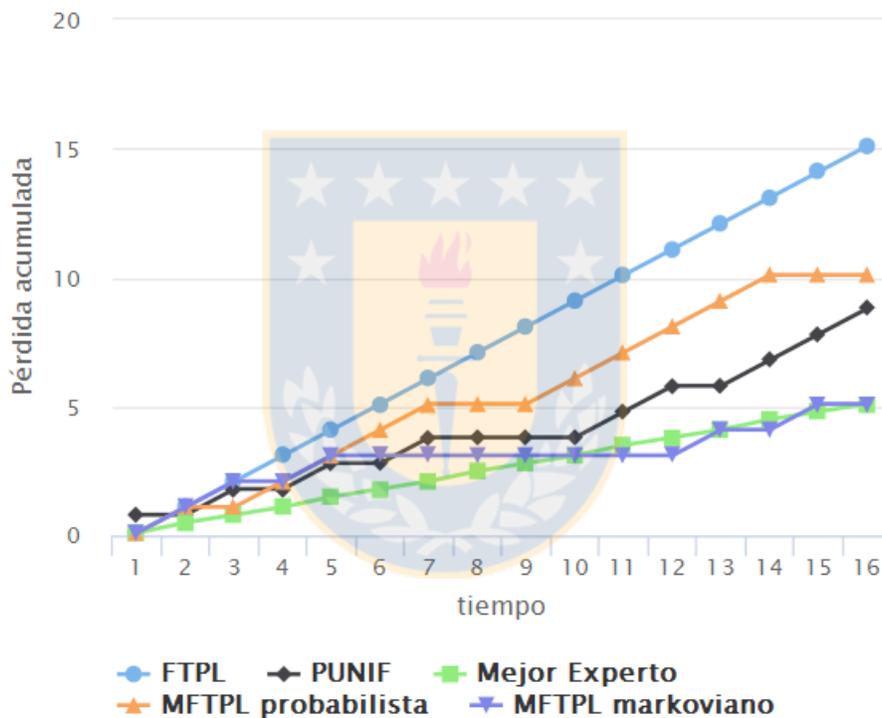


Figura 5.2: Pérdida acumulada de los predictores para una ejecución de estos sobre el ejemplo de la escalera usando una perturbación aleatoria $Unif[0, 0.1]$.

Debido a la estructura de los costos, la pérdida acumulada representa de manera muy cercana a la cantidad de fallos que comete cada predictor en su respectiva ejecución sobre este ejemplo. El *Mejor Experto* goza de ser determinista, es aquel experto cuya pérdida acumulada es la menor, por ende, independiente de las veces que ejecutemos los predictores, el mejor experto será el mismo y también su pérdida acumulada, no así los demás predictores presentados, que poseen un grado de aleatoriedad. Para el caso de FTPL, la Figura 5.2 nos indica que $L_n \approx 15$, y dada la estructura de costos, podemos concluir que desde el instante $t = 2$ en adelante falla constantemente. Este fenómeno se debe a la

perturbación aleatoria considerada, pues es tan pequeña respecto a la diferencia entre los costos presentes (0 y 1 desde $t = 2$ en adelante) que juega un rol casi despreciable para FTPL.

Es importante que el lector note que los predictores, al poseer una componente aleatoria en su mecanismo de decisión, no siempre entregarán el mismo resultado plasmado en la Figura 5.2.

Si ahora consideramos una perturbación $Unif[0, \Delta]$, con $\Delta = \sqrt{nN}$, la cual minimiza la pérdida acumulada de FTPL bajo un ambiente adversarial [4, pg. 76], obtenemos

Pérdida Acumulada

Ejemplo escalera

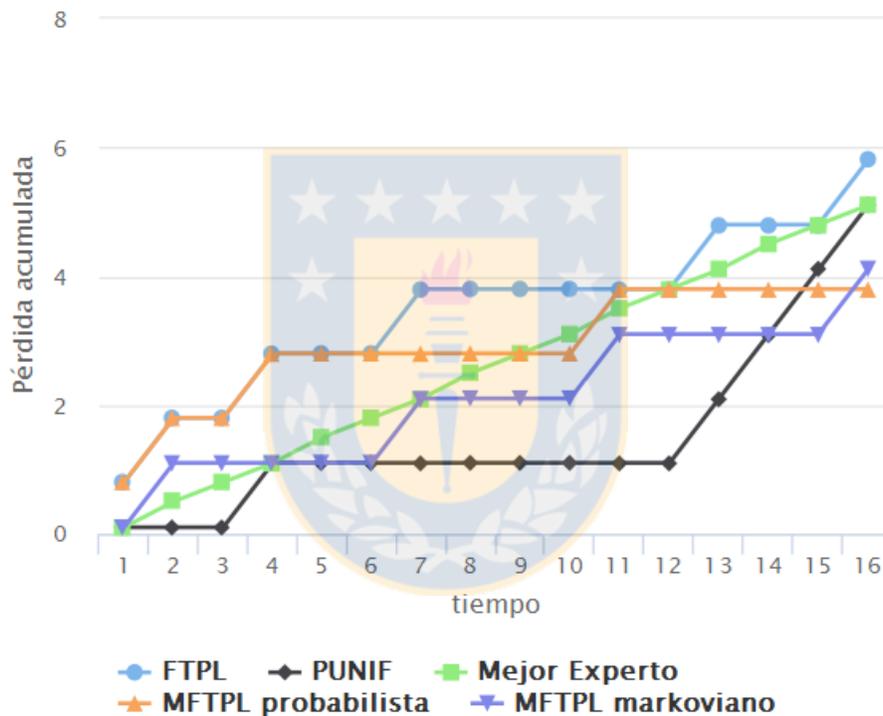


Figura 5.3: Pérdida acumulada de los predictores para el ejemplo de la escalera bajo una perturbación aleatoria $Unif[0, \sqrt{nN}]$, la cual minimiza la pérdida acumulada para el caso de FTPL.

Al comparar la Figura 5.2 con la Figura 5.3 podemos observar que el *Mejor experto* no cambia entre un gráfico y otro, esto se debe a que el *Mejor experto* utiliza la información de la perturbación aleatoria.

Acá observamos que un cambio en el parámetro Δ de la perturbación aleatoria Z influye en el comportamiento de nuestros predictores, pero es bueno notar que pese a que FTPL mejora su performance respecto al experimento anterior, MFTPL probabilista y MFTPL markoviano también sufren mejoras, esto es debido a que están poniendo atención en cuánto y cómo falla cada experto

desde sus respectivas perspectivas. Además, si pone atención en el predictor PUNIF, notará que en ocasiones puede resultar un muy buen predictor, como se ve en la Figura 5.3, y en otras no tanto.

Recordemos que MFTPL probabilista requiere conocer o creer que cada experto falla a una cierta tasa, mientras que MFTPL markoviano tiene presente que cada experto posee una cadena de markov de fallos que determina la aparición de estos en cada instante de tiempo. Por ende, para poder utilizar ambos predictores, es necesario estimar estas tasas y cadenas de fallo.

El predictor MFTPL markoviano cree que cada experto posee las probabilidades de transición que se muestran en el Cuadro 5.8

	E_1	E_2	E_3
P_{00}	0.50	0.50	0.55
P_{10}	1.00	1.00	1.00
P_{01}	0.50	0.50	0.45
P_{11}	0.00	0.00	0.00

Cuadro 5.8: Probabilidades de transición estimadas para el ejemplo de la escalera que utiliza MFTPL markoviano.

Por otro lado, nuestro predictor MFTPL probabilista cree que cada experto falla a las tasas dadas en el Cuadro 5.9 para todo instante de tiempo t .

	E_1	E_2	E_3
$q_{i,t}$	0.31	0.31	0.31

Cuadro 5.9: Tasas de fallo estimadas para el ejemplo de la escalera que utiliza MFTPL probabilista.

La perturbación utilizada se detalla en el Cuadro 5.10.

Es bueno hacer notar que los gráficos expuestos anteriormente son sólo ejecuciones particulares, las cuales cambian en cada ejecución de nuestros predictores debido al factor aleatorio intrínseco que poseen. Más adelante estaremos interesados en analizar experimentalmente el valor esperado de la pérdida acumulada para cada uno de nuestros predictores sobre este mismo ejemplo y otros de interés, así como también de sus fallos y arrepentimiento acumulado.

t	E_1	E_2	E_3
1	4.43	5.10	2.03
2	4.86	4.90	6.84
3	1.48	6.60	0.25
4	1.57	3.91	0.03
5	4.21	0.92	0.84
6	5.80	6.41	5.69
7	5.90	4.59	2.08
8	5.06	2.00	3.89
9	0.51	5.53	4.62
10	1.56	4.92	5.02
11	6.82	1.76	6.30
12	4.83	6.57	6.10
13	4.99	6.52	3.80
14	4.19	3.90	2.38
15	2.33	6.76	0.51
16	5.83	6.40	1.52

Cuadro 5.10: Perturbación $Unif[0, \sqrt{nN}]$ para ejemplo de la escalera. ■

5.4.1 Probabilidad de elección de un experto

Como se ha visto en las definiciones de MFTPL para sus distintos ambientes, hemos de notar que la única diferencia para cada escenario consiste en la probabilidad de la variable aleatoria \mathbf{M} , la cual condiciona la elección del predictor a aquellos expertos que poseen probabilidad de no fallar. Con la finalidad de conocer cotas para el valor esperado de fallos, analizaremos los distintos casos en los que se ve enfrentado el predictor MFTPL al momento de elegir un experto i .

Notemos que, la elección de MFTPL en cada instante de tiempo t se realiza sobre un conjunto $S \subset \{1, \dots, N\}$ tal que $M_{j,t} = 0 (\forall j \in S)$ y $M_{j,t} = 1 (\forall j \notin S)$, por tanto, la probabilidad de que la elección del predictor MFTPL sea i en el instante t comprenderá todas las combinaciones posibles de subconjuntos S con la condición ya mencionada, tal que i pertenezca a S , esto es,

$$\mathbb{P}(I_t = i) = \sum_{\substack{S \subseteq \{1, \dots, N\} \\ i \in S}} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) \mathbb{P}(L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \in S \setminus \{i\}))$$

Definición 5.4.3 — Función escalón. Se define la función escalón H por

$$H: \mathbb{R} \rightarrow \{0, 1\}$$

$$x \mapsto H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

Notar que la definición usual de la función escalón incluye a $x = 0$ en la preimagen de $\{1\}$, sin embargo, para nuestra utilidad, requerimos que esto no ocurra. Su aporte se verá en la demostración de la siguiente proposición.

Proposición 5.4.1 Dados N expertos, el valor esperado de fallos de los predictores MFTPL probabilista y markoviano en un instante t , $\mathbb{E}[X_t]$, bajo una perturbación continua \mathbf{Z} de la pérdida, está acotado por

$$\mathbb{E}[X_t] \leq \sum_{\substack{S \subseteq \{1, \dots, N\} \\ S \neq \emptyset}} \left[\max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} + \min_{i \in S \setminus \{k_S^*\}} \left\{ P_S^i \right\} \left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \cdot \max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \right] \cdot \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 1) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 0) \right) \right] \quad (5.3)$$

$$\mathbb{E}[X_t] \leq \sum_{\substack{S \subseteq \{1, \dots, N\} \\ S \neq \emptyset}} \left[\min_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} + \max_{i \in S \setminus \{i_S^*\}} \left\{ P_S^i \right\} \left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \cdot \min_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \right] \cdot \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 1) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 0) \right) \right] \quad (5.4)$$

donde $P_S^i := \mathbb{P}(L_{i,t-1} + Z_{i,t} \leq L_{j,t-1} + Z_{j,t} (\forall j \in S \setminus \{i\}))$, $i_S^* = \operatorname{argmin}_{i \in S} \mathbb{P}(i \text{ falla en } t)$ y $k_S^* = \operatorname{argmax}_{i \in S} \mathbb{P}(i \text{ falla en } t)$

Demostración. Siguiendo la técnica usada en el teorema 4.1.1, sabemos que

$$\mathbb{E}[X_t] = \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \mathbb{P}(I_t = i)$$

donde,

$$\begin{aligned} \mathbb{P}(I_t = i) &= \sum_{\substack{S \subseteq \{1, \dots, N\} \\ i \in S}} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) P_S^i \\ &= \sum_{S \subseteq \{1, \dots, N\}} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) P_S^i \cdot H(|S \cap \{i\}|) \end{aligned}$$

por tanto,

$$\begin{aligned} \mathbb{E}[X_t] &= \sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) \cdot \sum_{S \subseteq \{1, \dots, N\}} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) P_S^i \cdot H(|S \cap \{i\}|) \\ &= \sum_{S \subseteq \{1, \dots, N\}} \sum_{i=1}^N \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) \mathbb{P}(i \text{ falla en } t) P_S^i \cdot H(|S \cap \{i\}|) \\ &= \sum_{S \subseteq \{1, \dots, N\}} \sum_{\substack{i=1 \\ i \in S}}^N \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) \mathbb{P}(i \text{ falla en } t) P_S^i \\ &= \sum_{S \subseteq \{1, \dots, N\}} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 0) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 1) \right) \sum_{i \in S} \mathbb{P}(i \text{ falla en } t) P_S^i \quad (5.5) \end{aligned}$$

Ahora, siguiendo la técnica de demostración usada en la proposición 5.1.2, usamos $k_S^* = \operatorname{argmax}_{i \in S} \mathbb{P}(i \text{ falla en } t)$, mientras que del lema 5.1.1 tenemos que

$$P_S^{k_S^*} = 1 - \sum_{\substack{i \in S \\ i \neq k_S^*}} P_S^i$$

Por lo tanto,

$$\begin{aligned} \sum_{i \in S} \mathbb{P}(i \text{ falla en } t) P_S^i &= \mathbb{P}(k_S^* \text{ falla en } t) P_S^{k_S^*} + \sum_{\substack{i \in S \\ i \neq k_S^*}} \mathbb{P}(i \text{ falla en } t) P_S^i \\ &= \mathbb{P}(k_S^* \text{ falla en } t) \left[1 - \sum_{\substack{i \in S \\ i \neq k_S^*}} P_S^i \right] + \sum_{\substack{i \in S \\ i \neq k_S^*}} \mathbb{P}(i \text{ falla en } t) P_S^i \\ &= \mathbb{P}(k_S^* \text{ falla en } t) + \sum_{\substack{i \in S \\ i \neq k_S^*}} \underbrace{\left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(k_S^* \text{ falla en } t) \right)}_{\leq 0} P_S^i \\ &\leq \mathbb{P}(k_S^* \text{ falla en } t) + \min_{i \in S \setminus \{k_S^*\}} \left(P_S^i \right) \sum_{\substack{i \in S \\ i \neq k_S^*}} \left(\mathbb{P}(i \text{ falla en } t) - \mathbb{P}(k_S^* \text{ falla en } t) \right) \\ &= \mathbb{P}(k_S^* \text{ falla en } t) + \min_{i \in S \setminus \{k_S^*\}} \left(P_S^i \right) \left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \mathbb{P}(k_S^* \text{ falla en } t) \right] \end{aligned} \quad (5.6)$$

así, usando 5.6 en 5.5, obtenemos el resultado.

De manera análoga se obtiene la cota 5.4, debemos usar i_S^* en lugar de k_S^* y efectuar el mismo desarrollo. ■

Nota En el resultado anterior se excluye el conjunto $S = \emptyset$ debido a que eso implica que, en el instante t , el predictor cree que todos los expertos fallan, por tanto, por definición, ninguno de ellos puede ser elegido por MFTPL.

Corolario 5.4.2 Dados N expertos, el valor esperado de fallos de los predictores MFTPL probabilista y markoviano en un instante t , $\mathbb{E}[X_t]$, **bajo cualquier perturbación** continua \mathbf{Z} de la pérdida, está acotado por

$$\mathbb{E}[X_t] \leq \sum_{\substack{S \subseteq \{1, \dots, N\} \\ S \neq \emptyset}} \max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 1) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 0) \right) \quad (5.7)$$

Demostración. Aplicando la siguiente cota en la Proposición 5.4.1,

$$\underbrace{\min_{i \in S \setminus \{i_S^*\}} \left\{ P_S^i \right\}}_{\geq 0} \underbrace{\left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \cdot \max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \right]}_{\leq 0} \leq 0$$

se obtiene el corolario. ■

Nota

Es interesante saber **cuándo las cotas anteriores son ajustadas**. Pues, a diferencia de la proposición 5.1.2, podemos afirmar que estas cotas son ajustadas en un caso (hasta el momento):

- **Caso 1** Las probabilidades de fallo para cada experto i en el instante t coinciden.

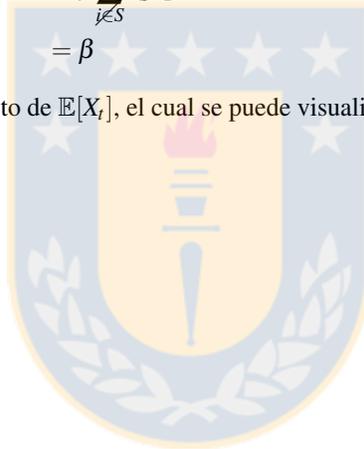
Notemos que en la demostración de la proposición 5.4.1, lo que define la cota es la expresión

$$\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) P_S^i$$

en el caso en que todas las probabilidades de fallo coinciden y son iguales a algún valor $\beta \in (0, 1)$, tenemos que

$$\begin{aligned} \sum_{i \in S} \mathbb{P}(i \text{ falla en } t) P_S^i &= \sum_{i \in S} \beta \cdot P_S^i \\ &= \beta \sum_{i \in S} P_S^i, \text{ por lema 5.1.1} \\ &= \beta \end{aligned}$$

teniendo el valor exacto de $\mathbb{E}[X_t]$, el cual se puede visualizar directamente en el corolario 5.4.2



5.5 Resumen de cotas

Hemos visto los predictores PUNIF, FTPL, MFTPL probabilista y MFTPL markoviano y muchas cotas para el valor esperado de fallos, $\mathbb{E}[X]$. Haremos una recapitulación de cada cota y haremos referencia a cuál o cuáles predictores aplica.

$$\mathbb{E}[X] \leq n \cdot \max_j \{p_j\}, \text{ ver 4.1.1} \quad (5.8)$$

$$\mathbb{E}[X] \leq n \max_t \max_j \left\{ a_0^j + \frac{(K_j)^t - 1}{K_j - 1} P_{0,1}^j \right\}, \text{ ver 4.2.2} \quad (5.9)$$

$$\mathbb{E}[X_t] = \frac{\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t)}{N}, \text{ ver 5.2.1} \quad (5.10)$$

$$\mathbb{E}[X_t] \leq \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N \min_{i \in S} \mathbb{P}(i \text{ falla en } t) \right) \max_{i \in S \setminus \{i^*\}} \mathbb{P}(I_t = i) + \min_{i \in S} \mathbb{P}(i \text{ falla en } t), \text{ ver 5.1.2} \quad (5.11)$$

$$\mathbb{E}[X_t] \leq \left(\sum_{i=1}^N \mathbb{P}(i \text{ falla en } t) - N \max_{i \in S} \mathbb{P}(i \text{ falla en } t) \right) \min_{i \in S \setminus \{k^*\}} \mathbb{P}(I_t = i) + \max_{i \in S} \mathbb{P}(i \text{ falla en } t), \text{ ver 5.1.2} \quad (5.12)$$

$$\mathbb{E}[X_t] \leq \sum_{\substack{S \subseteq \{1, \dots, N\} \\ S \neq \emptyset}} \left[\max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} + \min_{i \in S \setminus \{k_s^*\}} \left\{ P_S^i \right\} \left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \cdot \max_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \right] \cdot \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 1) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 0) \right) \right], \text{ ver 5.4.1} \quad (5.13)$$

$$\mathbb{E}[X_t] \leq \sum_{\substack{S \subseteq \{1, \dots, N\} \\ S \neq \emptyset}} \left[\min_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} + \max_{i \in S \setminus \{i_s^*\}} \left\{ P_S^i \right\} \left[\sum_{i \in S} \mathbb{P}(i \text{ falla en } t) - |S| \cdot \min_{i \in S} \left\{ \mathbb{P}(i \text{ falla en } t) \right\} \right] \cdot \left(\prod_{j \in S} \mathbb{P}(M_{j,t} = 1) \right) \left(\prod_{j \notin S} \mathbb{P}(M_{j,t} = 0) \right) \right], \text{ ver 5.4.1} \quad (5.14)$$

Predictor	5.8	5.9	5.10	5.11	5.12	5.13	5.14
PUNIF	✓	✓	✓	✓	✓	-	-
FTPL	✓	✓	-	✓	✓	-	-
MFTPL probabilista	✓	✓	-	✓	✓	✓	✓
MFTPL markoviano	✓	✓	-	✓	✓	✓	✓

Cuadro 5.11: Resumen de cotas para $\mathbb{E}[X]$

6. Experimentos

En este capítulo veremos experimentos en los que analizaremos empíricamente los resultados del valor esperado de fallos, incorporando los valores esperados de L_n y R_n y comparando estos valores entre los predictores que hemos visto.

Nuestros predictores MFTPL markoviano y MFTPL probabilista requieren como entrada valores de las probabilidades de transición y probabilidades de fallo de cada experto. Estas son estimadas mediante funciones que toman la tabla de costos completa (previo a la ejecución de cada predictor) y entregan, de manera frecuentista, las probabilidades de fallo (en el caso de MFTPL probabilista) y las de transición en la cadena de Markov de fallos (en el caso de MFTPL markoviano) para cada experto.

Como se ha mencionado, haremos estimaciones de los valores esperados de fallo, pérdida y arrepentimiento acumulados. Lo anterior se obtendrá mediante la aplicación del método Monte-Carlo con un total de 10000 ejecuciones para cada experimento. Mantendremos fija sólo la tabla de costos, y dependiendo del ambiente, otros parámetros como los fallos (ambiente adversarial), tasas de fallo (ambiente probabilista) o cadenas de Markov de fallos (ambiente markoviano) de cada experto.

Al final se presenta la aplicación de nuestros predictores sobre datos reales de latencia, donde no tenemos certeza del ambiente que los domina. Analizaremos el desempeño de los expertos considerando distintos valores de latencia para representar los fallos, viendo el desempeño de los predictores en estos distintos escenarios.

En lo que sigue presentaremos tablas para cada experimento donde explicitaremos los valores esperados de fallo, pérdida acumulada y arrepentimiento acumulado, que denotaremos por VE en los últimos dos casos, y por $\mathbb{E}[X]$ en el caso de los fallos. Además denotaremos por σ a la desviación estándar.

Las simulaciones fueron realizadas usando el software estadístico R, versión 3.3.2 y el paquete Snow para computación paralela.

6.1 Métodos de Monte-Carlo

Los métodos de Monte-Carlo son métodos numéricos basados en el muestreo aleatorio y utilizados para construir aproximaciones de objetos cuyo valor exacto es difícil de calcular. Aproximar en este caso, significa proveer de un valor (llamado aproximación), una cota para el error cometido y una metodología que permita reducir este error conforme a la necesidad. Por supuesto que reducir el error requiere de un costo computacional y tiempo de cálculo mayor.

La capacidad de los métodos de Monte-Carlo para obtener aproximaciones numéricas para problemas complejos proveen de una poderosa herramienta matemática. La manera en que deba implementarse el método, siempre está en directa relación con el problema en que será aplicado. Sin embargo, los diferentes enfoques tienden a seguir un patrón determinado. En general, se desea generar realizaciones de un objeto estocástico y posteriormente utilizar las muestras obtenidas para aproximar alguna función de interés. Así, el método hace uso del muestreo aleatorio como una herramienta para producir observaciones sobre las que se realizan inferencias estadísticas para extraer información sobre las observaciones.

Típicamente, las etapas del método de Monte-Carlo son:

1. Generar realizaciones de un objeto estocástico con una determinada densidad de probabilidad.
2. Realizar cálculos deterministas usando cada una de las muestras generadas.
3. Agregar los cálculos realizados al resultado final deseado.
4. Realizar un análisis de convergencia de la respuesta, es decir, asegurarse de que los resultados sean representativos y que cambiando el número de realizaciones de la variable o vector aleatorio, las estadísticas de la muestra se encuentran dentro de un rango de valores prescrito.

El método Monte-Carlo se apoya principalmente en dos de los principales teoremas de la teoría de probabilidades: la *Ley de los grandes números*, que garantiza la convergencia de las aproximaciones obtenidas a través del método, y el *Teorema del límite central*, que especifica la forma en que ocurre la convergencia.

A continuación enunciaremos tales teoremas:

Teorema 6.1.1 — Ley de los grandes números (16) . Sea X_1, X_2, \dots una sucesión de variables aleatorias independientes, cada una con media μ y varianza σ^2 . Definamos la variable aleatoria $S_n = X_1 + \dots + X_n$. Entonces, S_n/n converge en media cuadrática a μ , es decir,

$$\left\| \frac{S_n}{n} - \mu \right\| = \sqrt{\mathbb{E} \left[\left(\frac{S_n}{n} - \mu \right)^2 \right]} \rightarrow 0, \text{ cuando } n \rightarrow \infty$$

Teorema 6.1.2 — Teorema del límite central (7) . Sea X_1, X_2, \dots una sucesión de variables aleatorias i.i.d., cada una de media μ y varianza σ^2 . Sea $S_n = X_1 + \dots + X_n$. Definamos la variable aleatoria

$$Z_n = \frac{S_n - \mu n}{\sigma \sqrt{n}}$$

y notemos que ella tiene media 0 y varianza 1. Cuando $n \rightarrow \infty$, Z_n converge en distribución a una variable aleatoria Z con distribución acumulada de probabilidades Gaussiana estándar. Es decir,

$$\mathbb{P}(Z_n \leq x) \rightarrow \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{\alpha^2}{2}} d\alpha, \text{ cuando } n \rightarrow \infty$$

Utilizaremos el método Monte-Carlo para obtener los valores esperados de fallos, pérdida acumulada y arrepentimiento acumulado, que serán aproximadas usando el promedio aritmético de cada una usando 10000 muestras en cada caso.



6.2 Ambiente Adversarial

Utilizaremos el ya conocido ejemplo de la escalera que introdujimos en el Capítulo 5, pero para mayor cantidad de instantes de tiempo y de expertos.

Hemos de considerar:

- Instantes de tiempo $n = 1001$
- Expertos $N = 10$
- Perturbación aleatoria $Z \sim Unif[0, \Delta]$, con $\Delta = \sqrt{nN}$

En el Cuadro 6.1 se pueden apreciar los primeros 15 instantes de tiempo para la tabla de costos a considerar en el experimento del ambiente adversarial.

t	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
1	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
2	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
12	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00

Cuadro 6.1: Extracto de la tabla de costos para el ejemplo de un ambiente adversarial.

En el Cuadro 6.1 sólo se presentan costos no nulos en el primer instante de tiempo, luego los expertos poseen costo nulo en caso de no fallar. Puede notar que la tabla de costos se compone, desde el instante $t = 2$ en adelante, de matrices identidad formando la tendencia de escalera de fallos.

En las secciones siguientes analizaremos de manera separada los fallos, pérdida acumulada y arrepentimiento acumulado de cada predictor que hemos visto.

6.2.1 Fallos

En la Figura 6.1 presentamos la función de densidad para el valor esperado de fallos resultante del Monte-Carlo aplicado a nuestra tabla de costos.

Función de densidad del valor esperado de Fallos

Ambiente adversarial

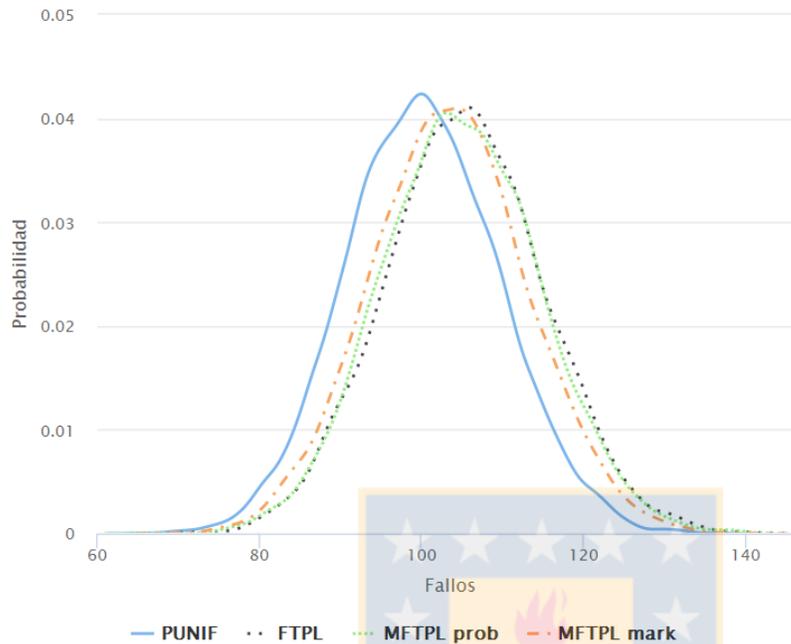


Figura 6.1: Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Adversarial)

Podemos observar que las funciones de densidad tienden a tener una distribución normal, centrada en el valor esperado de fallos de cada uno y con desviaciones bastante similares, los cuales se explicitan en el Cuadro 6.2. Además, los predictores FTPL, MFTPL probabilista y MFTPL markoviano presentan distribuciones extremadamente similares.

Predictor	$\mathbb{E}[X]$	σ
1 PUNIF	100.12	9.41
2 FTPL	105.27	9.79
3 MFTPL_prob	104.83	9.74
4 MFTPL_mark	103.35	9.66

Cuadro 6.2: Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente adversarial.

En general, los valores esperados de fallo de los cuatro predictores estudiados son muy cercanos en este ejemplo, donde el predictor PUNIF presenta el menor valor, mientras que FTPL es quien tiene el más alto. Para efectos de decisión, en este caso cualquier predictor tendrá una performance similar, con una proporción de fallos que ronda el 10% respecto del total de instantes de tiempo. Además, puede notar que las desviaciones estándar para los fallos son prácticamente iguales.

6.2.2 Pérdida acumulada

En la Figura 6.2 presentamos la función de densidad para la pérdida acumulada bajo el ejemplo adversarial.

Función de densidad del valor esperado de la Pérdida Acumulada

Ambiente adversarial

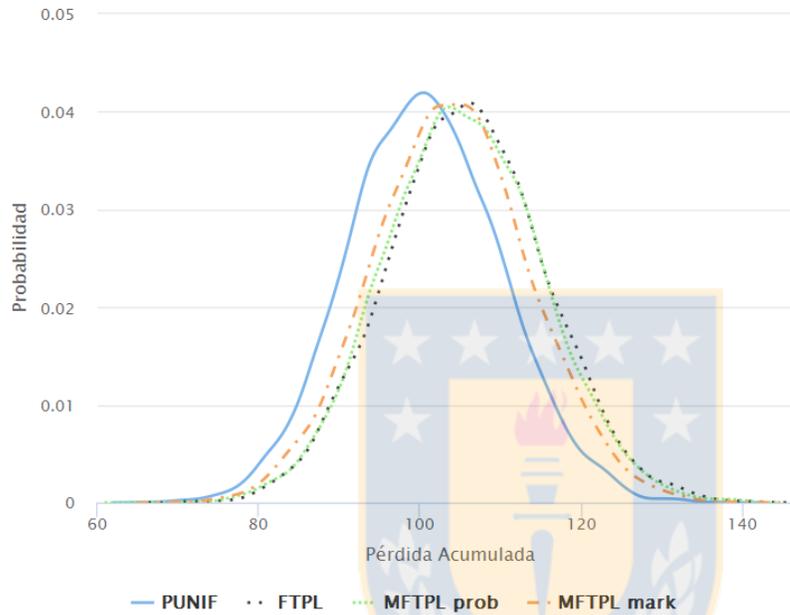


Figura 6.2: Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Adversarial).

Puede observar que la similitud con la función de densidad del valor esperado de fallos es muy grande, esto se debe a que, salvo en el instante $t = 1$, los fallos determinan los costos en nuestro ejemplo.

Los valores esperados de la pérdida acumulada para cada predictor en este experimento se detallan en la siguiente tabla

	Predictor	VE	σ
1	PUNIF	100.57	9.41
2	FTPL	105.72	9.79
3	MFTPL_prob	105.29	9.74
4	MFTPL_mark	103.80	9.66

Cuadro 6.3: Valores esperados y desviación estándar de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente adversarial.

6.2.3 Arrepentimiento acumulado

En cuanto a la función de densidad del valor esperado del arrepentimiento acumulado, se mantiene la normalidad de estas y también la tendencia que se ve en los fallos y la pérdida acumulada, donde FTPL, MFTPL probabilista y MFTPL adversarial tienden a ser muy similares, mientras que para el predictor PUNIF, la curva de densidad está centrada más a la izquierda que las demás.

Función de densidad del valor esperado del Arrepentimiento Acumulado

Ambiente adversarial

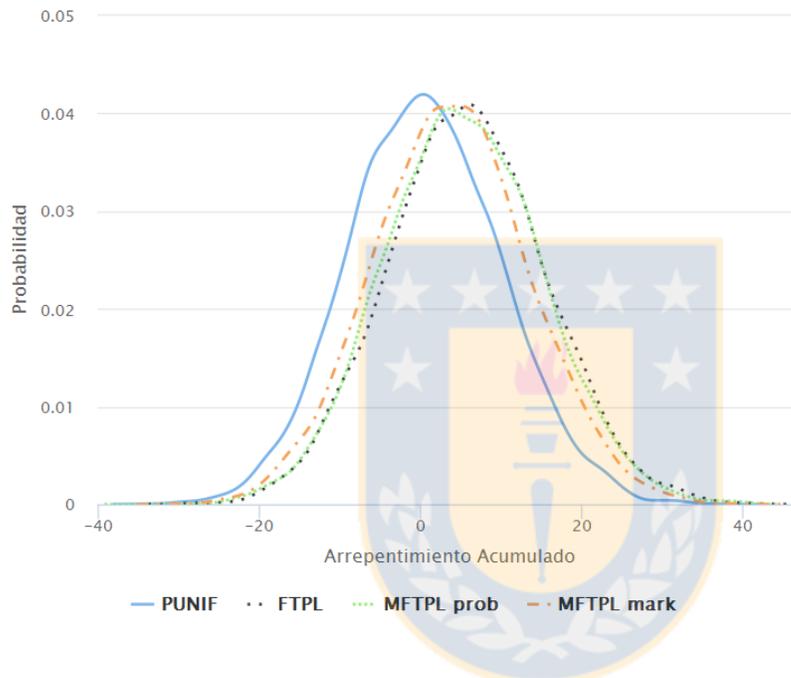


Figura 6.3: Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Adversarial)

Por supuesto que el arrepentimiento acumulado de cada predictor se condice con su pérdida acumulada, pero además, es interesante notar que el predictor PUNIF posee arrepentimiento muy cercano a 0, esto quiere decir que se comporta de manera similar al mejor experto en el último instante $t = 1001$. En el Cuadro 6.4 se muestra el valor esperado y desviación estándar del arrepentimiento acumulado para cada predictor en el ejemplo adversarial.

	Predictor	VE	σ
1	PUNIF	0.47	9.41
2	FTPL	5.62	9.79
3	MFTPL_prob	5.19	9.74
4	MFTPL_mark	3.70	9.66

Cuadro 6.4: Valores esperados y desviación estándar del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente adversarial.

En este ejemplo, tanto los costos como los fallos fueron fijos, en lo que sigue, sólo mantendremos fija la tabla de costos, los fallos serán determinados mediante tasas de fallo o cadenas de markov de fallos según sea el caso.

6.3 Ambiente Probabilista

Para este caso, experimentaremos con una tabla de costos fija a la cual asignaremos probabilidades de fallo para cada experto y se generarán fallos en base a tales tasas para cada experimento.

Hemos de considerar

- Instantes de tiempo $n = 1000$
- Expertos $N = 10$
- Perturbación aleatoria $Z \sim Unif[0, \Delta]$, con $\Delta = \sqrt{nN}$
- Tasas de fallo por experto dadas en el Cuadro 6.5

E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
0.01	0.08	0.15	0.23	0.30	0.70	0.75	0.80	0.85	0.90

Cuadro 6.5: Tasas de fallo de cada experto en el ejemplo de un ambiente probabilista.

En el Cuadro 6.6 se muestra un extracto de la tabla de costos a considerar. Note que no presenta fallos, esto se debe a que, como se mencionó en la parte introductoria del capítulo, utilizaremos esta tabla de costos fija y generaremos fallos en cada uno de los 10000 experimentos en función de las tasas de fallo explicitadas en el Cuadro 6.5.

t	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
1	0.55	0.70	0.63	0.14	0.15	0.21	0.84	0.02	0.75	0.71
2	0.31	0.03	0.65	0.91	0.38	0.01	0.47	0.08	0.41	0.73
3	0.96	0.79	0.96	0.96	0.68	0.09	0.06	0.66	0.89	0.31
4	0.18	0.92	0.93	0.19	0.83	0.96	0.12	0.01	0.21	0.76
5	0.88	0.57	0.13	0.23	0.94	0.92	0.03	0.17	0.15	0.69
6	0.42	0.28	0.50	0.71	0.29	0.03	0.35	0.15	0.84	0.66
7	0.27	0.34	0.08	0.19	0.79	0.80	1.00	0.21	0.02	0.19
8	0.35	0.98	0.14	0.64	0.66	0.78	0.33	0.76	0.82	0.06
9	0.95	0.25	0.47	0.60	0.62	0.10	0.68	0.71	0.99	0.74
10	0.28	0.60	0.18	0.49	0.58	0.98	0.30	0.46	0.96	0.07
11	0.58	0.37	0.11	0.39	0.69	0.09	0.14	0.11	0.41	0.60
12	0.26	0.17	0.65	0.61	0.23	0.39	0.03	0.32	0.36	0.16
13	0.59	0.51	0.99	0.22	0.82	0.18	0.42	0.66	0.11	0.82
14	0.35	0.90	0.93	0.61	0.00	0.66	0.32	0.01	0.87	0.94
15	0.09	0.67	0.60	0.90	0.88	0.50	0.28	0.35	0.29	0.86

Cuadro 6.6: Extracto de la tabla de costos para el ejemplo de un ambiente probabilista.

6.3.1 Fallos

A diferencia del ambiente adversarial, en el ambiente probabilista sabemos que nuestros predictores MFTPL probabilista y MFTPL markoviano están preparados para rescatar información del comportamiento de los fallos de cada experto. De acuerdo a lo anterior, es de esperar que, a diferencia del ejemplo anterior, nuestros predictores disminuyan la cantidad de fallos que cometen.

En la Figura 6.4 podemos encontrar las funciones de densidad de los fallos para cada predictor. Además, en el Cuadro 6.7 mostramos los valores esperados de fallo y desviación estándar estimados mediante Monte-Carlo para cada uno de ellos.

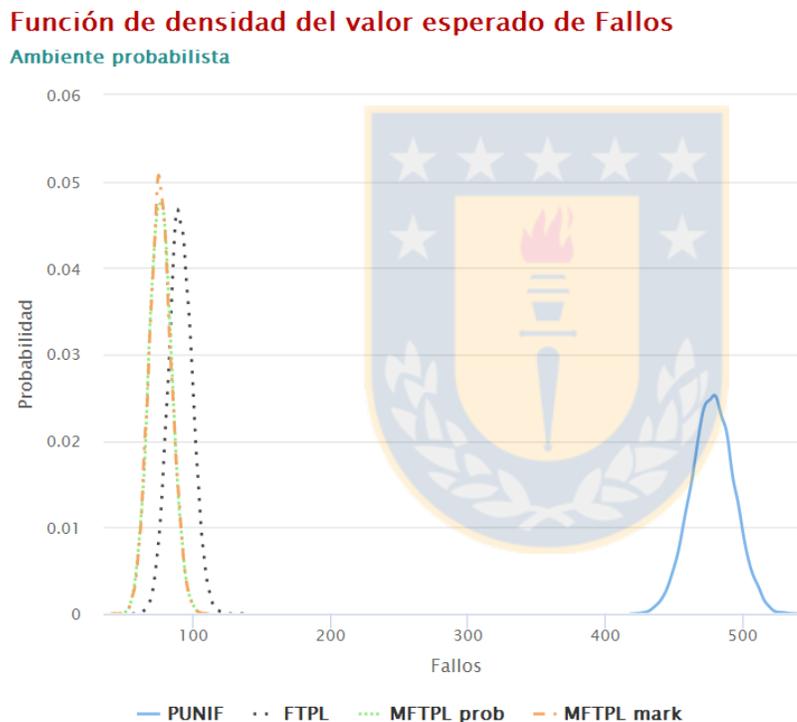


Figura 6.4: Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Probabilista).

Al observar la Figura 6.4 podemos notar que el predictor PUNIF está lejos de ser una buena opción en casos generales (como se podría haber pensado al ver su performance en el ejemplo del ambiente adversarial). Podemos observar además que los predictores MFTPL probabilista y MFTPL markoviano son quienes poseen la menor cantidad de fallos en los experimentos. Esto queda en evidencia más claramente en el Cuadro 6.7, de donde también podemos concluir que los predictores fallan en un rango bastante más acotado respecto a su valor esperado que PUNIF.

	Predictor	$\mathbb{E}[X]$	σ
1	PUNIF	477.60	15.72
2	FTPL	90.65	8.55
3	MFTPL_prob	76.20	8.19
4	MFTPL_mark	76.14	8.15

Cuadro 6.7: Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente probabilista.

6.3.2 Pérdida acumulada

La Figura 6.5 muestra las funciones de densidad para la pérdida acumulada de cada predictor sobre el ejemplo del ambiente probabilista. Notará que, como ha sido la tónica, hay una estrecha relación en cómo se comportan los fallos y la pérdida acumulada de cada predictor. La gráfica ya mencionada se complementa con los valores esperados y desviación estándar de L_n que aparecen en el Cuadro 6.8. Allí podemos observar que MFTPL probabilista (quien se espera que tenga ventaja en este ambiente) se comportan tan bien como MTPL markoviano. Lo anterior no debería ser una sorpresa, pues vimos en la subsección 4.2.1 que el modelo markoviano de fallos generaliza al probabilista.

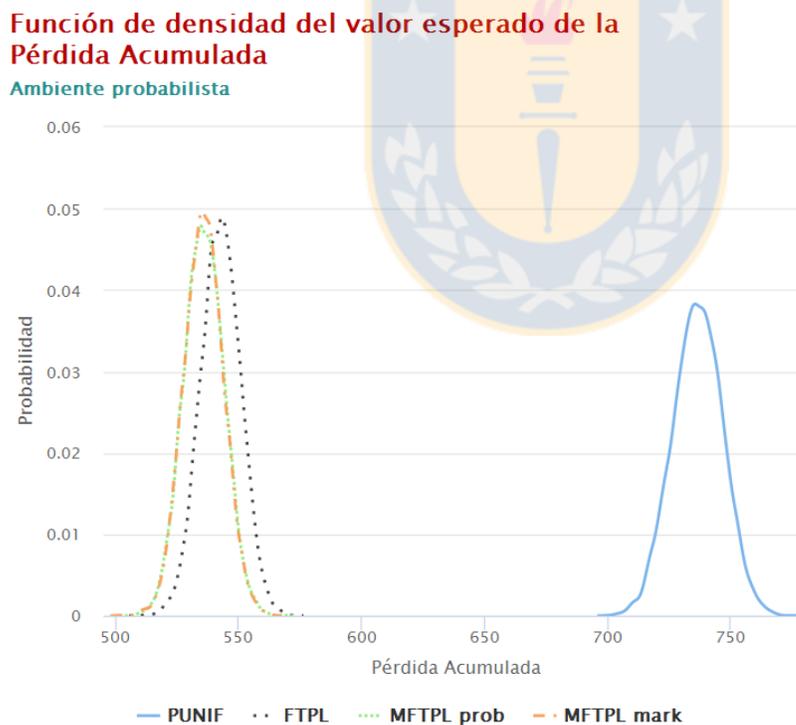


Figura 6.5: Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Probabilista).

En la pérdida acumulada se puede ver la misma tendencia que en los fallos, MFTPL probabilista

y MFTPL markoviano se comportan de manera muy similar, superando por poco a FTPL, mientras que PUNIF está lejos de ser un predictor deseable sobre este escenario.

	Predictor	VE	σ
1	PUNIF	736.42	10.14
2	FTPL	542.53	8.11
3	MFTPL_prob	535.71	8.08
4	MFTPL_mark	535.62	7.99

Cuadro 6.8: Valores esperados y desviación estándar de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente probabilista.

6.3.3 Arrepentimiento acumulado

La Figura 6.6 presenta las funciones de densidad del arrepentimiento acumulado para cada predictor, y en el Cuadro 6.9 puede apreciar los valores esperados y desviación estándar de R_n . Analizando la gráfica y el cuadro, podemos observar que las mismas tendencias que hemos visto en los fallos y la pérdida acumulada se repiten para el arrepentimiento acumulado.

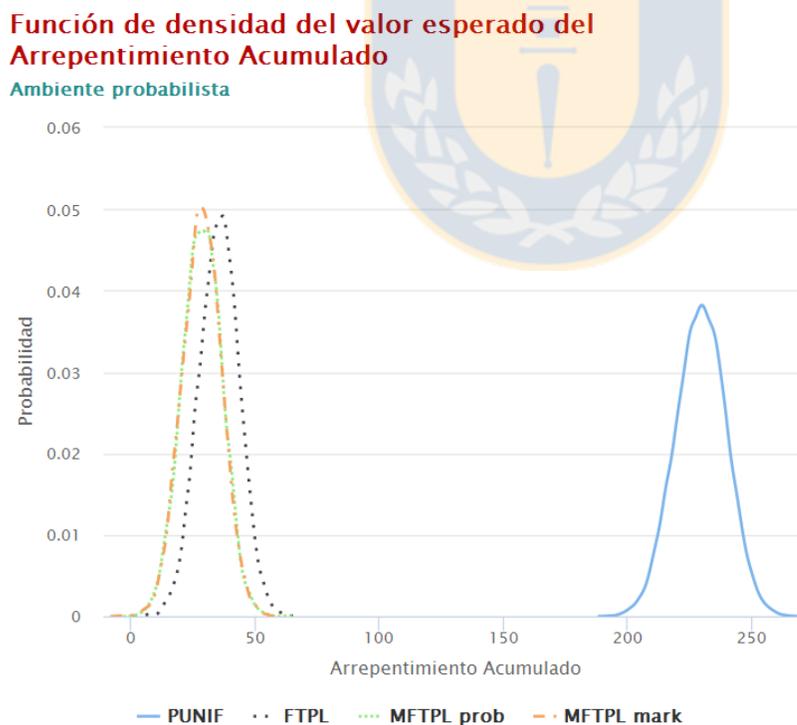


Figura 6.6: Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Probabilista).

	Predictor	VE	σ
1	PUNIF	229.30	10.26
2	FTPL	35.41	8.02
3	MFTPL_prob	28.59	7.97
4	MFTPL_mark	28.50	7.88

Cuadro 6.9: Valores esperados y desviación estándar del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente probabilista.

Es claro que hay una diferencia sustancial entre PUNIF y los demás predictores, dejando en evidencia que el uso de la información sobre la pérdida acumulada tiene una incidencia importante en las decisiones, como ya se sabe para FTPL, pero más aún, usar la información de los fallos en las decisiones permite una mejora aun mayor.

6.4 Ambiente Markoviano

Al igual que en el experimento del ambiente probabilista, aquí utilizaremos una tabla con costos fijos, pero esta vez, cada experto tendrá una cadena de Markov de fallos asociada, con la cual generaremos fallos para cada uno de sus instantes de tiempo.

Hemos de considerar

- Instantes de tiempo $n = 1000$
- Expertos $N = 10$
- Perturbación aleatoria $Z \sim Unif[0, \Delta]$, con $\Delta = \sqrt{nN}$
- Las probabilidades de transición de cada experto en su cadena de Markov de fallos dadas en el Cuadro 6.10

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
P_{00}	0.97	0.05	0.08	0.69	0.16	0.37	0.39	0.08	0.85	0.32
P_{11}	0.63	0.72	0.82	0.04	0.24	0.31	0.41	0.32	0.70	0.65

Cuadro 6.10: Probabilidades de transición de la cadena de Markov de fallos de cada experto en el ejemplo de un ambiente markoviano.

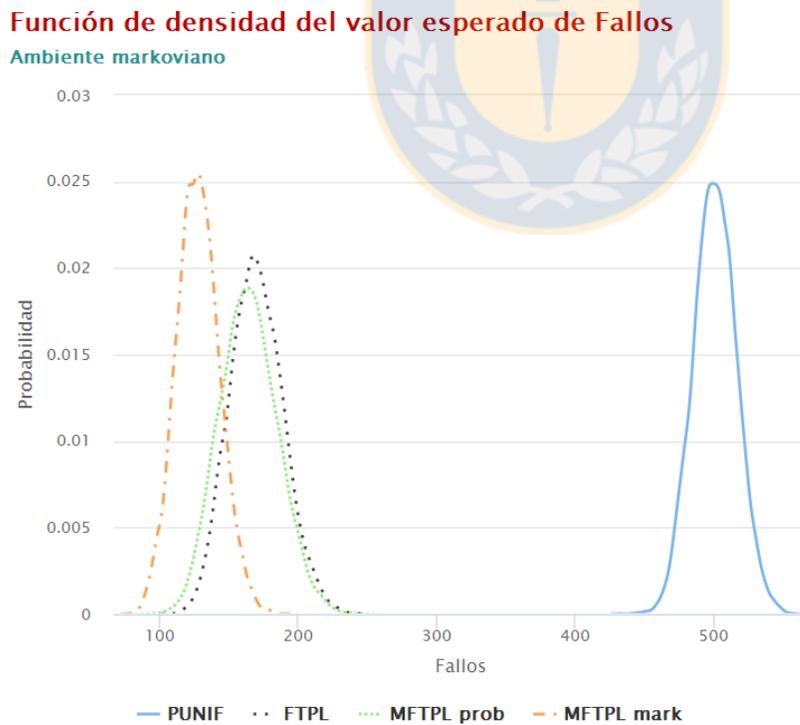
- Probabilidad inicial de fallo para cada experto: $a_0 = 1/2$

En el Cuadro 6.11 se pueden apreciar los primeros 15 instantes de tiempo para la tabla de costos a considerar, donde nuevamente se puede observar que no hay presencia de fallos en los expertos, pues estos serán generados en cada ejecución, siguiendo sus cadenas de Markov de fallos.

t	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
1	0.28	0.58	0.50	0.27	0.77	0.69	0.99	0.47	0.85	0.42
2	0.01	0.97	0.21	0.02	0.17	0.14	0.82	0.06	0.80	0.13
3	0.99	0.52	0.82	0.72	0.48	0.33	0.34	0.10	0.37	0.38
4	0.61	0.95	0.70	0.39	0.94	0.10	0.87	0.44	0.30	0.75
5	0.97	0.34	0.89	0.34	0.14	0.32	0.73	0.29	0.00	0.19
6	0.48	0.90	0.16	0.86	0.29	0.14	0.02	0.35	0.71	0.52
7	0.91	0.25	0.29	0.46	0.69	0.57	0.43	0.15	0.15	0.08
8	0.65	0.50	0.59	0.46	0.47	0.77	0.71	0.73	0.58	0.56
9	0.40	0.89	0.20	0.81	0.68	0.91	0.00	0.56	0.27	0.91
10	0.45	0.85	0.22	0.41	0.42	0.82	0.89	0.55	0.90	0.18
11	0.80	0.73	0.45	0.78	0.98	0.48	0.80	0.44	0.67	0.78
12	0.39	0.45	0.36	0.72	0.83	0.48	0.58	0.35	0.75	0.70
13	0.83	0.13	0.56	0.07	0.22	0.80	0.39	0.93	0.99	0.03
14	0.26	0.71	0.51	0.19	0.08	0.68	0.89	0.50	0.35	0.12
15	0.81	0.46	0.56	0.33	0.91	0.60	0.16	1.00	0.59	0.14

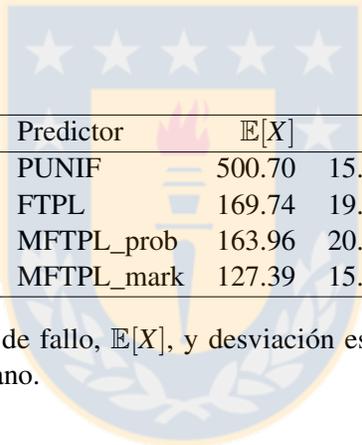
Cuadro 6.11: Extracto de la tabla de costos para el ejemplo de un ambiente adversarial.

6.4.1 Fallos

Figura 6.7: Función de densidad para valor esperado de fallos $\mathbb{E}[X]$ (Ambiente Markoviano).

En la Figura 6.7 se presenta la función de densidad para el valor esperado de fallos de cada predictor bajo el ejemplo del ambiente markoviano, en la que se observa una clara ventaja del predictor MFTPL markoviano sobre el resto de los predictores considerados. Puede observar en el Cuadro 6.12 que esta vez hay una diferencia bastante clara entre los predictores. MFTPL markoviano, además de poseer el menor valor esperado de fallos, es quien posee la menor desviación estándar, y por lo tanto, podemos decir que este predictor es más constante en su desempeño.

El predictor PUNIF, como ha sido usual en los experimentos anteriores, comete una cantidad de fallos exageradamente superior al resto, la mitad del total de instantes de tiempo aproximadamente. Por su parte, MFTPL probabilista y FTPL cometen casi los mismos fallos, que ronda el 17 %, dejando como claro mejor predictor a MFTPL markoviano, con un valor esperado de 127 fallos.



	Predictor	$\mathbb{E}[X]$	σ
1	PUNIF	500.70	15.64
2	FTPL	169.74	19.24
3	MFTPL_prob	163.96	20.89
4	MFTPL_mark	127.39	15.42

Cuadro 6.12: Valores esperados de fallo, $\mathbb{E}[X]$, y desviación estándar para cada predictor en el ejemplo de un ambiente markoviano.

6.4.2 Pérdida acumulada

La pérdida acumulada para el ejemplo del ambiente markoviano reafirma las conclusiones que se obtienen a partir de los fallos. El predictor MFTPL markoviano es claramente quien entrega los mejores resultados en este ejemplo, mientras que FTPL y MFTPL probabilista comparten el segundo lugar. Las funciones de densidad de L_n de cada predictor se presentan en la Figura 6.8, y podemos ver que, comparando los resultados del Cuadro con el Cuadro , los valores esperados y desviaciones estándar de los fallos y la pérdida acumulada presentan las mismas tendencias entre los predictores.

Función de densidad del valor esperado de la Pérdida Acumulada

Ambiente markoviano

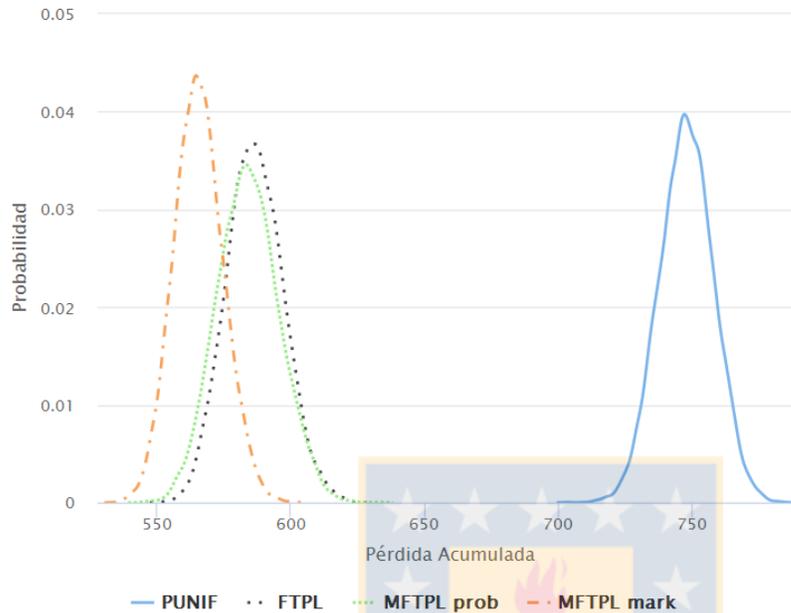


Figura 6.8: Función de densidad para valor esperado de la pérdida acumulada L_n (Ambiente Markoviano).

	Predictor	VE	σ
1	PUNIF	747.84	10.13
2	FTPL	586.38	10.58
3	MFTPL_prob	583.82	11.54
4	MFTPL_mark	565.37	9.23

Cuadro 6.13: Valores esperados de la pérdida acumulada L_n para cada predictor en el ejemplo de un ambiente markoviano.

6.4.3 Arrepentimiento acumulado

Para el caso del arrepentimiento acumulado, notará que MFTPL markoviano sigue siendo aquel predictor con mejor performance, sin embargo, las probabilidades de sus valores esperados varían al compararlos con la tendencia de los fallos y la pérdida acumulada, aquí FTPL y MFTPL probabilista presentan valores más altos para el valor esperado del arrepentimiento acumulado que el de MFTPL markoviano, caso contrario a lo ocurrido para la pérdida acumulada. Esto se evidencia comparando las curvas de la Figura 6.9 que contiene las funciones de densidad para R_n y la Figura 6.8 de las densidad de L_n .

Función de densidad del valor esperado del Arrepentimiento Acumulado

Ambiente markoviano

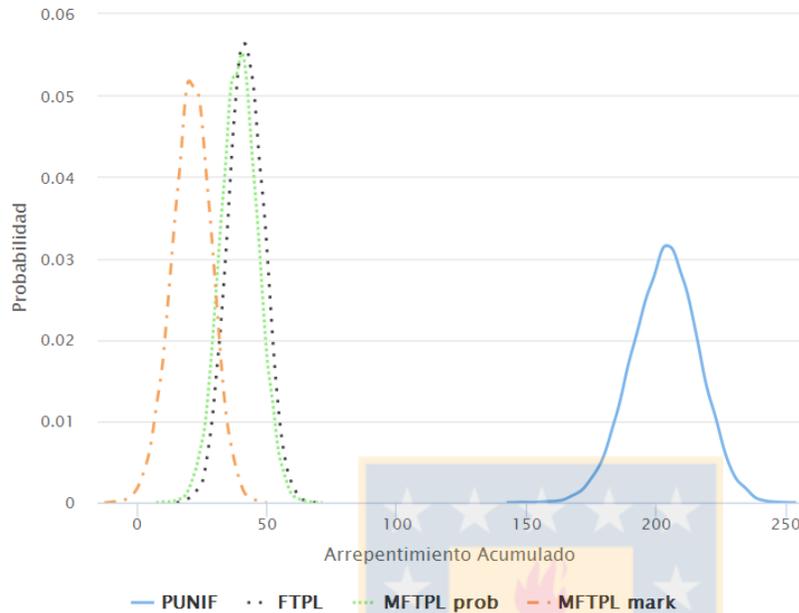


Figura 6.9: Función de densidad para valor esperado del arrepentimiento acumulado R_n (Ambiente Markoviano).

	Predictor	VE	σ
1	PUNIF	203.18	12.81
2	FTPL	41.72	6.93
3	MFTPL_prob	39.16	7.26
4	MFTPL_mark	20.71	7.70

Cuadro 6.14: Valores esperados del arrepentimiento acumulado R_n para cada predictor en el ejemplo de un ambiente markoviano

Como era de esperarse, el predictor MFTPL markoviano resulta ser la mejor opción bajo este ejemplo de un ambiente markoviano de fallos. A su vez, queda aún más claro que utilizar la información del problema que se está tratando es una técnica que da buenos resultados.

A continuación veremos cómo se comportan los predictores cuando los sometemos a datos reales, donde no sabemos cómo están modelados los fallos.

6.5 Datos Reales

En el siguiente ejemplo se utilizarán datos reales, donde los costos representan medidas de latencias de la *red NLNog* facilitada por el proyecto PANACEA. Los datos constan de 3836 mediciones de latencia por 5 días, cada 2 minutos. Cada medición corresponde a la latencia de los caminos de largo 1 entre los nodos disponibles.

Los nodos de la red se muestran en el Cuadro 6.15.

	Nombre nodo
1	Narita
2	San Francisco
3	Melbourne
4	Oslo
5	Warsaw
6	Cocoa
7	Dublin
8	HongKong
9	Tokyo
10	Haifa
11	Moscow
12	Santiago
13	Curitiba
14	Gibraltar
15	Houston
16	Calgary
17	Boston
18	Paris
19	Singapore

Cuadro 6.15: Nodos que componen la red de latencias *NLNog*.

Los valores de latencia para cada arco (i, j) se encuentran en el rango $[0, 2000]$. La latencia para caminos de largo mayor a 1 corresponde a la suma de las latencias de los arcos que conforman tal camino.

Para este experimento consideraremos como nodo fuente a *Narita* y nodo sumidero a *Singapore*. Elegiremos 15 caminos para nuestros experimentos: el camino de largo 1 que nos permite llegar desde *Narita* hasta *Singapore*, 7 caminos de largo 2 y 7 caminos de largo 3. Estos últimos fueron elegidos como aquellos caminos que poseen la mayor cantidad de valores de latencia mayores o iguales a 1000.

En el Cuadro 6.16 puede observar un extracto de los datos a utilizar de 9 de los 15 expertos para los 13 primeros instantes de tiempo.

t	1-19	1-6-19	1-5-19	1-7-19	1-4-19	1-8-19	1-10-19	1-12-19	1-12-10-19
1	80.41	410.60	615.34	553.74	627.76	111.00	735.94	764.05	1090.50
2	80.23	410.87	613.89	553.86	627.44	103.75	736.53	765.86	1090.34
3	81.37	410.61	611.90	554.55	636.80	101.28	745.73	764.89	1092.52
4	80.41	410.50	613.29	553.67	627.58	98.80	736.76	764.67	1091.90
5	81.25	410.44	613.83	553.42	627.54	108.14	736.55	763.08	1089.37
6	81.19	410.57	609.54	553.49	636.94	102.28	744.99	763.63	1089.28
7	81.26	410.74	609.70	555.84	637.41	103.79	745.53	763.50	1091.21
8	80.29	410.74	615.86	554.39	628.36	96.72	736.27	760.20	1089.98
9	80.37	412.32	609.97	553.59	628.24	92.68	736.43	763.68	1089.97
10	80.38	410.83	614.89	553.51	628.11	91.34	737.18	765.76	1092.76
11	80.22	410.69	545.38	553.48	627.64	98.09	736.34	763.55	1092.75
12	81.38	410.62	545.34	553.97	637.21	107.52	744.80	763.05	1091.76
13	81.26	410.58	543.66	553.61	636.93	96.59	745.12	762.39	1091.25

Cuadro 6.16: Extracto de la tabla de datos reales a utilizar.

Cada experto está identificado por el camino que lo compone, así, el experto 1 – 12 – 10 – 19, corresponde al camino *Nurita, Santiago, Haifa, Singapore* (ver Cuadro 6.16). Los otros expertos considerados se muestran en el Cuadro 6.17 .

	Experto
10	1-12-10-19
11	1-11-13-19
12	1-14-13-19
13	1-10-13-19
14	1-4-13-19
15	1-5-13-19
16	1-3-13-19

Cuadro 6.17: Expertos restantes considerados en los experimentos con datos reales que conectan Nurita con Singapore.

Recordemos que en nuestro modelo un fallo está determinado por el valor 1 en la tabla de costos, por tanto, la pregunta natural en este caso es ¿qué identifica un fallo? Definiremos 3 valores de fallo, correspondientes a valores máximos de latencia, estos son latencia igual o superior a 600, 1000 y 1400. Para cada uno de estos casos veremos el comportamiento de nuestros predictores en relación a los fallos y cómo es el valor esperado de cada uno respecto del mejor experto en cada instante t , más específicamente graficaremos la razón

$$F_{ME}^P(t) := \frac{\sum_{j=1}^t E[X_j]}{\text{Fallos Mejor experto hasta } t}$$

para cada uno de los expertos a lo largo de cada instante t .

En lo que sigue podrá ver las gráficas de los resultados para la razón descrita anteriormente, cabe señalar que el primer fallo cometido por el mejor experto (quien resulta ser el experto 1 – 19 para todo instante t) ocurre en el instante $t = 273$, por ende, notará que las gráficas comienzan en

dicho instante (anterior a eso, F_{ME}^P está indeterminado). Además, en cada gráfica se presenta la recta constante igual a 1 que señala el caso ideal, aquel donde el valor esperado de fallos hasta el instante t de un predictor es igual a la cantidad de fallos que comete el mejor de los expertos hasta dicho instante.

6.5.1 F_{ME}^P usando valor de fallo mayor o igual a 1400.

Partiremos definiendo un fallo como toda latencia mayor o igual a 1400 en la tabla de datos reales. De la Figura 6.10 podemos observar que los predictores FTPL, MFTPL probabilista y MFTPL markoviano cometen menos fallos (en valor esperado) que el mejor de los expertos fijos. De hecho, cometen fallos sólo en los primeros instantes (previo a $t = 300$) y luego evitan todos los fallos. Caso contrario ocurre con FTPL, quien incrementa sus fallos esporádicamente.

En este caso, debido a la gran cantidad de expertos y el alto valor que se considera como fallo (considerando que las latencias están en el intervalo $[0, 2000]$), no es raro que todos los predictores tengan valores esperados de fallo bajos (ver Cuadro 6.18), pues esto se condice con la gran ausencia de fallos. Sin embargo, es interesante notar que MFTPL sigue siendo el mejor predictor y PUNIF el peor, tal como se dio en los ejemplos de los ambientes probabilista y markoviano, pero esta vez, FTPL empeora su desempeño con la presencia de más fallos.

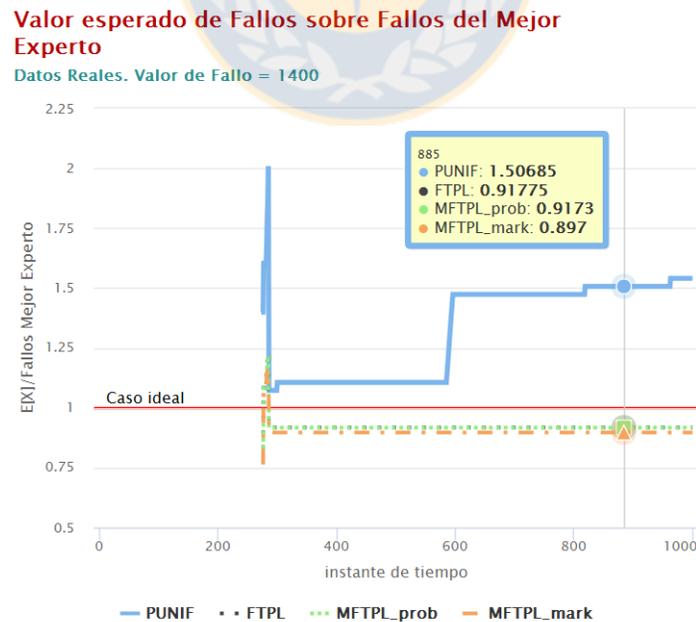


Figura 6.10: Razón F_{ME}^P para un valor de fallos fijado en 1400 (Datos Reales).

	Predictor	$\mathbb{E}[X]$	σ
1	PUNIF	3.08	1.59
2	FTPL	1.84	0.96
3	MFTPL_prob	1.83	0.96
4	MFTPL_mark	1.79	0.94

Cuadro 6.18: Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 1400.

6.5.2 F_{ME}^P usando valor de fallo mayor o igual a 1000.

Ahora hemos de considerar los fallos como latencias mayores o iguales a 1000. Haciendo esto, la cantidad de fallos en la tabla de costos sube drásticamente en comparación al caso anterior, prueba de ello es el valor esperado de fallos cometidos por PUNIF que se puede ver en el Cuadro 6.19.

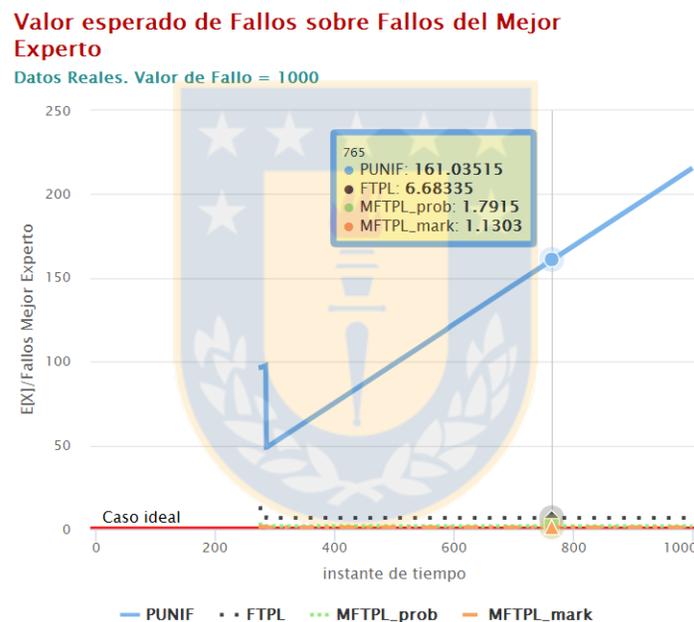


Figura 6.11: Razón F_{ME}^P para un valor de fallos fijado en 1000 (Datos Reales).

Como puede observar en la Figura 6.11 el valor esperado de fallos de PUNIF va aumentando en cada instante de tiempo en relación al mejor experto fijo, no así para los demás predictores, donde MFTPL markoviano es quien más se acerca al mejor experto, seguido por MFTPL probabilista, y más lejos está FTPL. Además, ocurre algo similar que en el caso que vimos anteriormente. Puede notar que, por la característica de las curvas, los tres predictores que mejor se comportan cometen fallos sólo al inicio, y luego la curva de F_{ME}^P de cada predictor se vuelve constante, lo cual nos dice que, o cometen los mismos fallos que el mejor experto, o ninguno comete desde un punto en adelante.

El punto más rescatable, y que debemos resaltar, es que los predictores MFTPL markoviano y MFTPL probabilista realzan su buen desempeño con la aparición de fallos.

	Predictor	$\mathbb{E}[X]$	σ
1	PUNIF	431.29	15.11
2	FTPL	13.37	3.00
3	MFTPL_prob	3.58	1.49
4	MFTPL_mark	2.26	0.96

Cuadro 6.19: Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 1000.

6.5.3 F_{ME}^P usando valor de fallo mayor o igual a 600.

Finalmente, consideraremos como valor de fallo las latencias que son iguales o superiores a 600. Con esto la cantidad de fallos aumenta y sin embargo, los valores esperados de fallo para MFTPL markoviano y MFTPL probabilista no sufren cambios muy bruscos, no así el de FTPL que incrementa a más del doble, como puede observar al comparar el Cuadro 6.19 con el Cuadro 6.20, donde tenemos los valores esperados y desviaciones estándar para los fallos de cada experto, para valores de fallo 1000 y 600, respectivamente.

Por su parte, PUNIF incrementa con creces su valor esperado de fallos en relación a los casos vistos anteriormente, llegando a 772, con una desviación estándar de apenas 13 fallos. Note que en este caso se repite la tendencia del caso anterior en las curvas de F_{ME}^P de cada predictor, donde MFTPL markoviano es quien está más cerca del mejor experto fijo, seguido muy de cerca de MFTPL probabilista.

Valor esperado de Fallos sobre Fallos del Mejor Experto

Datos Reales. Valor de Fallo = 600

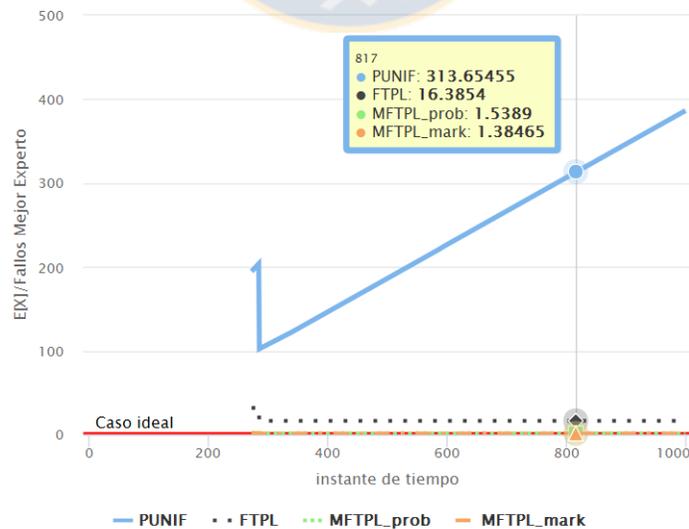


Figura 6.12: Razón F_{ME}^P para un valor de fallos fijado en 600 (Datos Reales).

	Predictor	$\mathbb{E}[X]$	σ
1	PUNIF	772.94	13.26
2	FTPL	32.77	4.32
3	MFTPL_prob	3.08	1.03
4	MFTPL_mark	2.77	0.87

Cuadro 6.20: Valor esperado de fallos y desviación estándar de cada predictor considerando un valor de fallo igual a 600.

El hecho de que MFTPL markoviano tenga una mejor performance que MFTPL probabilista nos da la idea de que el ambiente que domina los fallos en este ejemplo se parece más a uno markoviano que a uno probabilista.

Con este experimento podemos reafirmar que MFTPL markoviano es una mejora sustancial respecto a FTPL cuando hay presencia de fallos,. Además, MFTPL probabilista no está lejos de lo que logra MFTPL markoviano, pues ambos predictores, en términos globales, toman las mejores decisiones.



7. Conclusiones

En problemas de encaminamiento en redes abordados mediante el modelo de aprendizaje y predicción con ayuda de expertos, resultó muy útil definir modelos que determinen cómo se expresan los fallos en cada experto. Esta información fue primordial para obtener cotas superiores e incluso inferiores para todo predictor sometido a un problema donde los fallos seguían un modelo probabilista o markoviano de fallos. En el ambiente probabilista de fallos vimos que lo máximo a lo cual puede aspirar cualquier predictor es a comportarse tan bien como el mejor de los expertos (en valor esperado), esto nos da la idea general de que, en ambientes como el markoviano, también se debiera dar dicho caso, pero también da fuerza a la medida utilizada para verificar cuán bien se está comportando un predictor, que es la pérdida acumulada. Lo anterior no es necesariamente cierto en el ambiente adversarial, pues este no se rige por una medida de probabilidad que modele los fallos. Pudimos verificar que en el experimento de la escalera PUNIF estaba por debajo del mejor experto (por poco) en relación al valor esperado de fallos.

Los predictores MFTPL markoviano y MFTPL probabilista resultaron ser una mejora sustancial y robusta del predictor FTPL, en el sentido de que actúan tan bien como FTPL en problemas donde casi no hay presencia de fallos, y lo superan en los otros casos. En otras palabras, estos predictores generalizan al predictor FTPL, manteniendo su forma de operar en un ambiente exento de fallos y mejorándola en el caso contrario.

En base a nuestros experimentos, en los distintos ambientes que gobiernan los fallos: adversarial, probabilista y markoviano, MFTL markoviano supera a los demás predictores, con la salvedad del caso de la escalera (ambiente adversarial), construido como un peor caso, en el cual PUNIF es el vencedor por poco. Sin embargo, en los demás experimentos pudimos ver que PUNIF es una pésima elección.

Frente a datos reales, nuestros predictores, MFTPL markoviano y MFTPL probabilista resultan acercarse bastante al comportamiento del mejor experto, donde MFTPL markoviano resulta ser, por poco, la mejor opción. Lo anterior podemos fundamentarlo con el resultado del capítulo *Ambientes* donde mostramos que una cadena de Markov de fallos generaliza a las tasas de fallo, bajo una correcta elección de las probabilidades de transición, trabajo que realiza el predictor MFTPL markoviano en su ejecución. Además, para determinar cuál es el mejor predictor en algún escenario, basta con mirar

los fallos, L_n o R_n (uno sólo), pues las relaciones entre los resultados de estas tres variables siguen las mismas tendencias bajo las distintas medidas.

Como último punto, es sumamente importante mencionar que, si bien hemos dicho que MFTPL markoviano es el mejor predictor seguido del predictor MFTPL probabilista, estos predictores son en realidad una composición de dos algoritmos, por una parte está FTPL, y como segunda componente tenemos la forma de modelar los fallos (que es independiente de FTPL), cuya función es hacer creer al predictor (en este caso FTPL) que su elección debe realizarla sobre un subconjunto de expertos que con alta probabilidad no fallarán. Este algoritmo que hemos agregado delante de FTPL puede ser mezclado con distintos predictores, inclusive PUNIF. Lo importante es que, en presencia de fallos, este algoritmo capta de buena manera el comportamiento de los fallos de cada experto.



Bibliografía

- [1] Venkatachalam Anantharam, Pravin Varaiya y Jean Walrand. «Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part I: IID rewards». En: *IEEE Transactions on Automatic Control* 32.11 (1987), páginas 968-976 (véase página 21).
- [2] Venkatachalam Anantharam, Pravin Varaiya y Jean Walrand. «Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-Part II: Markovian rewards». En: *IEEE Transactions on Automatic Control* 32.11 (1987), páginas 977-982 (véase página 21).
- [3] David Blackwell. «An analog of the minimax theorem for vector payoffs». En: *Pacific Journal of Mathematics* 6.1 (1956), páginas 1-8 (véase página 19).
- [4] Nicolo Cesa-Bianchi y Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006 (véase páginas 13, 16, 31, 36, 44).
- [5] Nicolo Cesa-Bianchi y Gábor Lugosi. «Combinatorial bandits». En: *Journal of Computer and System Sciences* 78.5 (2012), páginas 1404-1422 (véase páginas 20, 21).
- [6] Nicolo Cesa-Bianchi y col. «How to use expert advice». En: *Journal of the ACM (JACM)* 44.3 (1997), páginas 427-485 (véase páginas 13, 16, 19).
- [7] Charles Miller Grinstead y James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012 (véase página 52).
- [8] András György, Tamás Linder y Gábor Lugosi. «Tracking the best of many experts». En: *International Conference on Computational Learning Theory*. Springer. 2005, páginas 204-216 (véase página 20).
- [9] András György, Tamás Linder y György Ottucsák. «The shortest path problem under partial monitoring». En: *Lecture Notes in Computer Science* 4005 (2006), página 468 (véase páginas 20, 21).
- [10] András György y col. «The on-line shortest path problem under partial monitoring». En: *Journal of Machine Learning Research* 8.Oct (2007), páginas 2369-2403 (véase página 20).

- [11] James Hannan. «Approximation to Bayes risk in repeated play». En: *Contributions to the Theory of Games* 3 (1957), páginas 97-139 (véase página 19).
- [12] Adam Kalai y Santosh Vempala. «Efficient algorithms for online decision problems». En: *Journal of Computer and System Sciences* 71.3 (2005), páginas 291-307 (véase páginas 17, 20, 31, 36).
- [13] Nick Littlestone y Manfred K Warmuth. «The weighted majority algorithm». En: *Information and computation* 108.2 (1994), páginas 212-261 (véase página 19).
- [14] Keqin Liu y Qing Zhao. «Adaptive shortest-path routing under unknown and stochastically varying link states». En: *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012 10th International Symposium on*. IEEE. 2012, páginas 232-237 (véase páginas 15, 20).
- [15] Michael Mitzenmacher y Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge university press, 2005 (véase página 23).
- [16] David Pollard. *A user's guide to measure theoretic probability*. Volumen 8. Cambridge University Press, 2002 (véase página 52).
- [17] Eiji Takimoto y Manfred K Warmuth. «Path kernels and multiplicative updates». En: *Journal of Machine Learning Research* 4.Oct (2003), páginas 773-818 (véase páginas 15, 17, 20).
- [18] Pouya Tehrani y Qing Zhao. «Distributed online learning of the shortest path under unknown random edge weights.» En: *ICASSP*. 2013, páginas 3138-3142 (véase página 20).
- [19] Cem Tekin y Mingyan Liu. «Online algorithms for the multi-armed bandit problem with markovian rewards». En: *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE. 2010, páginas 1675-1682 (véase página 21).
- [20] Vladimir Vovk. «Derandomizing stochastic prediction strategies». En: *Machine Learning* 35.3 (1999), páginas 247-282 (véase página 19).
- [21] Zhenhua Zou, Alexandre Proutiere y Mikael Johansson. «Online shortest path routing: The value of information». En: *American Control Conference (ACC), 2014*. IEEE. 2014, páginas 2142-2147 (véase página 20).