UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

MODELAMIENTO DE TRÁFICO EN NODOS DE ACCESO DE BANDA ANCHA

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN INGENIERÍA DE REDES DE COMUNICACIONES

ALBERTO JOSUÉ CASTRO ROJAS

PROFESOR GUÍA:

SR. PATRICIO PARADA SALGADO

MIEMBROS DE LA COMISIÓN:

SR. JUAN PÉREZ RETAMALES

SR. JORGE SILVA SÁNCHEZ

SANTIAGO DE CHILE

JUNIO 2012

Resumen

La planificación de las operaciones de una empresa puede ser entendida como un conjunto de técnicas cuyo objetivo es conseguir el máximo beneficio mediante el uso de insumos y medios utilizados por ella en su proceso productivo. Específicamente, en el contexto de la provisión de servicios de telecomunicaciones se pueden distinguir problemas asociados a la planificación de las redes de telecomunicaciones, entre los cuales se encuentra la evaluación de desempeño de los sistemas de comunicaciones y las proyecciones de tráfico.

Los modelos matemáticos de teorías de colas y de renovación, de identificación y control de sistemas, y de series de tiempo, se utilizan habitualmente para modelar y proyectar tráfico de redes de datos. En el presente trabajo se implementan y comparan distintas metodologías con base en la representación de series de tiempo, con el objetivo de modelar tráfico en nodos de red de acceso a servicios de banda ancha.

En este trabajo se implementan las series de tiempo como filtros de tiempo discreto. Se desarrollan modelos ARMA con representaciones ARMAX y GARCH, que se utilizan en control de sistemas y econometría, respectivamente. Las series de tiempo de tráfico de los nodos de acceso de banda ancha también se modelan con filtros de Kalman de tiempo discreto.

Los resultados obtenidos en el trabajo comprenden la comparación de los modelos ARMAX, GARCH y filtros de Kalman para predicciones de tráfico. Las métricas de desempeño, como errores de predicción y test de hipótesis de correlación, entre la serie estimada y la de verificación, se aplican a los modelos de tráfico de los nodos de red de banda ancha. Los filtros de Kalman, en la mayoría de las comparaciones, presentan mejores parámetros de desempeño que los modelos ARMA, y entre los modelos ARMA, la construcción ARMAX tiene los menores índices de error. En cuanto a los modelos GARCH, en todos los casos, tienen los mayores errores entre las series de validación y de salidas de los modelos.

El aporte de este trabajo es la comparación de distintas metodologías de construcción de modelos de tráfico, con técnicas de distintos dominios que no se habían utilizado en un problema común como modelos para pronósticos de tráfico de redes de datos. De esta forma se implementa una nueva revisión de distintos modelos de series de tiempo en el contexto de planificación de sistemas de comunicaciones, que permite tener distintas alternativas de evaluación de modelos de tráfico, en contraste a las restricciones actuales de evaluación, como construcción de parámetros de desempeño.

Se propone que el trabajo puede ser extendido incorporando filtros adaptivos, además de la representación de series de tiempo que incluyan la dimensión espacial. Así, las series espacio-temporales podrían representarse como filtros adaptivos para disminuir la complejidad de los modelos predictivos.

A veces, y el sueño es triste,
en mis deseos existe
lejanamente un país
donde ser feliz consiste
solamente en ser feliz.
{Fernando Pessoa}

Índice general

Lista de tablas

Lista de figuras

Acrónimos

AIC	Akaike Information Criterion
ACF	AutoCorrelation Function
AR	AutoRegressive
ARI	AutoRegressive Integrated
ARIMA	AutoRegressive Integrated Moving Average
ARMA	AutoRegressive Moving Average
ARMAX	AutoRegressive Moving Average model eXogenous inputs
BIC	Bayesian Information Criterion
BIBO	Bounded-Input Bounded-Output
DSLAM	Digital Subscriber Line Access Multiplexer
FIR	Finite Impulse Response
FARIMA	Fractional ARIMA
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
IIR	Infinite Impulse Response
IMA	Integrated Moving Average
KKT	Karush–Kuhn–Tucker
MA	Moving Average
RLS	Recursive Least-Squares
SARIMA	Seasonal ARIMA
STARMA	Spatio-Temporal ARMA
VAR	Vector AutoRegressive

1. Introducción

1.1. Motivación del trabajo

1.1.1. Planificación y optimización de redes

La planificación de redes de datos, en el contexto de los operadores de telecomunicaciones, considera varios objetivos y restricciones [19]. Entre ellos podemos mencionar:

- Restricciones de presupuesto.
- Redes en servicio de distintas tecnologías y nuevas alternativas técnicas.
- Productos y servicios, actuales o futuros.
- Nuevas demandas, pérdida de clientes y competencia del mercado.
- Estabilidad de las plataformas de redes, es decir, control de tasa de fallas, entre otros aspectos.

Los problemas de optimización pueden formularse, por ejemplo, con el objetivo de maximizar el flujo de tráfico, sujeto a restricciones económicas, técnicas y proyecciones de crecimientos.

Las dificultades para plantear problemas de optimización son múltiples, desde la propia definición del problema, complejidad de resolución, identificación de variables relevantes, simulación de distintos escenarios y redes, etc. En el proceso de planificación los pronósticos de tráfico permiten generar modelos para los problemas de optimización.

El objetivo de este trabajo es desarrollar pronósticos de tráfico de nodos de acceso de banda ancha, y en particular de los nodos DSLAM[1] que permiten acceso hacia redes de datos, como Internet, a través de pares telefónicos. Los procesos y metodologías de proyecciones de tráfico también pueden aplicarse a otras tecnologías de redes y nodos, como por ejemplo, para nodos de accesos de fibra, ya disponibles, como OLT[2], o para tecnologías emergentes como nodos e-nodoB[3], en caso de accesos móviles, además de accesos de banda ancha por medio de redes de cable.

Las proyecciones de tráfico de los DSLAM se encuentran en el contexto de los problemas de planificación de redes por construcción de perfiles de tráfico de los nodos de acceso, que permiten modelar los nodos y variables de interés como capacidades y estimaciones de tráfico.

[1] DSLAM: Digital Subscriber Line Access Multiplexer
[2] OLT: Optical Line Termination
[3] e-nodeB: evolved node B

De esta manera, se busca estimar los nuevos niveles de ocupación que alcanzarán los nodos de acceso, ya sea por nuevas demandas o nuevos servicios, además de apoyar procesos operativos como control de tráfico, fallas y congestión. La anticipación en capacidades requeridas permite holguras de tiempo, en cuanto a procesos de ampliación de los nodos de acceso, con re-localización o adquisición de nuevos recursos, además de análisis de cambios o uso de tecnologías nuevas o complementarias para cursar más tráfico.

1.1.2. Descripción del problema

El tráfico generado por un usuario i, en el instante k y que denotaremos por U_k, es agregado por nodo de acceso de banda ancha DSLAM a la salida del nodo, X_k, se expresa como

$$X_k = \sum_{i}^{n_{U_k}} U_i(k) \tag{1.1}$$

donde n_{U_k} es una variable aleatoria que representa la cantidad de usuarios conectados al nodo de acceso de banda ancha en el instante k.

Este modelo se descompone en: cantidad de usuarios conectados n_{U_k}, y tráfico agregado (ancho de banda agregado respectivo) X_k.

El objetivo del trabajo es la modelación y estimación del tráfico del nodo de acceso de banda ancha, X_k, basado en modelos ARMA y filtros de Kalman.

La tecnología del nodo de acceso de banda ancha, en este caso DSLAM, no limita la descripción del problema y las metodologías para su resolución, que pueden cubrir distintos tipos y tecnologías de nodos de red, de acceso, transporte o core.

1.2. Modelos de tráfico para nodos de acceso de banda ancha

El modelo de tráfico del nodo de acceso de banda ancha que vamos a utilizar es

$$X_{k+m} = A_k X_k + w_k$$

donde A_k es un vector a ser estimado y w_k representa el error de las mediciones de las observaciones de tráfico.

1.2.1. Proyecciones de tráfico

El principal objetivo del trabajo es

Filtrar y proyectar tráfico de nodos de accesos de banda ancha, denotados por X_{k+m}, para distintos intervalos de tiempo, utilizando observaciones de tráfico, X_k, de períodos previos.

1.2.2. Actividades realizadas

Las principales actividades realizadas para encontrar una solución al problema de modelamiento de tráfico en los nodos de acceso de banda ancha, son:

1. Estimación de X_{k+m}, para proyecciones de m pasos, a partir de k. Los pasos de proyección son de un paso, $k = 1$ y de 4 pasos, $k = 4$.
2. Filtrado de X_k y estimación del sistema para k.
3. Comparación de distintos modelos de proyecciones para X_k, con distintas métricas de desempeño.
4. Estimación de métricas y distribución de X_k.

1.3. Modelamiento de tráfico de nodos y redes

El modelamiento de tráfico de nodos y redes usualmente considera el uso de las siguientes metodologías

- Teoría de colas, para análisis de capacidades de procesamiento de los nodos [40].
- Distribuciones estadísticas de uso de los recursos, como Poisson (tiempo discreto), exponencial y normal [35].
- Métricas de distintas tecnologías, como pérdida de paquetes IP, delays en redes, etc., que se modelan con distintas distribuciones [15].
- Distintos modelos heurísticos para tráfico, por ejemplo, para tráfico en Internet.
- Modelos de nodos, dependiendo de la tecnología de acceso [10].

Los modelos de los nodos de acceso de banda ancha normalmente describen a la tecnología de acceso, y sobre estos modelos se usan las variables de distintas distribuciones estadísticas, para realizar perfiles de tráfico, análisis de capacidades y proyecciones.

1.3.1. Simulación de redes

La aproximación a modelos de nodos de redes vía simulación [30][38], es común, y presupone modelos a priori, tanto de uso de capacidades de los nodos como estructura de los mismos, con dependencia de las distintas tecnologías de comunicaciones, tanto de acceso, como de transporte.

La simulación de redes [1] se enfoca en la distribución de nodos, en las características de los nodos, y en su interacción, y con menor enfásis en la caracterización de nodos vía estimación de parámetros, a partir de observaciones de tráfico. Ello tampoco contempla análisis de series de observaciones para realización de proyecciones de tráfico.

1.4. Series de tiempo discreto

En este trabajo se utilizan metodologías generales de modelación de series de observaciones, llamadas series de tiempo, con tráfico de los nodos de acceso a redes de banda ancha.

Las series de tiempo son secuencias de observaciones a intervalos de tiempo regulares, periódicos o irregulares. Ejemplos de series de tiempo se encuentran en diversas disciplinas como economía, ciencias naturales, comunicaciones, etc. La figura 1.1 muestra las series de tiempo de la evolución de cantidad de usuarios de banda ancha móvil y fija, en Chile, con muestras mensuales de los años 2010 y 2011.

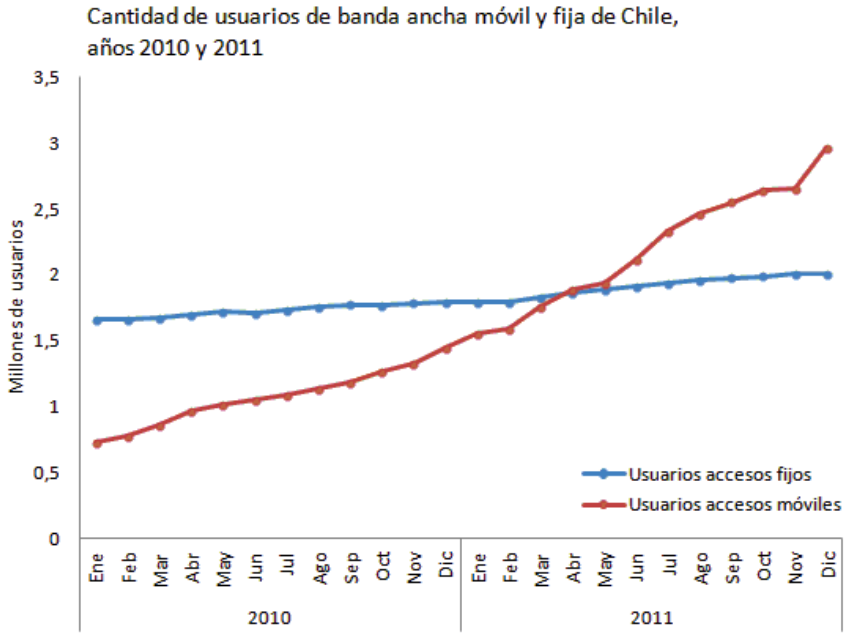


Figura 1.1.: Ejemplo de series de tiempo de cantidad de usuarios de banda ancha de accesos móviles y fijos entre Enero de 2010 y Diciembre de 201. Fuente Subtel [24].

Las series de tiempo discreto se generan por muestras de procesos aleatorios continuos o por agregación de los valores de las observaciones de manera periódica.

El análisis de las series de tiempo se realiza con distintos modelos estadísticos. Si las observaciones son dependientes de las muestras previas, entonces pueden hacerse pronósticos de observaciones futuras mediante uso de las muestras pasadas. Habitualmente, las series de tiempo son secuencias estocásticas de valores y las distribuciones de probabilidad de las observaciones permiten desarrollar modelos condicionados a la información disponible.

1.4.1. Objetivos del análisis de las series de tiempo

Los objetivos del análisis de las series de tiempo son diversos. Los principales son:

- Descripción: vía gráficos de las muestras versus tiempo, para observación de tendencias, estacionalidad, muestras fuera de rango, muestras sin información, etc.
- Dependencias: entre una o más variables que expliquen las variaciones en las series. Los modelos de regresión, de correlación, de sistemas lineales, etc., intentan describir las relaciones entre las variables para modelamiento de las series de tiempo.
- Predicción: de los valores futuros de las series con las observaciones previas. Los modelos predictivos pueden emplearse en diversas áreas, por ejemplo, econometría, comunicaciones, señales, etc.
- Control: de las variables que describen las series de tiempo, para control de sistemas. Habitualmente la predicción se usa como control, para mantener las variables dentro de rangos de operación de los sistemas con los valores previos.

1.4.2. Alternativas de análisis de series de tiempo

Existen varias alternativas de análisis de las series de tiempo. Las técnicas básicas son descriptivas y de autocorrelación.

Los análisis en los dominios del tiempo y de la frecuencia se suman a diversos modelos como sistemas lineales, modelos no lineales y multivariados. Además de modelos de espacio estado y filtros de Kalman.

Las alternativas de análisis de las series de tiempo, empleadas en el trabajo, son:

- Técnicas descriptivas.
- Análisis en el dominio del tiempo.
- Sistemas lineales.
- Modelos de espacio de estado.
- Filtros de Kalman [31][12].

En este trabajo, los modelos de tráfico de los nodos de red se construyen con los siguientes modelos de series de tiempo:

- ARMA: AutoRegressive Moving Average, modelo autoregresivo de media móvil.
  - ARIMA: AutoRegressive Integrated Moving Average, modelo autoregresivo integrado de media móvil.
  - ARMAX: AutoRegressive Moving Average model eXogenous inputs, modelo autoregresivo de media móvil con entradas exógenas.
- GARCH: Generalized AutoRegressive Conditional Heteroskedasticity, modelos generalizados autoregresivos de heterocedasticidad condicional.
- Modelos de espacio de estado y filtros de Kalman.

Las series de tiempo se analizan con los modelos ARIMA [37], las funciones de autocorrelación y autocorrelación parcial. En este trabajo también se emplean los modelos GARCH y filtros de Kalman para evaluar los desempeños de las proyecciones de tráfico con distintas métricas de comparación, entre los modelos, como errores de predicción, correlaciones y test de hipótesis de similitud entre serie de validación y serie estimada.

Existe numerosa literatura de los filtros de Kalman y modelos de espacio de estado [39]. Sus aplicaciones en el contexto de comunicaciones y redes son variadas, como por ejemplo: la determinación de la posición de los terminales, tanto en redes WLAN, como móviles [16], con uso masivo en terminales con Global Positioning System (GPS) para geo-referenciación [28], entre otras alternativas.

1.5. Trabajos relacionados

El desempeño de las proyecciones de tráfico debe ser evaluado en distintos intervalos de tiempo. Los modelos de tráfico tienen que construirse para ser válidos en distintos escenarios de evaluación y de planificación de redes.

El método de regresión lineal es útil cuando no hay demasiada variación en las series de observaciones de tráfico [3]. Las predicciones de tráfico pueden mejorar por la combinación de modelos y metodologías como diferenciación de las observaciones, series de tiempo y filtros de tiempo discreto.

El pronóstico del valor de una observación, $\widehat{X}_{k+m}$, se realiza a partir de la distribución condicional de X_{k+m} dados $X_1, \ldots, X_k$, con $m > k > 0$. De esta forma el pronóstico del valor de una observación, en m pasos desde k, es

$$\widehat{X}_k(m) = \mathbb{E}(X_{k+m} \,|\, X_1 = x_1, \ldots, X_k = x_k)$$

y un intervalo de predicción [A,B], con umbral de confianza $1 - \alpha$, tal que

$$\mathbb{P}(A \leq X_{k+m} \leq B \,|\, X_1 = x_1, \ldots, X_k = x_k) = 1 - \alpha$$

El problema de proyección consiste en encontrar un buen modelo y calcular las distribuciones condicionales mencionadas. Las series de tiempo, filtros de tiempo discreto y filtros de Kalman utilizados en este trabajo, son alternativas para resolver este problema.

Los modelos y predicciones para tráfico de redes de datos se pueden clasificar en:

- Modelos a partir de series de tiempo ARMA/ARIMA y variaciones asociadas, como por ejemplo modelos FARIMA (ARIMA Fraccional) [20], para predicciones de tráfico en redes de datos, en distintos accesos, como WLAN [6], WIMAX y xDSL.
- Modelos de series de tiempo entrenados con redes neuronales [18, 29].
- Aplicación de filtros de Kalman a modelos de tráfico estadísticos [26].

Los filtros de Kalman también son utilizados para proyecciones de tráfico de redes de datos, en tiempo real [13], dado el carácter dinámico del filtro [45]. En la caracterización de enlaces y tráfico agregado también se han utilizado filtros de Kalman [26].

A continuación se describen las metodologías de series de tiempo aplicadas a modelos de tráfico de redes de datos.

1.5.1. Series de tiempo ARMA y modelos de tráfico de redes de datos

Los trabajos que utilizan los modelos ARMA se focalizan en la obtención de los coeficientes y grados que especifican a la serie de tiempo que representa a los datos de tráfico [27].

Los modelos ARMA se describen en forma polinomial, en dominio Z, y se validan con datos de tráfico asociadas a las observaciones de construcción de los modelos. También se utilizan ecuaciones de diferencias.

En la construcción de los modelos ARMA se utilizan diversos criterios de selección del orden, como AIC [44] o BIC (descritos en capítulo 2) para ajustar los coeficientes de los polinomios, buscando minimizar los errores de representación de la serie de observaciones de tráfico. Esta aproximación es similar al ajuste, por diversos métodos, de una curva que representa a la serie de datos. Entre estos métodos de ajuste más empleados se encuentran:

1. Regresión lineal, multi-regresión lineal [32].
2. Series de tiempo basadas en series de Fourier, ajuste de los coeficientes de las series de tiempo por las expresiones de los coeficientes de las series de Fourier [2].
3. Identificación de sistemas dinámicos a través de modelos de espacio de estado. En este caso se utiliza analogía entre las series de tiempo y los sistemas dinámicos de espacio de estado.

1.5.2. Ejemplo de uso de series de tiempo en redes de datos

Para las redes de datos de accesos inalámbricos con tecnologías como WLAN, WIMAX y Móviles (3G/2G) se utilizan las series de tiempo para las siguientes aplicaciones:

- Modelos de tráfico de corto plazo, para proyecciones.
- Métricas para análisis de las redes y tráficos.

A continuación se describe un ejemplo de aplicación de series de tiempo en redes WIMAX.

Modelos ARMA para tráfico en redes WIMAX

En [7] se usan los modelos de series de tiempo ARMA y ARIMA para modelar tráfico de redes WIMAX.

Setup

Los datos son obtenidos desde la herramienta de software de medición Netflow Analyzer [25].

El intervalo de recolección de los datos es de 9 días, con medición de las observaciones cada 15 minutos, es decir 864 muestras. Se utilizaron los 7 primeros días para la construcción de los modelos, dejando los restantes para validación.

Identificación del modelo

El modelo no es estacionario, lo cual se verifica mediante el cálculo de promedios no nulos. Por ello, se señala que se trabaja con ARIMA y no se utilizan los modelos AR, MA o ARMA, que necesitan que las series sean estacionarias en el sentido amplio.

Modelo ARIMA

Se evaluan modelos ARIMA con la herramienta de software Regression Analysis of Time Series (RATS) [23].

Para construir los modelos, primero se diferencia la serie una vez, y posteriormente se prueban distintos órdenes de modelos (p,q). El orden del modelo se estima a tráves de las funciones de autocorrelación y correlación parcial.

La búsqueda de los coeficientes del modelo se realiza con RATS [23]. La comprobación preliminar de los modelos se realiza con las funciones de autocorrelación y correlación parcial de los residuos, lo que determina nuevas iteraciones, para encontrar primero el orden (p,q) y después los coeficientes, por entontrarse en la primera relación, correlación entre el modelo de residuos y la serie. Las iteraciones posteriores son 3.

El resultado es un modelo de orden 18 con 6 parámetros.

Modelos ARI e IMA

El desarrollo de los modelos Autoregresivo Integrado (ARI) y Media Móvil Integrado (IMA) se realiza del mismo modo que ARIMA.

Para el modelo ARI, el orden es 22, y para IMA (MA(I)) no se indica.

Comparación de modelos y conclusiones

Los modelos se comparan utilizando las siguientes métricas de comparación

- Criterio de parsimonia: modelo más simple, en cuanto a orden y coeficientes, es el seleccionado.
- Error cuadrático medio.
- Valor absoluto de la desviación estándar.
- Coeficiente de correlación.

La conclusión del reporte es la elección del modelo ARIMA, de un error de 6 % evaluado con los datos de verificación. Finalmente, la sugerencia de los autores de este trabajo [7], es realizar análisis para intervalos de tiempo mayores, con propuesta de un año, o más, además de comparación con otros modelos como SARIMA y FARIMA.

1.6. Organización del documento

Este documento esta dividido en 5 capítulos, cuyos contenidos son los siguientes.

En el capítulo 2, marco teórico, se describen los modelos ARMA en los formatos ARIMA, ARMAX y GARCH. Las series de tiempo también se desarrollan como procesos estocásticos, para continuar con la representación de filtros digitales. Se describen los modelos de espacio de estado y filtros de Kalman, además de criterios para estimación de los grados de los modelos ARMA. Finalmente se enumeran ejemplos de aplicación de los modelos ARMA, además de filtros de Kalman, para proyecciones de tráfico, entre otras aplicaciones.

El capítulo 3, de desarrollo, comienza con las metodologías de optimización de los modelos ARMA, para minimizar errores de predicción, con uso de mínimos cuadrados y condiciones de Kuhn–Tucker. Se enumeran las distintas alternativas de implementación de los modelos ARMA, en formatos ARMAX y GARCH. La revisión de estabilidad de los modelos ARMA para un nodo de acceso de banda ancha de ejemplo, permite la introducción de filtros de tiempo discreto. El capítulo finaliza con la presentación de proyecciones, de m pasos, basadas en los modelos ARMA y filtro de Kalman, para un nodo de ejemplo.

La sección de resultados y discusión, capítulo 4, se inicia con la descripción de las distintas métricas de errores de predicción y test de hipótesis de correlación, test de Kolmogorov-Smirnov, entre las series de predicción de m pasos y de verificación. Las discusiones se basan en las comparaciones de desempeño, para proyecciones, de los modelos ARMA y filtro de Kalman. Finalmente se describen las propuestas de extensiones del trabajo.

Las conclusiones se desarrollan para: resultados y aportes de este trabajo, implementación de los modelos y alternativas de próximos trabajos utilizando filtros adaptivos y series espacio-temporales.

En los anexos se listan los distintos códigos de implementación de los modelos ARMA y filtro de Kalman. Las expresiones de los modelos ARMA, para 4 nodos de red de acceso de banda ancha, se introducen junto a los gráficos para las proyecciones, tanto de los modelos ARMA y filtro de Kalman. La sección de anexos finaliza con el desarrollo de complementos para los capítulos de desarrollo y resultados.

2. Marco Teórico

En este capítulo se describen los modelos ARMA y filtros de Kalman en el contexto de sistemas de tiempo discreto. Los modelos se desarrollan como procesos estocásticos que también se representan como filtros de tiempo discreto. Los modelos ARMA, filtros de tiempo discreto y filtros de Kalman se utilizan para la representación de series de tiempo discreto.

Las relaciones entre procesos estocásticos y series de tiempo permiten utilizar distintos parámetros de desempeño para evaluar los modelos y sus proyecciones.

2.1. Modelos de procesos estocásticos de tiempo discreto

Un proceso o señal estocástica $x[k_n]$ es un conjunto de variables aleatorias $\{x_{k_1}, x_{k_2}, \ldots, x_{k_n}\}$ de n elementos. La repetición de las observaciones entrega nuevos valores para los elementos de las secuencia que son distintos de valores previos, y así las observaciones pueden modelarse como un proceso estocástico aleatorio.

La secuencia $\{k_n\}$ de una señal pueden caracterizarse por distintas variables aleatorias con diferentes densidades de probabilidad.

La media en el instante n es

$$\mu_n = \mathbb{E}[X(n)] = \int_{-\infty}^{\infty} x \cdot f_n(x) dx$$

y su varianza es

$$\sigma_n^2 = \mathbb{E}\left\{[X(n) - \mu_n]^2\right\} = \int_{-\infty}^{\infty} [x - \mu_n]^2 \cdot f_n(x) dx$$

donde $f_n(x)$ es la densidad de probabilidad de $x[k_n]$.

La media y varianza de las secuencias de señales $x[k_n]$ pueden cambiar en cada instante n. En el caso que las densidades de probabilidad sean iguales $\forall$ n, las series de tiempo se denominan estacionarias, concepto que revisaremos a continuación.

2.1.1. Series de tiempo estacionarias

Las series de tiempo pueden ser descritas como procesos aleatorios estacionarios si para distintos intervalos de tiempo las densidades de probabilidad son idénticas. [34, cap.1].

Proceso estrictamente estacionario

Una serie de tiempo es estrictamente estacionaria si las probabilidades del conjunto de valores

$$\{x_{k_1}, x_{k_2}, \ldots, x_{k_n}\}$$

son iguales a las probabilidades de conjunto desplazado en τ

$$\{x_{k_{1+\tau}}, x_{k_{2+\tau}}, \ldots, x_{k_{n+\tau}}\}$$

es decir

$$\mathbb{P}\{x_{k_1} \leq c_1, \ldots, x_{k_n} \leq c_n\} = \mathbb{P}\{x_{k_{1+\tau}} \leq c_1, \ldots, x_{k_{n+\tau}} \leq c_n\}$$

para todo $n = 1,2,\ldots$, todos los tiempos $k_1, k_2, \ldots, k_n$, todas las constantes $c_1, c_2, \ldots, c_n$, y todos los desplazamientos $\tau = 0, \pm 1, \pm 2, \ldots$.

En una serie estrictamente estacionaria todas las funciones de distribución multivariada deben tener correspondientes distribuciones multivaridadas desplazadas para todos los valores τ.

Así, por ejemplo, cuando $n = 1$, se tiene

$$\mathbb{P}\{x_s \leq c\} = \mathbb{P}\{x_k \leq c\}$$

para cualquier par de puntos s y k. Además, si la media μ_k de la serie x_k existe, entonces $\mu_s = \mu_k$ para todo s y k, y μ_k, es constante.

Cuando $n = 2$, se tiene

$$\mathbb{P}\{x_s \leq c_1, x_k \leq c_2\} = \mathbb{P}\{x_{s+\tau} \leq c_1, x_{k+\tau} \leq c_2\}$$

para cualquier par de puntos s y k, y desplazamiento τ. Si la varianza γ de la serie existe, entonces la función de autocovarianza de la serie x_k cumple con

$$\gamma(s, k) = \gamma(s + \tau, k + \tau)$$

para todo k y τ, es decir, la función de autocovarianza sólo depende de la diferencia entre k y τ, y no de su valor individual.

La definición de una serie de tiempo estrictamente estacionaria no es ampliamente utilizada, por las dificultades de aplicación a series de observaciones y muestras de datos acotadas. Las imposiciones de las serie de tiempo estacionaria a todos los momentos de la serie de datos se flexibiliza, generalmente, a sólo los primeros momentos de la serie, es decir, media y varianza.

Proceso estacionario débil o estacionario en sentido amplio

Una serie de tiempo x_k es estacionaria débil, si es un proceso de varianza finita, tal que

1. La media, μ_k, es constante, es decir no depende del tiempo k.
2. La función de autocovarianza $\gamma(s,k)$, sólo depende de s y k, a través de su diferencia $|s-k|$.

Las funciones media y autocovarianza se definen en la siguiente sección [2.1.2].

Nota: los procesos estacionarios débiles se denotan habitualmente como procesos estacionarios. Los procesos estrictamente estacionarios se mencionan como tales.

Procesos conjuntamente estacionarios

Dos series x_k e y_k son conjuntamente estacionarias, si cada serie es estacionaria, y la función de covarianza cruzada

$$\gamma_{xy}(\tau) = cov(x_{k+\tau}, y_k) = \mathbb{E}\{(x_{k+\tau} - \mu_x)(y_k - \mu_y)\}$$

es sólo función del desplazamiento de tiempo o lag τ .

2.1.2. Medidas de procesos estacionarios

Las series de tiempo son secuencias de variables aleatorias $x[k_n]$ en instantes de tiempo $k_1, k_2, \ldots, k_n$ para un entero positivo n. La distribución conjunta de la serie para que sea menor que n constantes, $c_1, c_2, \ldots, c_n$, es

$$F(c_1, c_2, \ldots, c_n) = \mathbb{P}(x_{k_1} \leq c_1,\, x_{k_2} \leq c_2\,, \ldots,\, x_{k_n} \leq c_n) \tag{2.1}$$

F es una función de distribución multivariada, lo que dificulta su análisis, que sólo se simplifica en el caso de distribuciones normales conjuntas. Por ello, aunque F describe a la serie completa, no se utiliza como descripción de la serie. La distribución 2.1 podría evaluarse en sus n argumentos, para las distribuciones marginales

$$F_k(x) = \mathbb{P}\{x_k \leq x\}$$

cuya densidad marginal es

$$f_k(x) = \frac{\partial F_k(x)}{\partial x}$$

cuando la derivada parcial existe.

Las medidas de distribuciones de funciones de probabilidad, para procesos estacionarios, son las siguientes [42, cap.2]

Media de la variable x_k

$$\mu = \mathbb{E}\{x_k\}$$

donde $\mathbb{E}$ es el valor esperado. Dado que la función $\mathbb{E}\{x_k\}$ es independiente del tiempo, la notación puede simplificarse a $\mu = \overline{x}$, el cual es un estimador insesgado

$$\widehat{\mu}_x = \frac{1}{n}\sum_{k=1}^{n} x_k$$

Función de autocovarianza

La función de autocovarianza se define por el segundo momento de F, esto es

$$\gamma_x(s,k) = cov(x_s, x_k) = \mathbb{E}[(x_s - \mu_s)(x_k - \mu_k)]$$

para todo s y k. Se denotará $\gamma_x(s,k)$ sólo por $\gamma(s,k)$. La autocovarianza indica la dependencia lineal entre dos puntos de la misma serie, en distintos instantes. Series con poca variabilidad tienen funciones de autocovarianza estables, aún para s y k distantes.

La autocovarianza es el producto cruzado promedio de la distribución $F(x_s, x_k)$.

Si $\gamma_x(s,k) = 0$ para $s \neq k$ entonces x_s y x_k son no correlacionados, pero aún pueden tener alguna estructura de dependencia estadística. En el caso que x_s y x_k son normales bivariadas, la condición $\gamma_x(s,k) = 0$ para $s \neq k$ es suficiente para asegurar la independencia de los procesos.

Para $s = k$ la autocovarianza es

$$\gamma_x(k,k) = \mathbb{E}[(x_k - \mu_k)^2] = var(x_k)$$

Las medidas recién señaladas, media y autocovarianza, pueden normalizarse en el intervalo [-1,1].

Autocovarianza de series estacionarias

Como la función de autocovarianza $\gamma(s,k)$ de la serie x_k sólo depende de la diferencia $|s-k|$, la notación puede simplificarse definiendo $s = k + \tau$ donde τ representa el desplazamiento de tiempo o 'lag'. Luego

$$\gamma(k+\tau, k) = cov(x_{k+\tau}, x_k) = cov(x_\tau, x_0) = \gamma(\tau, 0) \equiv \gamma(\tau)$$

porque la diferencia entre $k+\tau$ y k, es la misma que la diferencia entre τ y 0. La función de autocovarianza de una serie estacionaria x_k no depende de k. Por conveniencia, es posible omitir el segundo argumento de $\gamma(\tau, 0)$, por lo que la función de autocovarianza se expresa por

$$\gamma(\tau) \;=\; cov(x,\tau) = \mathbb{E}\left\{(x_k - \overline{x})(x_{k+\tau} - \overline{x})\right\} = \mathbb{E}\left\{x_k x_{k+\tau}\right\} - \overline{x}^2$$

con estimador

$$\widehat{\gamma}(\tau) = \frac{1}{n-\tau}\sum_{k=1}^{n-\tau}(x_{k+\tau} - \overline{x})(x_k - \overline{x})$$

Cuando la función de autocovarianza se evalua para $\tau = 0$, se obtiene la varianza de la serie

$$\gamma(0) = \sigma_x^2 = \mathbb{E}\{(x_k - \overline{x})^2\}$$

Varianza o media cuadrática
Corresponde a

$$\sigma_x^2 = \mathbb{E}\{(x_k - \overline{x})^2\}$$

y su estimador es

$$\widehat{\sigma}_x^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \overline{x})^2$$

Función de autocorrelación (ACF)

$$R_x(\tau) = \mathbb{E}\{x_k \cdot x_{k+\tau}\}$$

La función de autocorrelación ACF, en términos de la función de autocovarianza se define como

$$R(s,k) = \frac{\gamma(s,k)}{\sqrt{\gamma(s,s)\,\gamma(k,k)}}$$

ACF relaciona linealmente la predicción de x_k con x_s en el tiempo k. $R(s,k) \in [-1,1]$, debido a $|\gamma(s,k)|^2 \leq \gamma(s,s)\gamma(k,k)$.

La relación de predicción de x_k desde x_s es $x_k = \beta_0 + \beta_1 x_s$, la correlación es +1 cuando $\beta_1 > 0$, y -1 cuando $\beta_1 < 0$.

La función de autocorrelación puede simplificarse a

$$R(\tau) = \frac{\gamma(\tau)}{\gamma(0)}$$

Función de correlación cruzada (CCF)

$$R_{xy}(\tau) = \frac{\gamma_{xy}(\tau)}{\sqrt{\gamma_x(0)\gamma_y(0)}} = \mathbb{E}\{x_k \cdot y_{k+\tau}\} = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n} x_k \cdot y_{k+\tau} = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n} x_{k-\tau} \cdot y_k$$

Para esta función, se tiene también que $-1 \leq R_{xy}(\tau) \leq 1$.

Ejemplo con series simuladas

Como ejemplo de evaluación numérica de las funciones (aproximaciones) de autocovarianza y covarianza cruzada, consideremos los siguientes conjuntos de datos

k	1	2	3	4	5	6	7	8	9	10
moneda	C	C	S	C	S	S	S	C	S	C
x_k	1	1	-1	1	-1	-1	-1	1	-1	1
y_k	6,7	5,3	3,3	6,7	3,3	4,7	4,7	6,7	3,3	6,7
$y_k - \overline{y}$	1,56	0,16	-1,84	1,56	-1,84	-0,44	-0,44	1,56	-1,84	1,56

Tabla 2.1.: Tabla de datos simulados de ejemplo

generados por "lanzamiento" (simulación) de una moneda, con C = cara = 1 y S = sello = -1, para los valores de x_k. Para y_k, se define el proceso:

$$y_k = 5 + x_k - 0,7 \cdot x_{k-1}$$

La tabla de datos, se genera con $x_0 = -1$ y n = 10.

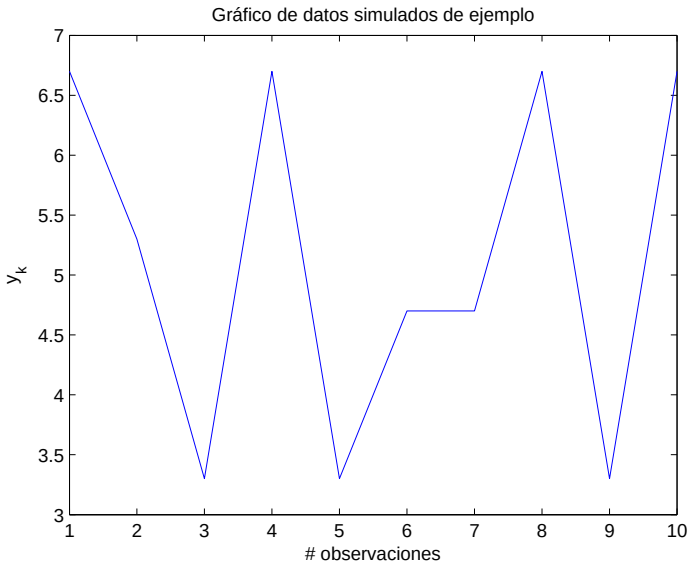


Figura 2.1.: Datos simulados de ejemplo

La función de autocovarianza para y_k, para $\tau = 0, 1, 2, \ldots$, se calcula a partir de $[xy]$, por ejemplo, para $\tau = 3$, la aproximación de la autocovarianza es

$$\begin{aligned}
\widehat{\gamma}_y(3) &= \frac{1}{10}\sum_k (y_{k+3} - \overline{y})(y_k - \overline{y}) \\
&= \frac{1}{10}[(1,56)(1,56) + (-1,84)(0,16) + (-0,44)(-1,84) + (-0,44)(1,56) \\
&= +(1,56)(-1,84) + (-1,84)(-0,44) + (1,56)(-0,44)] = -0,048
\end{aligned}$$

con

$$\widehat{\gamma}_y(0) = \frac{1}{10}\left[(1,56)^2 + (0,16)^2 + \cdots + (1,56)^2\right] = 2,030$$

entonces

$$\widehat{R}_y(3) = \frac{-0,048}{2,030} = -0,024$$

Usando el hecho de que la media de x_k es 0, y la varianza es 1, se puede mostrar que

$$R_y(1) = \frac{-0,7}{1+0,7^2} = -0,47$$

y que $R_y(\tau) = 0$, para $|\tau| > 1$. La siguiente tabla compara los valores de las función ACF, formales y aproximaciones, para n = 10 y n = 100; notar la mayor variabilidad de los valores para las muestras del conjunto menor, comparados al conjunto mayor.

		n = 10	n = 100
τ	$R_y(\tau)$	$\widehat{R}_y(\tau)$	$\widehat{R}_y(\tau)$
0	1	1	1
±1	-0,47	-0,55	-0,45
±2	0	0,17	-0,12
±3	0	-0,02	0,14
±4	0	0,15	0,01
±5	0	-0,46	-0,01

Tabla 2.2.: Comparación de valores de función ACF, teóricos y aproximaciones

2.1.3. Densidad espectral de potencia

La densidad espectral de potencia de un proceso estacionario es la transformada de Fourier de su función de autocorrelación

$$S^*_{xx}(iw) = \Im\{R_{kk}(\tau)\} = \sum_{\tau=-\infty}^{\infty} R_{kk}(\tau)e^{-j\tau wT_0}$$

En forma alterna, usando transformada Z

$$S_{xx}(z) = \varrho\{R_{xx}(\tau)\} = \sum_{\tau=-\infty}^{\infty} R_{kk}(\tau)\cdot z^{-\tau}$$

2.2. Filtros de tiempo discreto

Los filtros de tiempo discreto pueden especificarse tanto en el dominio del tiempo como en el de la frecuencia. En el dominio de tiempo discreto, los tipos de filtros son aproximaciones recursivas de polinomios o de repuesta infinita al impulso (infinite impulse response IIR) y filtros no recursivos o de respuesta finita al impulso (finite impulse response FIR) que resultan de aproximaciones polinomiales.

Los filtros IIR se especifican en magnitud y fase en el dominio de la frecuencia, mientras que los filtros FIR se especifican con la respuesta al impulso en el dominio del tiempo. El problema de diseño de filtros discretos es la construcción de polinomios o fracciones polinomiales (numerador y denominador con polinomios), con órdenes y coeficientes, que al evaluarse sean estables y causales.

2.2.1. Filtros IIR y FIR

Filtro IIR

Un filtro discreto de función de transferencia

$$H(z) = \frac{Q(z)}{P(z)} = \frac{\sum_{m=0}^{M-1} q_m \cdot z^{-m}}{1 + \sum_{k=1}^{N-1} p_k \cdot z^{-k}} = \sum_{n=0}^{\infty} h[n] \cdot z^{-n}$$

es un filtro IIR si la respuesta al impulso $h[n]$ es de longitud infinita y recursivo porque la entrada del filtro $H(z)$ es $x[n]$ y la salida es $y[n]$, con la siguiente relación recursiva de entrada-salida

$$y[n] = -\sum_{k=1}^{N-1} p_k \cdot y[n-k] + \sum_{m=0}^{M-1} q_m \cdot x[n-m]$$

Filtro FIR

La función de transferencia de un filtro FIR es

$$H(z) = Q(z) = \sum_{m=0}^{M-1} q_m \cdot z^{-m}$$

La respuesta al impulso es $h[n] = q_n$, $n = 0, \ldots, M-1$, y cero para cualquier otro valor, es decir tiene una longitud finita. El filtro no es recursivo de relación entrada-salida

$$y[n] = \sum_{m=0}^{M} q_m \cdot x[n-m] = (q * x)[n]$$

que también se expresa por la convolución de los coeficientes del filtro (o respuesta al impulso) y la entrada.

2.2.2. Estabilidad BIBO

La estabilidad Bounded-Input Bounded-Output (BIBO) establece que para una entrada acotada $x[n]$, la salida $y[n]$ del filtro también es acotada. Si la entrada esta acotada por $M < \infty$ tal que $|x[n]| < M \ \forall \, n$ la salida también esta acotada, es decir $|y[n]| < L \ \forall \, n$ con $L < \infty$.

Un sistema de tiempo discreto es estable BIBO si la respuesta al impulso $h[n]$ del sistema es finita

$$\sum_{k} |h[k]| < \infty$$

Como la entrada $x[n]$ esta acotada por $M < \infty$ tal que $|x[n]| < M \ \forall \, n$, la salida $y[n]$ puede expresarse como una convolución que también esta acotada por

$$|y[n]| \leq \left| \sum_{k=-\infty}^{\infty} x[n-k] \cdot h[k] \right| \leq \sum_{k=-\infty}^{\infty} |x[n-k]| \cdot |h[k]|$$

$$\leq M \sum_{k=-\infty}^{\infty} |h[k]| \leq M \cdot N < \infty$$

dado que la repuesta al impulso $h[n]$ está siempre acotada.

2.3. Representación de series de tiempo por filtros discretos

2.3.1. Resumen de modelos, ecuaciones de diferencia estocásticas

Las ecuaciones de diferencia estocásticas son de la forma [42]

$$y_k + c_1 y_{k-1} + \ldots + c_n y_{k-n} = d_0 x_k + d_1 x_{k-1} + \ldots + d_m x_{k-m}$$

con y_k salida del filtro. Aplicando la transformada Z tenemos

$$G_F(z^{-1}) = \frac{y(z)}{x(z)} = \frac{d_0 + d_1 z^{-1} + \ldots + d_m z^{-m}}{1 + c_1 z^{-1} + \ldots + c_n z^{-n}} = \frac{D(z^{-1})}{C(z^{-1})}$$

donde x_k es una señal independiente de y_k.

La representación de distintos procesos estocásticos puede efectuarse como un proceso de una función de tiempo discreto de una señal estocástica, teniéndose como casos particulares a las series de tiempo que se representan en la figura 2.2.

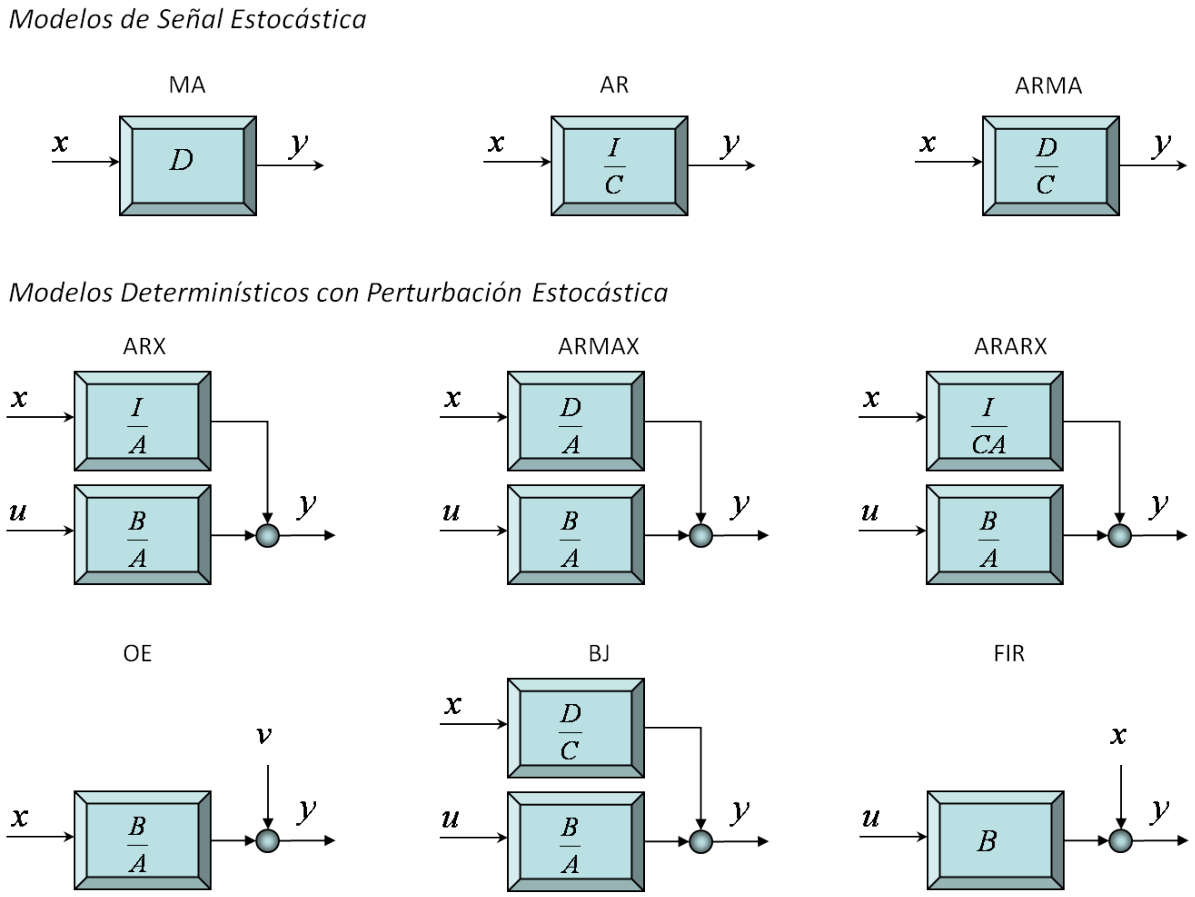


Figura 2.2.: Representación de procesos estocásticos en diagramas de bloques de operaciones fundamentales

Leyenda de figura 2.2

Abreviación	Descripción
MA	Moving Average
AR	AutoRegressive
ARMA	AutoRegressive Moving Average
ARX	AutoRegressive eXogenous inputs
ARMAX	AutoRegressive Moving Average model eXogenous inputs
ARARX	AutoRegressive AutoRegressive eXogenous inputs
OE	Output Error
BJ	Box–Jenkins
FIR	Finite Impulse Response

Las secuencias de observaciones $x[k_n]$ se denotan por X, teniéndose

- X_k = secuencia de n elementos.
- S = conjunto de secuencias.

2.3.2. Operador backshift

Si el largo de la X secuencia, de n elementos, se denota por *l(X)*, se tiene

- $l(BX) = l(X)$
- $(BX)_1 = 0$
- $(BX)_k = X_{k-1} \quad t = 2,...,l(X)$

La representación de X como vector permite representar al operador backshift B como

$$B \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix} = \begin{pmatrix} 0 \\ X_1 \\ \vdots \\ X_{n-1} \end{pmatrix} \quad \text{con } l(X) = n$$

Si $l(X) \leq n$, la forma matricial del operador backshift B es

$$BX = B_n X$$

donde B_n es una matrix $n \times n$

$$B_n = \begin{pmatrix} 0 & 0 & 0 & \dots & \\ 1 & 0 & 0 & \dots & \\ 0 & 1 & 0 & \dots & \\ \vdots & \vdots & \vdots & \ddots & \\ & & & 0 & 0 \\ & & & 1 & 0 \end{pmatrix}$$

Si $n = l(X)$, y se aplica el operador backshift B n veces, se tiene $(B_n)^n = 0$.

Además, se tiene

$$\frac{1}{\sum_{n=0}^{\infty} B^n} = 1 - B \quad \text{y} \quad \frac{1}{1-B} = \sum_{n=0}^{\infty} B^n$$

2.3.3. Propiedades de los filtros y respuesta al impulso

Los filtros tienen las siguientes propiedades [3]

- Una secuencia de longitud n se mapea a otras secuencias de igual longitud, dentro de S (conjunto de secuencias).

- Existe una secuencia infinita de números h_i, $i = 0, 1, 2, \ldots$, llamada respuesta al impulso, tal que, para todo $X \in S$, se tiene

$$(FX)_k = h_0 X_k + h_1 X_{k-1} + \cdots + h_{k-1} X_1 \qquad k = 1, \ldots, l(X) \tag{2.2}$$

La expresión 2.2 puede también describirse como

$$F = \sum_{i=0}^{\infty} h_i \cdot B^i \tag{2.3}$$

Para una secuencia $X \in S$, de largo n, entonces $F \cdot X = \sum_{i=0}^{n-1} h_i \cdot B^i \cdot X$.

Un filtro tiene ***respuesta al impulso finita*** (FIR), si $h_n = 0$ para todo $n \geq N \in \mathbb{N}$, en caso contrario se denota como de ***respuesta al impulso infinita*** (IIR).

2.3.4. Representación de $AR(\infty)$ como filtro con inversa

Sea F un filtro con inversa e $Y = F \cdot X$, y $g_0, g_1, \ldots$ la respuesta al impulso de F^{-1}. Entonces $X = F^{-1} \cdot Y$ para $k \geq 1$ con

$$X_k = g_0 Y_k + g_1 Y_{k-1} + \cdots + g_{k-1} Y_1 \tag{2.4}$$

que puede expresarse en términos de Y_k

$$Y_k = A_0 X_k + A_1 Y_{k-1} + \cdots + A_{k-1} Y_1 \tag{2.5}$$

$$A_k = \begin{cases} A_0 = \dfrac{1}{g_0} = h_0 \\ A_m = -\dfrac{g_m}{g_0} = -g_m h_0 \ \text{ para } \ m = 1, 2, \ldots \end{cases}$$

La secuencia $A_0, A_1, \ldots$ que representa a F, se denomina $AR(\infty)$ o **secuencia Auto-Regresiva**, que puede utilizarse para calcular la salida Y_k como función de los valores previos y la entrada X_k. Esta representación es válida para filtros con inversa.

Si F^{-1} es FIR, entonces existe p tal que $A_m = 0$ para $m \geq p$. El filtro F se denomina entonces **Auto-Regresivo de orden p**, que denotaremos por $AR(p)$.

2.3.5. Proceso de media móvil

Los procesos de media móvil o **Moving Average** (MA) se representan por [42]

$$y(k) = C_0 v(k) + C_1 v(k-1) + \cdots + C_q v(k-q)$$

La suma de $v(k), v(k-1), \ldots v(k-q)$ describe a un proceso acumulativo.

2.3.6. Transformada *Z*

El manejo de filtros permite usar la función de transferencia, como respuesta al impulso h, en serie de potencias [3]

$$H(z) = h_0 z + h_1 z^{-1} + h_1 z^{-2} + \cdots \quad z \in \text{ROC}$$

La función de transferencia del filtro

$$F = \frac{Q_0 + Q_1 B + \cdots + Q_q B^q}{P_0 + P_1 B + \cdots + P_p B^p}$$

con $P_0 \neq 0$ es

$$H(z) = \frac{Q_0 + Q_1 z^{-1} + \cdots + Q_q z^{-q}}{P_0 + P_1 z^{-1} + \cdots + P_p z^{-p}}$$

2.3.7. Estabilidad

Un filtro F con respuesta al impulso h_n es BIBO estable ssi

$$\sum_{n=0}^{\infty} |h_n| < +\infty$$

Para una secuencia $X \in S$ siendo $||X||_\infty = \text{máx}_{t=1,\ldots,l(X)} |X_t|$. Si F es estable y $Y = F \cdot X$, entonces

$$||Y||_\infty \leqslant M||X||_\infty$$

con $M = \sum_{n=0}^{\infty} |h_n|$. De esta manera, la estabilidad BIBO [sección 2.2.2] indica que si la entrada del filtro esta acotada, también lo está la salida del filtro. En cambio, si la entrada del filtro crece, la salida también lo hace indefinidamente y el filtro no es estable. Los filtros inestables podrían diverger numéricamente.

2.4. Modelos ARMA/ARIMA como filtros de tiempo discreto

2.4.1. ARMA

Los modelos *ARMA(p,q)* son aquellos que combinan una parte Auto Regresiva y una parte de Media Móvil, y tienen la siguiente estructura [3]

$$X_k + A_1 X_{k-1} + \cdots + A_p X_{k-p} = \epsilon_k + C_1 \epsilon_{k-1} + \cdots + C_q \epsilon_{k-q} \tag{2.6}$$

con ϵ_k independientes e idénticamente distribuidos N_{0,σ^2}, y $X_{-p+1} = \ldots = X_0 = 0$

Los parámetros del proceso $A_1, \ldots, A_p$ representan los coeficientes del proceso auto-regresivo, *AR(p)*, en tanto que $C_1, \ldots, C_q$ representan los coeficientes del proceso de media móvil, *MA(q)*, y σ^2 representa la varianza del ruido.

La ecuación 2.6 puede expresarse también como un filtro utilizando operadores backshift B, al expresar el proceso ARMA, como

$$X = \mu + F\epsilon \tag{2.7}$$

$$F = \frac{1 + C_1 B + \cdots + C_q B^q}{1 + A_1 B + \cdots + A_p B^p} \tag{2.8}$$

El filtro 2.8 debe ser invertible y estable. La condición de estabilidad significa que los ceros de $1 + A_1 z^{-1} + \cdots + A_p z^{-p}$ y $1 + C_1 z^{-1} + \cdots + C_q z^{-q}$ están dentro del círculo unitario [8].

2.4.2. ARIMA

Los modelos *ARIMA(p,d,q)* corresponden al proceso de diferenciar d veces el modelo *ARMA(p,q)* original. La operación de diferenciar los modelos ARMA corresponde a diferenciar el filtro en *lag-k*. Para $d \geq 1$, el proceso ARIMA no es estacionario.

La representación de ARIMA es

$$(1 - B)^d (1 + A_1 B + \cdots + A_p B^p) \cdot Y = (1 + C_1 B + \cdots + C_q B^q) \cdot \epsilon \tag{2.9}$$

2.5. Estimación de parámetros y órdenes de los modelos ARMA/ARIMA

Existen varias metodologías para determinar el orden de los modelos ARMA. Entre ellas, las que se utilizan en este trabajo son dos:

- Iterar los modelos con tests estadísticos de validación de los errores de los coeficientes.
- Utilizar criterios de complejidad para descripción de las secuencias ARMA.

Los criterios de información indican la cantidad de parámetros que ajustan los modelos. El ajuste se realiza para estimar y minimizar σ_k^2, donde

$$\widehat{\sigma}_k^2 = \frac{1}{n} \sum_{k=1}^{n} (x_k - \overline{x})^2$$

es el estimador que minimiza la suma cuadrática de errores residuales, al utilizarse modelos de regresión lineal.

2.5.1. Akaike Information Criterion (AIC) y Bayesian Information Criterion (BIC)

El Criterio de Información Akaike (AIC) corresponde a un modelo parámetrico de la forma

$$\mathrm{AIC} = -2l(\hat{\theta}) + 2k \tag{2.10}$$

donde k es la dimensión del parámetro θ y $l(\hat{\theta})$ es la estimación de máxima verosimilitud de θ.

AIC estima la cantidad de bits del código óptimo necesario para describir la secuencia X_k al estimarse la distribución de X_k de las observaciones Y_k. AIC mide la eficiencia del modelo para describir las observaciones. El modelo a seleccionar, usando como criterio el valor de AIC será aquel que tenga el *menor* valor de AIC.

Para problemas de regresión lineal, AIC puede expresarse como

$$\mathrm{AIC} = \log \widehat{\sigma}_k^2 + \frac{n + 2k}{n}$$

donde n es la cantidad de observaciones de las secuencias [34].

Se ha mostrado que normalmente AIC tiende a sobre-estimar el orden k del modelo. Por ello, una métrica alternativa es el Criterio de Información Bayesiano (BIC).

En el caso de los modelos de regresión lineal, BIC se define como

$$\mathrm{BIC} = -2l(\hat{\theta}) + k \ln n \tag{2.11}$$

donde n es el número de observaciones. Otra forma alternativa de BIC es

$$\mathrm{BIC} = p \cdot \ln n + 2n \cdot \ln \widehat{\sigma} + C'$$

$C' = n(1 + \ln(2\pi)) + \ln n$ y p es el número de grados de libertad para los parámetros del modelo de regresión lineal.

Determinación del mejor grado para los modelos

Para determinar el grado de los modelos ARMA o ARIMA, se usan ambos criterios BIC y AIC. Así, para un modelo *ARMA(p,q)*, con diferencias de X_k, se tienen $p + q$ coeficientes, además de μ y σ^2. En el caso de AIC

$$\mathrm{AIC} = N \cdot \ln \hat{\sigma}^2 + 2 \cdot (p + q + 2)$$

AIC considera los grados de libertad del sistema, no cuántas veces las observaciones se diferencian [9].

2.6. Modelos ARMAX y GARCH

Los modelos ARMA para proyecciones usados en los ámbitos de control de sistemas y económicos corresponden a ARMAX y GARCH, respectivamente.

Modelos GARCH

Los modelos GARCH son clasificados como modelos de volatividad condicional y están basados en la optimización exponencial de pesos de valores previos, para predicción posterior de la volatividad. Los parámetros del modelos GARCH habitualmente son estimados usando técnicas de máxima verosimilitud [36].

El modelo de retornos ARCH(1) correponde a [34, cap.5]

$$\begin{aligned} y_k &= \sigma_k \epsilon_k \\ \sigma_k^2 &= \alpha_0 + \alpha_1 \cdot y_{k-1}^2 \end{aligned}$$

con $\epsilon_k \sim N_{(0,1)}$. Al igual que con los modelos ARMA, se imponen restricciones al modelo, en este caso α_1 no negativo, o σ_k^2 no negativo. En el modelo de retornos ARCH la varianza condicional depende del retorno previo, esto es

$$y_k \mid y_{k-1} \sim N(0, \alpha_0 + \alpha_1 \cdot y_{k-1}^2)$$

El modelo ARCH(1) se extiende a ARCH(m), con $y_k = \sigma_k \epsilon_k$, cuando

$$\sigma_k^2 = \alpha_0 + \alpha_1 \cdot y_{k-1}^2 + \cdots + \alpha_m \cdot y_{k-m}^2$$

La función de máxima verosimilitud en este caso es

$$L(\alpha \mid y_1, \ldots, y_m) = \prod_{k=m+1}^{n} f_\alpha(y_k \mid y_{k-1}, \ldots, y_{k-m})$$

donde $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_m)$ y la densidad condicional guassiana $f_\alpha(\cdot \mid \cdot)$, para $k > m$, se tiene

$$y_k \mid y_{k-1}, \ldots, y_{k-m} \sim N(0, \alpha_0 + \alpha_1 \cdot y_{k-1}^2 + \cdots + \alpha_m \cdot y_{k-m}^2)$$

La generalización de ARCH, modelos llamados GARCH, tiene la siguiente forma para GARCH(1,1)

$$\begin{aligned} y_k &= \sigma_k \cdot \epsilon_k \\ \sigma_k^2 &= \alpha_0 + \alpha_1 \cdot y_{k-1}^2 + \beta_1 \cdot \sigma_{k-1}^2 \end{aligned}$$

Con la restricción $\alpha_1 + \beta_1 < 1$, y con un modelo ARMA(1,1) no gaussiano, el modelo para un proceso cuadrático es

$$y_k^2 = \alpha_0 + (\alpha_1 + \beta_1) \cdot y_{k-1}^2 + v_k - \beta_1 \cdot v_{k-1}$$

donde v_k se define por $v_k = \sigma_k^2 \cdot (\epsilon_k^2 - 1)$.

Modelos ARMAX

En identificación de sistemas es usual controlar el sistema con una entrada exógena u_k

$$y_k + c_1 \cdot y_{k-1} + \ldots + c_n \cdot y_{k-n} = d_0 \cdot v_k + b_1 \cdot u_{k-1} + \ldots + b_n \cdot u_{k-n}$$

El vector u_k se relaciona con las observaciones mediante

$$x_k = \Gamma \cdot u_k + \sum_{j=1}^{p} a_j \cdot x_{k-j} + \sum_{i=1}^{q} c_i \cdot w_{k-i} + w_k$$

donde Γ es una matriz de parámetros $p \times r$, con r la longitud del vector de control y w_k ruido blanco de media nula. Los variables a y c corresponden a los parámetros de los modelos AR y MA, respectivamente.

2.7. Modelos de espacio de estado

En sistemas reales es usual utilizar el modelo de observación [9]

$$\text{Observación} = \text{Señal} + \text{Ruido}$$

y para datos

$$\text{Datos} = \text{Ajuste} + \text{Residuos}$$

El objetivo es partir de los datos observados, usar modelos y métodos de modelos de espacio de estado, para la estimación de las variables de los modelos y proyección de sus valores, como por ejemplo, para la identificación de sistemas lineales.

Las variables de estado del sistema concurren en el vector de estado del sistema. La observación de una serie de tiempo univariada, para cada observación, en un tiempo k, se denota por y_k. El vector de estado en un tiempo k es x_k. De esta manera, la observación asociada al modelo de espacio de estado es

$$y_k = C_k \cdot x_k + n_k \tag{2.12}$$

donde n_k es el error de observación en el instante k.

El vector de estado x_k principal componente del modelo, normalmente no puede ser observado directamente, y tampoco se posee una estimación a priori del mismo. Por ello, se pueden pre-suponer valores iniciales para x_k, en base a estadísticas de las series de valores disponibles. Los métodos y modelos de identificación de sistemas, permiten en forma sistemática, estimar un valor inicial de x, es decir x_0, para que después el sistema actualice los siguientes valores.

El proceso continúa con la alternativa de actualización recursiva de una estimación de x_k (con x_0 como valor inicial), con la ecuación de actualización del sistema dinámico, de la forma

$$x_{k+1} = A_k \cdot x_k + \nu_k \tag{2.13}$$

con A_k, matriz de cambio conocida, y ν_k, vector de desviaciones.

Las dos ecuaciones (2.12) (2.13) previas forman el modelo de espacio de estado, la primera como ecuación de observación o medida y la última como ecuación de transición, o estado de sistema.

Los errores en las ecuaciones de observación y transición generalmente se consideran independientes, para todo k, de x_k y su secuencia de valores. La secuencia x_{k+1} se considera una cadena de Markov de orden 1, es decir, que su valor depende sólo de x_k, y no de valores previos.

La aplicación de los modelos de espacio de estado a problemas de sistemas dinámicos es común, dado que asumen conocidos a priori, las ecuaciones de movimiento, como varios otros problemas de ingeniería. Se describirá entonces como usar modelos de series de tiempo en la forma de modelos de espacio de estado, y aplicación entonces de filtros, como filtros de Kalman, descritos en sección [2.8]. Cabe mencionar que el uso de los modelos de series de tiempo, en conjunto con modelos dinámicos, deriva en diferentes propuestas de modelos, generalmente de menos expansión.

Ejemplos de modelos de espacio de estado en series de tiempos [9]:

- Modelos lineales dinámicos.
- Modelos estructurales.

Para la construcción de los modelos de series de tiempos, normalmente no se asumen estructuras o propiedades de la serie, a priori, como sucede generalmente con los sistemas dinámicos. Por ello, los requerimientos de identificación de variancias y covariancias. Para inicializar el sistema, puede realizarse estimación e identificación del mismo, con información y datos previos.

2.8. Filtro de Kalman de tiempo discreto

El filtro de Kalman de tiempo discreto puede ser derivado directamente desde la representación de modelos de espacio de estado [42]

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + V\nu_k \\ y_k &= Cx_k + Du_k + n_k \end{aligned}$$

donde ν_k y n_k son procesos de ruido blanco normales independientes de media cero y varianzas

$$\mathbb{E}\{\upsilon_k \cdot \upsilon_k^T\} = M$$

$$\mathbb{E}\{n_k \cdot n_k^T\} = N$$

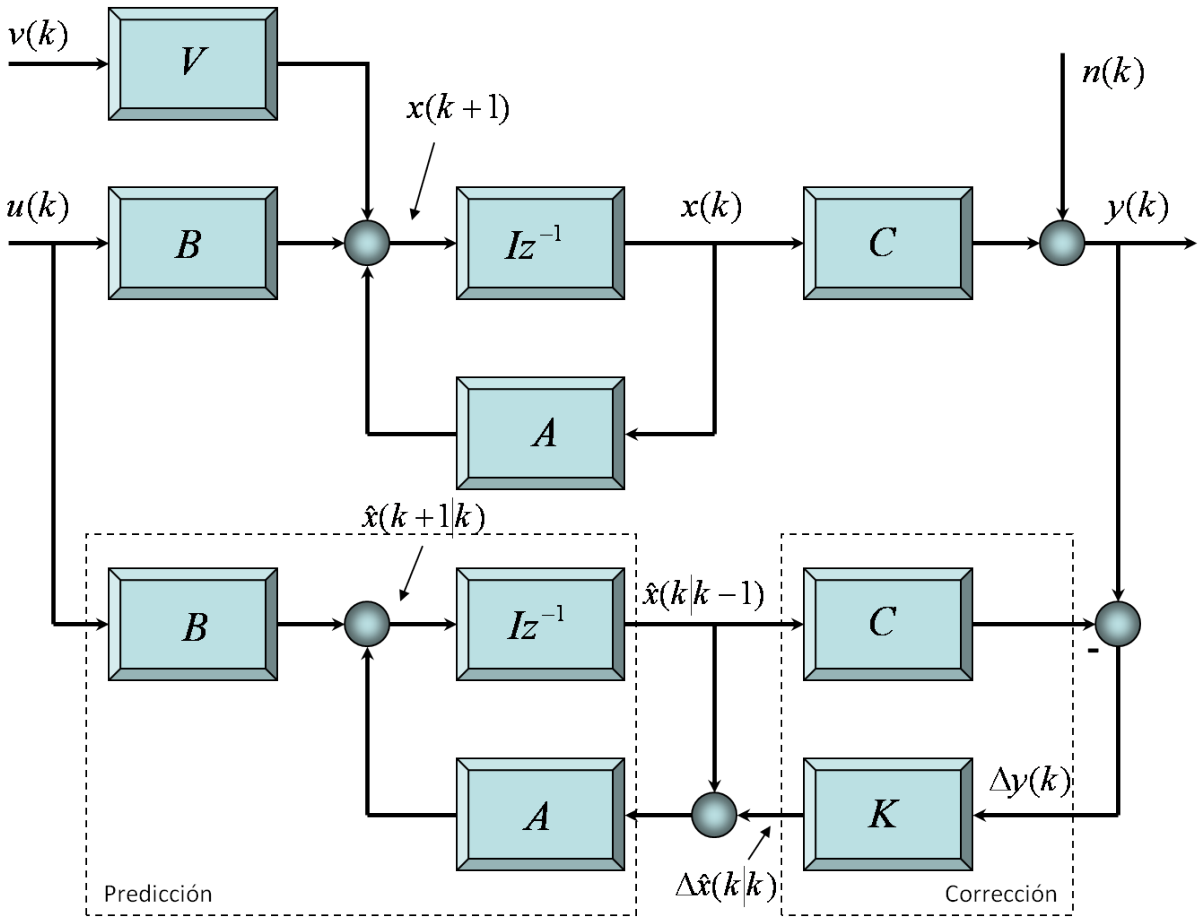


Figura 2.3.: Diagrama de bloques del filtro de Kalman de tiempo discreto

Se busca determinar un filtro lineal óptimo para minimizar el error cuadrático medio de los errores de predicción. El error se expresa por un vector norma 2

$$\begin{aligned} V &= \mathbb{E}\{||\widehat{x}_{k+1} - x_{k+1}||_2^2\} \\ &= \mathbb{E}\{(\widehat{x}_{k+1} - x_{k+1})^T(\widehat{x}_{k+1} - x_{k+1})\} \end{aligned}$$

La función de costo, al minimizarse, permite construir el filtro de Kalman. El entorno descrito corresponde a un filtro para predicción/corrección, es decir, con proyecciones de un paso que pueden ajustarse (corregirse) en base a las salidas y_{k+1}.

x_k corresponde al estado en k, $\widehat{x}_{k+1|k}$ a la predicción de los estados hasta k, y $\widehat{x}_{k+1|k+1}$ a la predicción de los estados hasta $k+1$. La notación $k+1|k$ denota que el estado $k+1$ se determina en base a las observaciones en k.

La matriz de covarianza P_k se escribe como

$$P_k = \mathbb{E}\{(\widehat{x}_k - x_k)(\widehat{x}_k - x_k)^T\}$$

La predicción de un paso, se deriva por

$$x_{k+1} = Ax_k + Bu_k + Vv_k$$

donde υ_k es indeterminado, con media cero. La actualización del estado es

$$\widehat{x}_{k+1|k} = A\widehat{x}_k + Bu_k \tag{2.14}$$

basado en el paso k.

La matriz de covarianza P_{k+1}^- se determina por

$$\begin{aligned} P_{k+1}^- &= \mathbb{E}\{(\widehat{x}_{k+1|k} - x_{k+1})(\widehat{x}_{k+1|k} - x_{k+1})^T\} \\ &= \mathbb{E}\{(A\widehat{x}_k - Ax_k + V\upsilon_k)(A\widehat{x}_k - Ax_k + V\upsilon_k)\} \\ &= A \cdot \mathbb{E}\{(\widehat{x}_k - x_k)(\widehat{x}_k - x_k)^T\}A^T \\ &\quad + A \cdot \mathbb{E}\{(\widehat{x}_k - x_k)\upsilon^T\}V^T \\ &\quad + V \cdot \mathbb{E}\{\upsilon_k(\widehat{x}_k - x_k)^T\}A^T \\ &\quad + V \cdot \mathbb{E}\{\upsilon_k \cdot \upsilon_k^T\}V^T \end{aligned}$$

Finalmente

$$P_{k+1}^- = AP_kA^T + VMV^T$$

$^-$ representa la matriz de predicción de covarianza, del paso previo, a la corrección con la nueva observación. Se utiliza que $\widehat{x}_k$ y que x_k no están correlacionadas con υ_k, para obtener la matriz P_{k+1}^-, y que además υ_k es de media cero, por lo que

$$\begin{aligned} \mathbb{E}\{x_k \cdot \upsilon_k^T\} &= 0 \\ \mathbb{E}\{\hat{x}_k \cdot \upsilon_k^T\} &= 0 \end{aligned}$$

De esta forma, para predicción, la forma del filtro de Kalman es:

Paso 1: Predicción

$$\begin{aligned} \widehat{x}_{k+1|k} &= A\widehat{x}_k + Bu_k \\ P_{k+1}^- &= AP_kA^T + VMV^T \end{aligned}$$

En forma análoga [42], se obtienen las expresiones para corrección:

Paso 2: Corrección

$$\begin{aligned} K_{k+1} &= P_{k+1}^-C^T(CP_{k+1}^-C^T + N)^{-1} \\ \widehat{x}_{k+1|k+1} &= \widehat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\widehat{x}_{k+1|k}) \\ P_{k+1} &= (I - K_{k+1}C)P_{k+1}^- \end{aligned}$$

donde K_k se denomina ganancia del filtro de Kalman. Para las condiciones iniciales del filtro, puede utilizarse

$\widehat{x}_0 = 0$

P_0 : covarianza de la serie de observaciones, hasta x_0.

m pasos de predicción

El filtro para predicción descrito antes es de un paso, para m pasos [36], se tiene

$$\widehat{x}_{k+m-1|k-1} = CA^{m-1}\widehat{x}_k + Du_{k+m-1} + \sum_{j=1}^{m-1} CA^{j-1}Bu_{k+m-1-j}$$

y covarianza del error de predicción

$$CA^{m-1}P_k(A^T)^{m-1}C^T + EE^T + \sum_{j=1}^{m-1} CA^{j-1}EE^T(A^T)^{j-1}C^T$$

con E matriz de covarianza del ruido blanco, de media cero.

3. Desarrollo

En este capítulo se presenta el trabajo realizado para la generación de los modelos de tráfico de los nodos de banda ancha. Los métodos de minimización de errores y de optimización se aplican en la construcción de los modelos de las series de tiempo, a partir de las series de observaciones.

La selección de los polinomios de los modelos se basa en criterios de información que minimizan las densidades de probabilidad de los parámetros que representan a las series de tiempo, coeficientes y orden de los polinomios, con aproximación a las distribuciones empíricas que las describen. Los filtros de Kalman se inicializan con los valores de correlación de las series de observaciones.

En la sección final del capítulo se presentan los modelos ARMA para los 4 nodos de acceso de banda ancha, además de las proyecciones utilizando los modelos ARMAX, GARCH y filtros de Kalman.

3.1. Metodología

Para la construcción de los modelos ARMA además de filtros de Kalman se utiliza la siguiente metodología:

1. Se registran las observaciones de tráfico por nodo de acceso de banda ancha.
2. Las observaciones son procesadas para agrupar las mediciones por día y posteriormente se normalizan con los totales de tráfico agregado por día.
3. Los modelos ARMA se construyen con los criterios de información AIC y BIC, mediante un proceso iterativo en que se seleccionan los grados de los modelos. Los métodos de minimización permiten obtener los coeficientes de los modelos, a la vez que se seleccionan los grados de los modelos.
4. Posteriormente los coeficientes y grados de los modelos ARMA, en formato ARMAX y GARCH, se mapean a filtros de tiempo discreto.
5. Utilizando la representación con filtros de los modelos ARMA se procede a la predicción de tráficos para uno y 4 pasos.
6. Los filtros de Kalman se emplean para proyecciones de uno y 4 pasos.
7. Finalmente se realiza la comparación de los distintos modelos ARMAX, GARCH y filtro de Kalman utilizando distintas métricas de desempeño, en la sección de resultados.

A continuación se desarrollan los métodos para construcción de los modelos, del paso 3 de la metodología.

3.2. Mínimos cuadrados

Entre varias alternativas de resolución del problema de optimización de minimización del error, en la construcción de los modelos de las series de tiempo, es usual el utilizar la minimización en norma 2, es decir, mínimos cuadrados.

Se define el error de observación/medición por

$$e_k = y_k - y_{m_k}$$

El algoritmo de mínimos cuadrados minimiza la siguiente función de costos

$$V = e_1^2 + e_2^2 + \cdots + e_n^2 = \sum_{k=1}^{n} e_k^2$$

Entre las razones para utilizar una función de costos cuadráticas se tienen [42]

- La ventaja del uso de mínimos cuadrados es que requiere sólo conocer momentos de orden 1 y 2.
- Si el ruido se distribuye normalmente la estimación es asíntoticamente insesgada, es decir, idéntica para todos los valores de y_k.
- La minimización cuadrática comparada a otros métodos, es habitualmente de uso más extendido, por lo que se tienen varias alternativas de resolución numérica.

Sin embargo, la función cuadrática de costo, podría amplificar los efectos de observaciones fuera de rango.

Para el sistema lineal, como se muestra en la sección de filtros de Kalman [2.8]

$$\begin{aligned} y_k &= Cx_k + n_k \\ e_k &= y_k - C_m x_k \end{aligned}$$

se tiene que el error corresponde a $e_k =$ observación - predicción, además

$$V = \sum_{k=1}^{n} e_k^2 = \sum_{k=1}^{n} \left(y_k - C_m x_k\right)^2$$

con C_m para m pasos desde $k = m$.

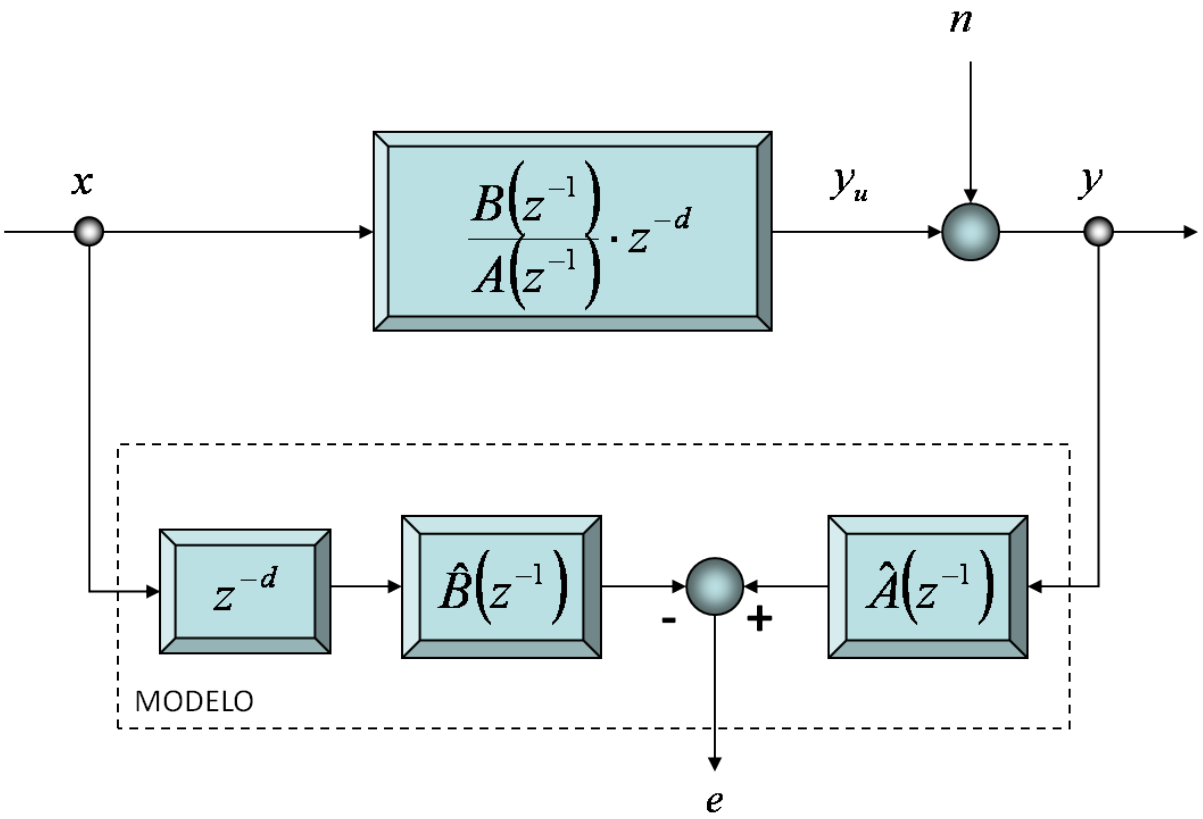


Figura 3.1.: Esquema de la estimación de mínimos cuadrados

La optimización del modelo corresponde a minimizar

$$\begin{aligned} \frac{dV}{dC_m} &= -2\sum_{k=1}^{n}(y_k - C_m x_k)\cdot x_k \\ \frac{dV}{dC_m}\Big|_{C_m=\widehat{C}} &\doteq 0 \implies -2\sum_{k=1}^{n}\left(y_k - \widehat{C}x_k\right)\cdot x_k = 0 \end{aligned}$$

Entonces, para $\widehat{C}$ se tiene

$$\widehat{C} = \frac{\sum_{k=1}^{n} y_k \cdot x_k}{\sum_{k=1}^{n} x_k^2} = \frac{\hat{R}_{xy}(0)}{\hat{R}_{xx}(0)}$$

$\widehat{C}$ se estima por la función de correlación cruzada y la función de autocorrelación, para $\tau = 0$.

3.3. Condiciones de Kuhn–Tucker para optimización

Las condiciones de Kuhn–Tucker - KT (o Karush–Kuhn–Tucker - KKT) [5], representan una generalización del método de Lagrange para condiciones con deigualdades, en el contexto de resolución de problemas de programación no lineal.

El método KKT se usa en las iteraciones, junto con las ecuaciones de Yule–Walker, para la resolución del problema de búsqueda de coeficientes y parámetros de los modelos ARMA, al utilizarse minimización cuadrática con múltiples condiciones (descripción básica en sección [3.2]).

3.3.1. Condiciones de Kuhn–Tucker

En el método de Lagrange se utiliza el método del gradiente y restricciones con condiciones de igualdad, para encontrar los puntos que cumplen las condiciones. De manera similar, se utilizan las condiciones de gradiente, ortogonalidad y restricciones con inegualdades, para encontrar los puntos estacionarios de las restricciones.

Las condiciones KT de primer orden necesarias para un problema general de optimización que se expresan como

$$\text{minimizar } z = f(x)$$

sujeto a

$$\begin{aligned} g_i(x) &\leq 0, \quad i = 1, 2, \ldots, m \\ h_i(x) &= 0, \quad i = 1, 2, \ldots, p \end{aligned}$$

Las condiciones KT representan un conjunto de ecuaciones que deben ser resueltas para todos los puntos locales que minimizan el problema. Los puntos que satisfacen sólo las condiciones necesarias pueden ser candidatos a considerarse mínimos locales y se denominan puntos KT. Luego, se comprueban las condiciones suficientes para determinar si un punto KT es o no un mínimo local.

Todas las restricciones deben ser $\leq$, una restricción de la forma

$$h(x_1, x_2, \ldots, x_n) \geq 0$$

puede re-escribirse por

$$-h(x_1, x_2, \ldots, x_n) \leq 0$$

También las restricciones de la forma

$$h(x_1, x_2, \ldots, x_n) = 0$$

pueden ser reemplazadas por

$$\begin{aligned} h(x_1, x_2, \ldots, x_n) &\leq 0 \quad \text{y} \\ -h(x_1, x_2, \ldots, x_n) &\leq 0 \end{aligned}$$

Por ejemplo, $x_1 + 3x_2 = 4$ puede reemplazarse por

$$\begin{aligned} x_1 + 3x_2 &\leq 4 \quad \text{y} \\ -x_1 - 3x_2 &\leq -4 \end{aligned}$$

3.3.2. Algoritmo de Solución de Problema de Minimización

1. Contruir el Lagrangiano del problema

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x)$$

2. Determinar todas las soluciones $(\overline{x}, \overline{\lambda})$ del siguiente sistema de ecuaciones no lineales, y restricciones de desigualdad.

	rango	condiciones
$\dfrac{\partial L}{\partial x_j} = 0$	$j = 1, 2, \ldots, n$	gradiente
$g_i(x) \leq 0$	$i = 1, 2, \ldots, m$	factibilidad
$\lambda_i \cdot g_i(x) = 0$	$i = 1, 2, \ldots, m$	ortogonalidad
$\lambda_i \geq 0$	$i = 1, 2, \ldots, m$	no negatividad

Tabla 3.1.: Condiciones y sistemas de ecuaciones KKT

3. Si todas las funciones $g_i(x)$ son convexas, el punto $\overline{x}$ es un mínimo global; en caso contrario, la revisión de cada solución $(\overline{x}, \overline{\lambda})$ indicará si $\overline{x}$ es un mínimo local.

Para cada desigualdad, se deben considerar dos alternativas: desigualdades inactivas y activas.

Restricciones de desigualdad inactivas

Cada restricción que sigue $g(\overline{x}) < 0$, no determina óptimo, por lo que no se evaluan para las condiciones de optimización.

Restricciones de desigualdad activas

Corresponden a las restricciones $g(\overline{x}) = 0$.

Condiciones necesarias y suficientes

A continuación se describen las condiciones necesarias y suficientes para que $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ sea la solución óptima del problema de programación no lineal (NLP)

$$\text{minimizar } z = f(x_1, x_2, \ldots, x_n)$$

sujeto a

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &\leq 0 \\ g_2(x_1, x_2, \ldots, x_n) &\leq 0 \\ &\vdots \\ g_m(x_1, x_2, \ldots, x_n) &\leq 0 \end{aligned}$$

Teorema para Condiciones necesarias, Problema de Minimización

Si $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ es una solución óptima, entonces $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ debe satisfacer las m condiciones previas, y existen m multiplicadores $\overline{\lambda} = (\overline{\lambda}_1, \overline{\lambda}_2, \ldots, \overline{\lambda}_m)$ que cumplen

$$\begin{aligned} \frac{\partial f(\overline{x})}{\partial x_j} + \sum_{i=1}^{m} \overline{\lambda}_i \frac{\partial g_i(\overline{x})}{\partial x_j} &= 0, \quad j = 1, 2, \ldots, n \\ \overline{\lambda}_i [g_i(\overline{x})] &= 0, \quad i = 1, 2, \ldots, m \\ \overline{\lambda}_i &\geq 0, \quad i = 1, 2, \ldots, m \end{aligned}$$

Las condiciones KT son aplicadas a problemas NLP, en que las variables deben ser no negativas. Las condiciones KT se usan para el siguiente problema NLP

$$\text{minimizar } z = f(x_1, x_2, \ldots, x_n)$$

sujeto a

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &\leq 0 \\ g_2(x_1, x_2, \ldots, x_n) &\leq 0 \\ &\vdots \\ g_m(x_1, x_2, \ldots, x_n) &\leq 0 \\ -x_1 &\leq 0 \\ -x_2 &\leq 0 \\ &\vdots \\ -x_n &\leq 0 \end{aligned}$$

En que los multiplicadores $\lambda_1, \lambda_2, \ldots, \lambda_n$ con las restricciones de no negatividad, permiten describir el teorema anterior como condiciones necesarias, con restricciones no negativas.

Teorema para Condiciones necesarias, Problema de Minimización, restricciones no negativas

Si
$$\frac{\partial f(\overline{x})}{\partial x_j} - \sum_{i=1}^{m} \overline{\lambda}_i \frac{\partial g_i(\overline{x})}{\partial x_j}$$

representa un problema de minimización, y $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ es la solución óptima del problema NLP, entonces $\overline{x}$ es la solución óptima de

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &\leq 0 \\ g_2(x_1, x_2, \ldots, x_n) &\leq 0 \\ &\vdots \\ g_m(x_1, x_2, \ldots, x_n) &\leq 0 \\ -x_1 &\leq 0 \\ -x_2 &\leq 0 \\ &\vdots \\ -x_n &\leq 0 \end{aligned}$$

y existen los multiplicadores $\overline{\lambda} = (\overline{\lambda}_1, \overline{\lambda}_2, \ldots, \overline{\lambda}_m, \overline{\mu}_1, \overline{\mu}_2, \ldots, \overline{\mu}_n)$ que satisfacen

$$\begin{aligned} \frac{\partial f(\overline{x})}{\partial x_j} - \sum_{i=1}^{m} \overline{\lambda}_i \frac{\partial g_i(\overline{x})}{\partial x_j} - \mu_j &= 0, \quad j = 1, 2, \ldots, n \\ \overline{\lambda}_i [g_i(\overline{x})] &= 0, \quad i = 1, 2, \ldots, m \\ \left[\frac{\partial f(\overline{x})}{\partial x_j} + \sum_{i=1}^{m} \overline{\lambda}_i \frac{\partial g_i(\overline{x})}{\partial x_j}(\overline{x}) \right] \overline{x} &= 0 \\ \overline{\lambda}_i &\geq 0 \quad i = 1, 2, \ldots, m \\ \overline{\mu}_j &\geq 0 \quad i = 1, 2, \ldots, m \end{aligned}$$

Con $\overline{\mu}_j \geq 0$, la primera ecuación del sistema equivale a

$$\frac{\partial f(\overline{x})}{\partial x_j} + \sum_{i=1}^{m} \overline{\lambda}_i \frac{\partial g_i(\overline{x})}{\partial x_j} = 0, \quad j = 1, 2, \ldots, n$$

Teorema para condiciones suficientes, Problema de Minimización

Si $f(x_1, \ldots, x_n)$ es una función convexa y $g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)$ son funciones convexas, entonces cualquier punto $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ que satisface las condiciones del Teorema para Condiciones necesarias - Problema de Minimización [3.3.2], es una solución óptima del sistema.

Se desarrolla un ejemplo de minimización de la función $z = (x_1 - 2)^2 + (x_2 - 3)^2 + (x_3 - 3)^2$ en el Anexo C.1.

3.4. Criterio de máxima verosimilitud

Los modelos estocásticos asignan probabilidades relativas a las observaciones del sistema. Las clases de modelos son conjuntos de densidades de probabilidades $\{p_\theta,\ \theta \in \Theta\}$, donde $\theta \in \Theta$ representan a los vectores de variables no observables.

Si las q series de tiempo son observadas en un intervalo de tiempo N, entonces p_θ es una densidad de probabilidad en $\{\mathbb{R}^q\}^N$. El método de máxima verosimilitud elige el modelo que asigna la mayor probabilidad a las observaciones de los datos.

Si se denota a las secuencias de datos por $d \in (\mathbb{R}^q)^N$, la función de máxima verosimilitud $L(\theta) := p_\theta(d)$ se maximiza sobre el conjunto Θ de parámetros.

La estimación de máxima verosimilitud ML requiere que las densidades de probabilidad sean funciones explícitas de los parámetros θ. La maximización de la función ML habitualmente es un problema de programación no lineal que involucra varias variables, por ello la necesidad de utilizar distintos métodos de optimización, como los descritos antes en sección 3.3.

Bajo condiciones generales los estimadores de máxima verosimilitud tienen propiedades asintóticas óptimas [41].

3.5. Implementación series de tiempo, modelos ARMA

3.5.1. Alternativas para análisis de series de tiempo

De la revisión de alternativas para análisis de series de tiempo, para modelos ARMA/ ARIMA, para contexto de aplicaciones en Matlab[1], se tienen varias opciones:

- Toolbox de Identificación de Sistemas (IS), con funciones ARMAX.
- Toolbox de Econometría, con funciones GARCH, Autocorr, Parcorr, entre otras.
- Toolbox de terceros [libre uso], por ejemplo: ARMASA [4], para predicciones.
- Códigos con funciones de IS, Economía, Señales o de Estadísticas [stats].
- Toolbox de Análisis de Señales, con función *filter* y asociadas.
- Opcional: toolbox de Estadísticas, con objeto Time Series, TS.

Finalmente, además de formato de TS en modelos de espacio de estado, para uso de filtros de Kalman, de uno o m pasos de proyección.

[1] Matlab© de Mathworks, http://www.mathworks.com/. Versión de trabajo: R2010a

3.5.2. Descripción de las observaciones

Descripción de setup de las observaciones

Los nodos de acceso de banda ancha, en observación, corresponden a 4 nodos DSLAM de Santiago, Chile.

Los nodos corresponden a los siguientes emplazamientos de equipos técnicos de la red de Movistar Chile: Apoquindo, La Dehesa, Santa Lucía y La Pincoya.

En cada sitio, se encuentran más nodos de acceso de banda ancha y otros tipos de nodos de red.

Nodo	ID	# accesos
Apoquindo	apoq	768
La Dehesa	dehesa	1.280
Santa Lucía	slucia	624
La Pincoya	pcoya	768

Tabla 3.2.: Cantidad de líneas de acceso, de servicio de datos, por nodo

Fechas y cantidad de observaciones

- Las observaciones se realizaron entre el 2 Enero 2011 y el 30 de Abril 2011.
- Diariamente se obtuvieron 96 muestras, una cada quince minutos.
- Se elimina el día 13 Marzo de 2011, por no tenerse registro de los datos de tráfico para los nodos.
- En total, las series de observaciones corresponden a
  - 11.232 puntos, 117 días
- Para la construcción de los modelos y validación, se utilizan
  - Construcción: 8.352 puntos, 87 días
  - Validación: 2.880 puntos, 30 días, a continuación de los días de construcción de los modelos

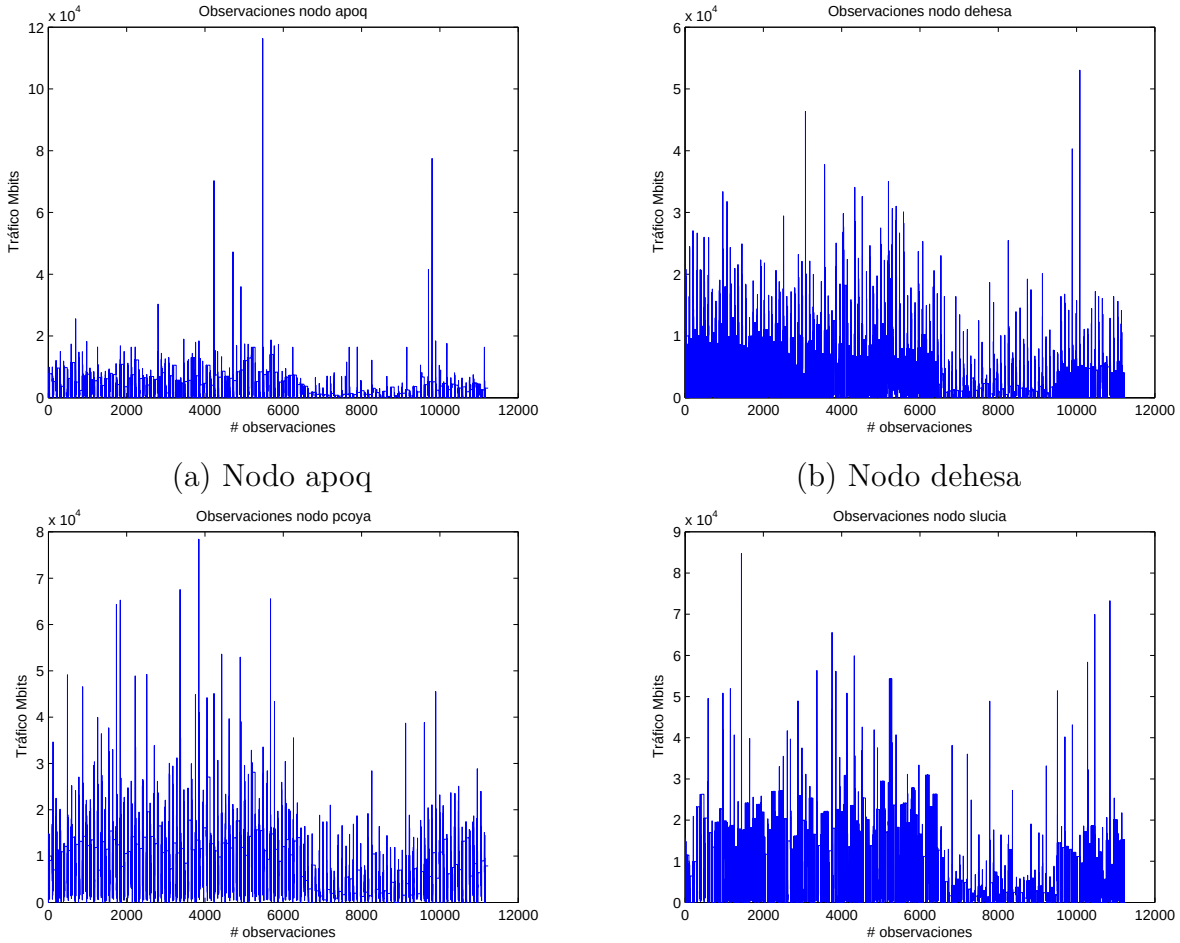
Secuencia de observaciones de tráfico

Los gráficos de las secuencias de observaciones se muestran en la figura [3.2].

3.5.3. Función armax

La función *armax* del toolbox de Identificación de Sistemas (IS) permite describir modelos ARMAX para distintos tipos de sistemas lineales, dinámicos, etc. La modelación ARMAX se utiliza para identificación de los coeficientes de los modelos polinomiales mixtos (numeradores-denominadores) de los procesos AR y MA .

El uso de la transformación de dominios entre frecuencia y tiempo discreto, Fourier y Laplace, sirve para la identificación de polos y ceros, además de análisis de estabilidad de los modelos ARMA.



(a) Nodo apoq

(b) Nodo dehesa

(c) Nodo pcoya

(d) Nodo slucia

Figura 3.2.: Observaciones de tráfico de los nodos (a) apoq, (b) dehesa, (c) pcoya y (d) slucia

Implementación

El modelo ARMA corresponde a $ARMAX(p, q, 0) = ARMA(p, q)$. Los códigos en Matlab con las funciones armax para la estimación de los modelos ARMA se encuentran en la sección de Anexos A.1.

La función armax, entrega como resultado un objeto idpoly, con los coeficientes y grados (p,q), de AR y MA, respectivamente. En formato alterno, en el dominio Z, los valores de coeficientes y grados se utilizan en la contrucción de un filtro de tiempo discreto que representa al proceso ARMA.

3.5.4. Función garchfit

El modelo ARMA, al utilizar la función *garchfit*, corresponde a $ARMAX(p, q, 0)+GARCH(0, 0) = ARMA(p, q)$. Código en Anexos A.2.

La función *garchfit* entrega como resultado un objeto struct, con los coeficientes y grados (p, q), de AR y MA, respectivamente. La función *garchdisp* utiliza un objeto struct con los valores del modelo $ARMA(p, q)$ para mostrar los coeficientes y grados del modelo ARMA, como en el ejemplo de la tabla [3.3] para nodo apoq.

Mean: ARMAX(2,12,0); Variance: GARCH(0,0).
Conditional probability distribution: Gaussian.
Number of model parameters estimated: 16

Parameter	Value	Standard error	T statistic
C	-0,0130	0,1032	-0,1262
AR(1)	-0,0224	0,0340	-0,6599
AR(2)	0,9168	0,0313	29,2074
MA(1)	-0,7625	0,0344	-22,1669
MA(2)	-0,9137	0,0565	-16,1672
MA(3)	0,6867	0,0255	26,8884
MA(4)	-0,0293	0,0059	-4,9562
MA(5)	0,0459	0,0082	5,5554
MA(6)	-0,0335	0,0091	-3,6580
MA(7)	0,0201	0,0105	1,9212
MA(8)	0,0343	0,0101	3,4141
MA(9)	-0,0252	0,0128	-1,9677
MA(10)	0,0004	0,0130	0,0319
MA(11)	-0,0069	0,0111	-0,6255
MA(12)	-0,0131	0,0103	-1,2647

RSquared = 0,2150

Tabla 3.3.: Ejemplo de salida de función *garchdisp* para nodo apoq

En formato alterno, en dominio Z, los valores de coeficientes y grados de los modelos ARMA se utilizan en la generación de filtros de tiempo discreto. Con la función *filter*, del toolbox de análisis de señales, se construyen los filtros de tiempo discreto.

3.5.5. Modelos ARMA desde funciones armax y garch

Los grados de los modelos, para los nodos, son

Grados de los modelos ARMA

	función armax		función garch	
nodo	p	q	p	q
apoq	16	19	2	12
dehesa	10	18	6	9
slucia	18	13	11	2
pcoya	20	19	13	1

Tabla 3.4.: Resumen grados de los modelos ARMA, desde funciones armax y garch

Los coeficientes de los modelos, que se obtienen de las funciones armax y garch, se encuentran en Anexos A.3.1.

3.5.6. Modelos ARMA utilizando filtros

Los filtros de tiempo discreto [8][3] se utilizan como representación común de los modelos ARMA y GARCH para las proyecciones de tráfico de los nodos de acceso de banda ancha. La representación de los modelos ARMA como filtros (ecuación [2.8]) permite analizar la estabilidad de los modelos, además de uso de framework común para predicciones.

Los gráficos representan las salidas de los filtros para nodo apoq de modelo ARMA desde función armax y de filtro de Kalman, de proyección de un paso, para período de una semana y período de un mes de la secuencia de verificación de los modelos.

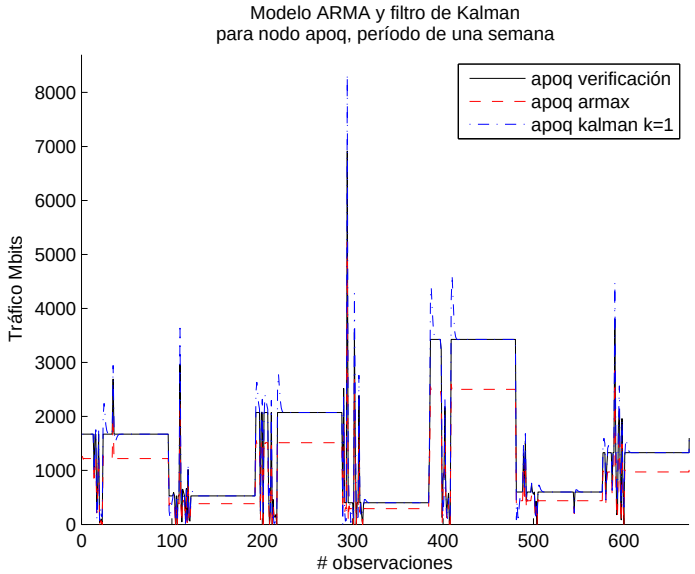


Figura 3.3.: Modelo ARMA y filtro de Kalman para nodo apoq, período de una semana

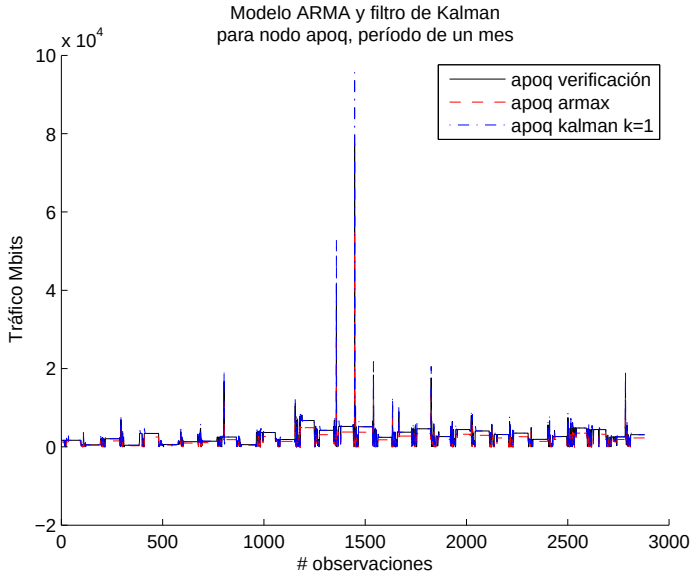


Figura 3.4.: Modelo ARMA y filtro de Kalman para nodo apoq, período de un mes

3.6. Análisis de estabilidad de los filtros

Para la revisión de estabilidad de los filtros se analiza la ubicación de los polos y zeros, además de respuesta al impulso.

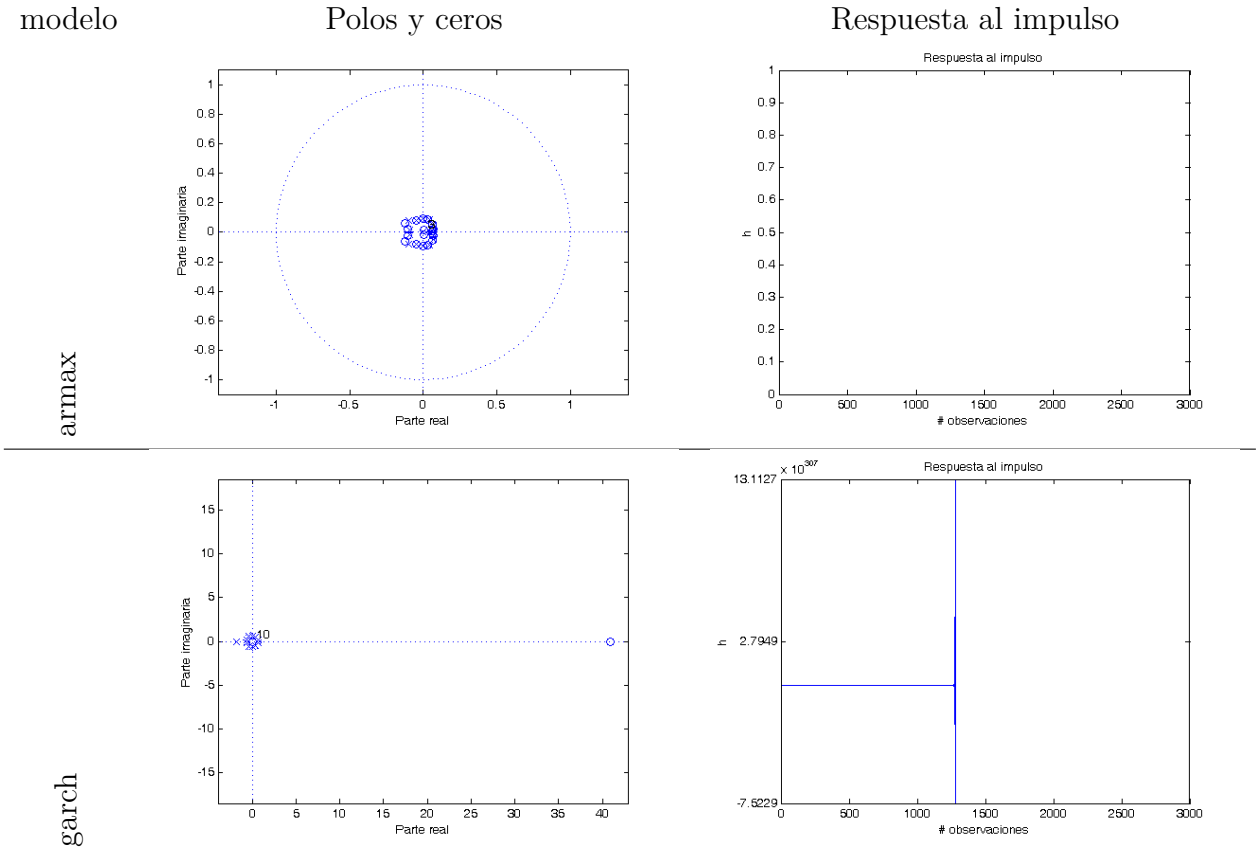


Figura 3.5.: Polos y zeros más respuesta al impulso de nodo apoq, distintos modelos ARMA De la revisión de estabilidad, se tiene que el modelo garch es inestable.

En la siguiente sección se presentan las predicciones con los modelos ARMAX y GARCH que permiten comprobar la ineficiencia de modelo GARCH, para predicción.

3.7. Predicción de *m* pasos con modelos ARMA

Los modelos ARMAX y GARCH se utilizan para las proyecciones de tráfico, de uno y 4 pasos. El período de duración de la secuencia de validación es de un mes (2.880 observaciones). En los siguientes gráficos, usando el nodo apoq como ejemplo, se observa que los modelos GARCH no son adecuados para realizar proyecciones de tráfico, como se verifica posteriormente con distintos parámetros de desempeño en la sección [4.1.1].

3.7.1. Predicción de un paso

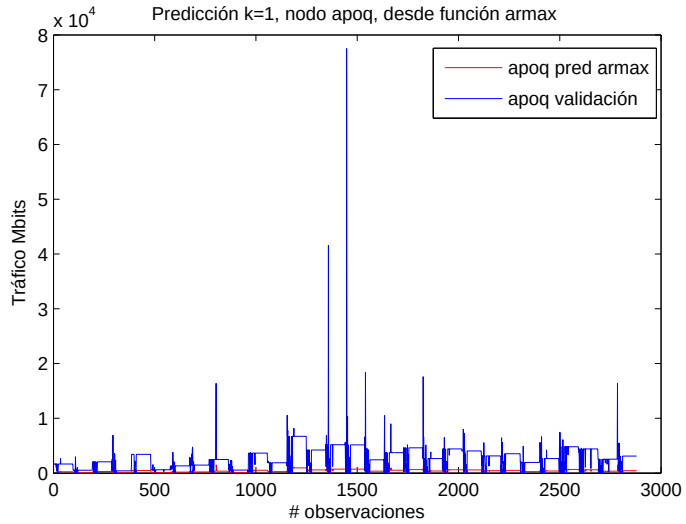


Figura 3.6.: Predicción nodo apoq, un paso, desde función armax

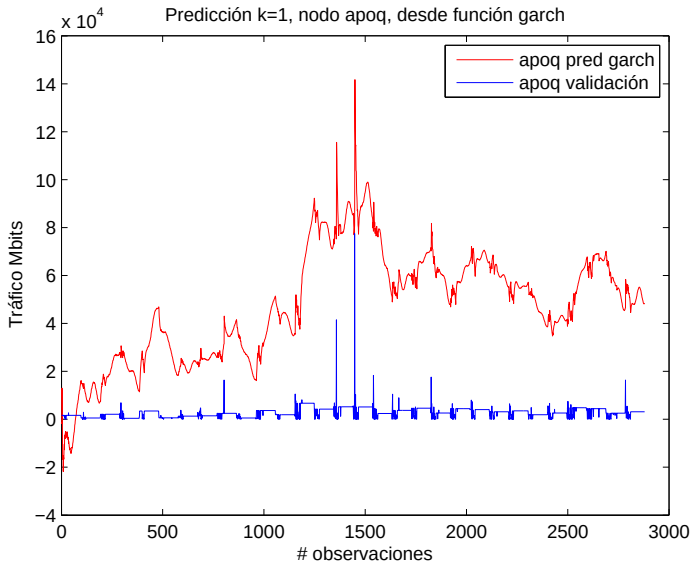


Figura 3.7.: Predicción nodo apoq, un paso, desde función garch

Los gráficos para los nodos dehesa, slucia y pcoya, se encuentran en Anexos A.4.

3.7.2. Predicción de 4 pasos

Resultados de los modelos ARMAX y GARCH, de nodo apoq, para proyecciones de 4 pasos

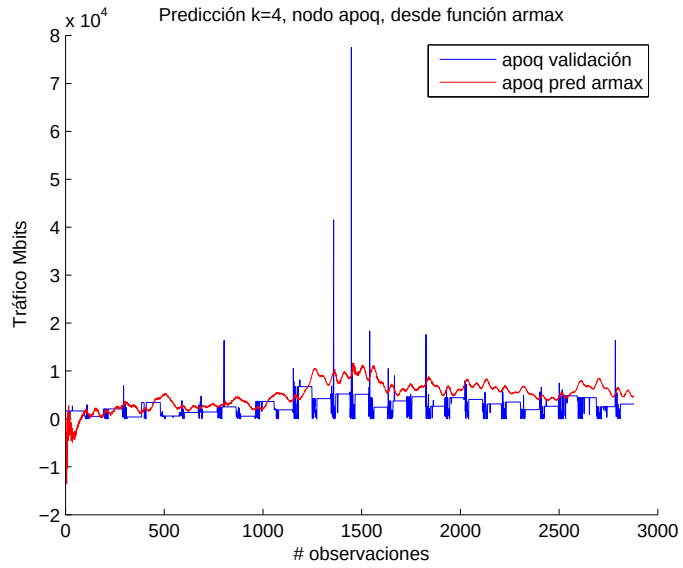


Figura 3.8.: Predicción nodo apoq, de 4 pasos, desde función armax

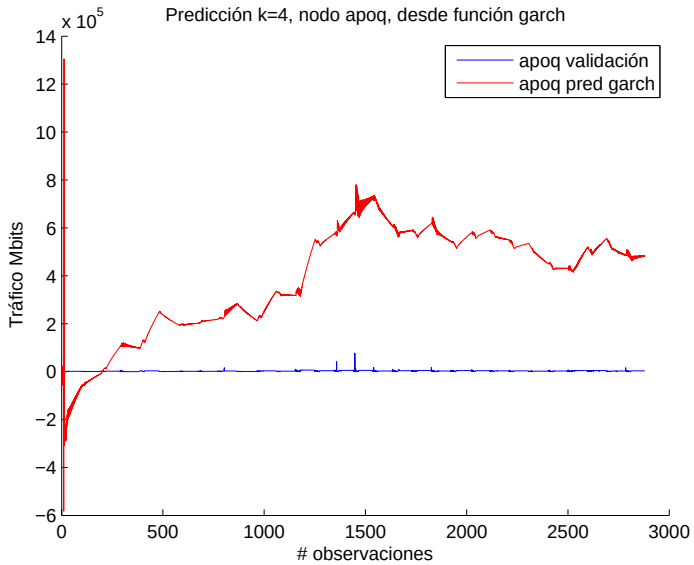


Figura 3.9.: Predicción nodo apoq, de 4 pasos, desde función garch

Los gráficos para los nodos dehesa, slucia y pcoya, se encuentran en Anexos A.4.2.

3.8. Filtros de Kalman para predicción

La implementación de los filtros de Kalman y su aplicación por nodo, se encuentra en Anexos A.5.1. Los gráficos de las secuencias para los períodos de verificación de un mes de los nodos apoq, dehesa, slucia y pcoya se encuentran en Anexos, en las secciones A.5.2 y A.5.3 para proyecciones de uno y 4 pasos, respectivamente.

3.8.1. Predicción de un paso

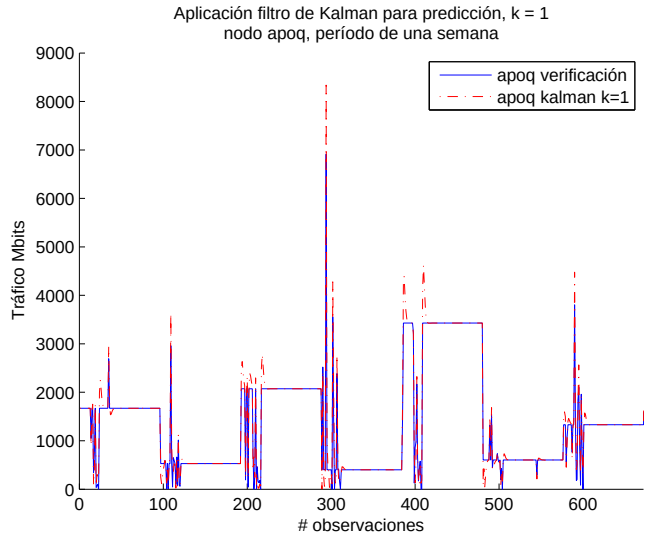


Figura 3.10.: Filtro Kalman para predicción $k = 1$, nodo apoq, período de una semana

3.8.2. Predicción de 4 pasos

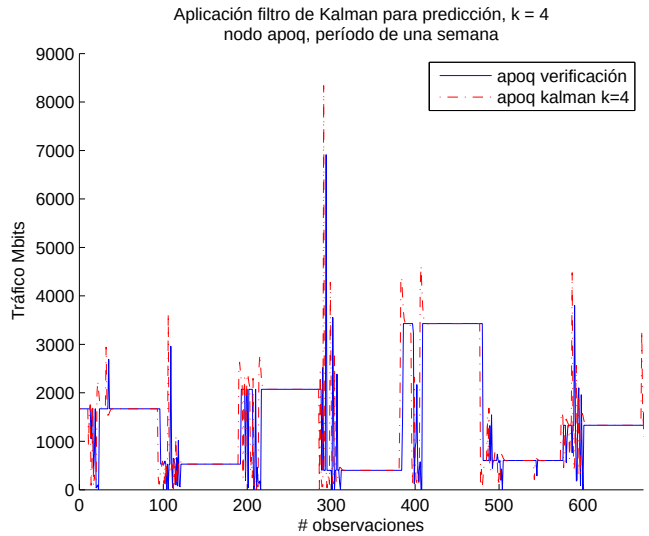


Figura 3.11.: Filtro de Kalman para predicción $k = 4$, nodo apoq, período de una semana

4. Resultados y discusión

Los distintos modelos de las series de tiempo, para las observaciones de tráfico de los nodos de banda ancha, se comparan utilizando métricas de desempeño y métodos para evaluación de las relaciones entre las proyecciones y series de validación de los modelos.

La discusión de los resultados abarca tanto a los modelos de las series de tiempo como a las proyecciones, que resultan al aplicar filtros de tiempo discreto y filtros de Kalman. Finalmente se describen posibles extensiones de este trabajo utilizando filtros adaptivos y series espacio-temporales.

4.1. Resultados

4.1.1. Comparación de desempeño de los modelos de series de tiempo

Para la comparación de desempeño de los modelos de series de tiempo y sus proyecciones con series de verificación, se utilizan las siguientes métricas y metodologías:

- Errores de predicción.
- Correlación cruzada.
- Test de Kolmogorov-Smirnov.

4.1.2. Errores de predicción

Los errores de predicción se basan en la definición usual de error

$$e_k = y_k^{\text{verificación}} - \widehat{y}_k^{\text{estimado}},$$

donde $y_k^{\text{verificación}}$ es la secuencia de observaciones que permite la validación de los distintos modelos. Este vector de mediciones es distinto al vector de generación de los modelos ARMA [3.4]. El vector de validación es temporalmente posterior al de construcción de los modelos.

Las métricas de errores de predicción [22] [43] [32] corresponden a:

Mean Absolute Error, MAE

$$\text{MAE} = \frac{1}{n}\sum_{k}^{n} e_k$$

MAE entrega la magnitud del error medio de la serie.

Mean Absolute Percentage Error, MAPE

$$\text{MAPE} = 100\,\% \frac{1}{n} \sum_{k}^{n} \left| \frac{e_k}{y_k} \right|$$

con $y_k = y_k^{validación}$. Esta medida describe en que porcentaje medio se desvía la predicción de los valores de validación. Además permite comparar series de distintas escalas.

Average Error, MEAN

$$\text{MEAN} = \frac{1}{n} \sum_{k}^{n} e_k$$

Similar a MAE, con mantención de signo de e_k. MEAN es una indicación, en promedio, si las predicciones están sub o sobre-estimadas.

Root Mean Squared Error, RMSE

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{k}^{n} e_k^2}$$

4.1.3. Resultados de errores de predicción por modelo

Las siguientes tablas resumen los valores de las métricas para errores de predicción de m pasos, para predicciones de un paso ($k = 1$) y de 4 pasos ($k = 4$).

Tabla resumen para predicción de un paso

nodo	modelo	MAE	MAPE	MEAN	RMSE
apoq	armax	2.334,1	2.382,8	2.306,7	3.240,9
	garch	32.438,5	735.978,9	-32.036,1	35.794,3
	kalman	160	10.361,5	-87,6	627,8
dehesa	armax	1.142,4	38.398,5	45,6	2.305
	garch	124.645,6	1.548.942,1	12.214,8	166.966,4
	kalman	339,8	9.427,5	-168,2	693,6
pcoya	armax	65.696,4	32.398.509,7	-65.475,4	73.439,5
	garch	1.425.870,8	1.237.716.672,5	-1.405.177,2	1.591.576,8
	kalman	350,7	150.697,3	-124,7	871,8
slucia	armax	137.762,4	6.796.870,6	-136.368,8	157.839
	garch	2.271.710,1	112.312.593,4	-2.206.754,9	2.658.869,6
	kalman	642,1	73.304,1	-383,7	1.335,4

Tabla 4.1.: Métricas de errores de predicción, $k = 1$

Tabla resumen para predicción de 4 pasos

nodo	modelo	MAE	MAPE	MEAN	RMSE
apoq	armax	2.846,5	135.188,0	-2.301,2	3.757,4
	garch	398.088,9	9.198.486,7	-386.722,9	443.138,4
	kalman	691,1	62.544,2	-89,1	2.926,3
dehesa	armax	1.165,1	42.519,3	61,2	2.322,5
	garch	134.335,4	2.047.526,7	32.958,7	188.083,8
	kalman	1.943,7	38.929,2	470,2	3.553,6
pcoya	armax	4.415,7	1.957.235,7	3.965,9	5.674,6
	garch	19.375,2	22.921.032,0	-18.514,1	25.332,8
	kalman	1.675,5	724.063,0	-130,7	4.363,3
slucia	armax	22.845,6	1.450.060,5	-22.150,5	26.795,2
	garch	861.384,7	42.907.032,2	-836.102,2	1.008.552,3
	kalman	2.614,7	438.237,6	-397,0	6.390,4

Tabla 4.2.: Métricas de errores de predicción, $k = 4$

Los resultados de los errores de predicción muestran que los modelos GARCH son inadecuados para proyecciones de los nodos de banda ancha. Los modelos ARMAX presentan errores de predicción de orden similar tanto para proyecciones de uno o 4 pasos, en el caso de los nodos apoq y dehesa que atienden a usuarios de similares características, que cursan más tráfico que los usuarios de los nodos pcoya y slucia.

Los filtros de Kalman presentan menores errores de predicción, no para todos los casos, como se muestra en la tabla de errores para 4 pasos, donde son mayores a los errores de los modelos ARMAX, para el nodo dehesa en las métricas MAE, MEAN y RMSE.

Las métricas de errores de los modelos ARMAX y filtros de Kalman no aumentan en todos los casos desde proyecciones de un paso a pronósticos de 4 pasos.

4.1.4. Correlación cruzada

Los gráficos de correlación cruzada corresponden al nodo apoq, para predicción de uno y 4 pasos. La correlación se realiza entre la serie de validación y la serie estimada por los modelos ARMAX, GARCH y filtro de Kalman. La correlación se muestra normalizada y para la misma duración de las secuencias de observaciones.

Correlación cruzada para predicción de un paso

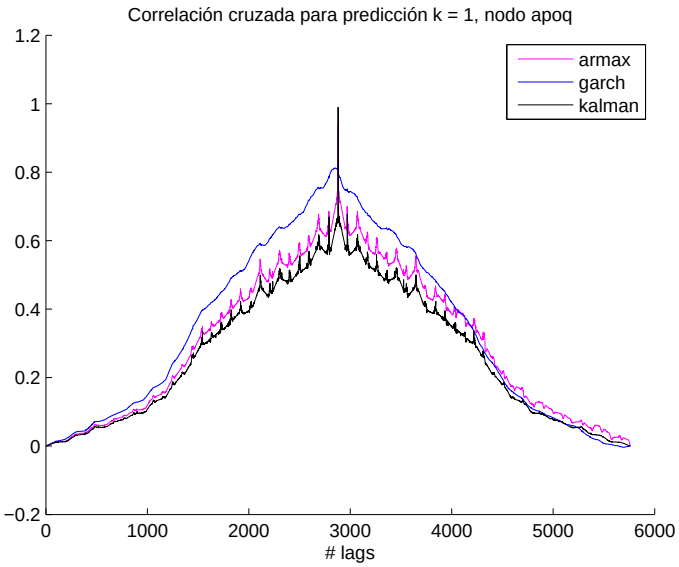


Figura 4.1.: Correlación cruzada de serie de validación y estimada, $k = 1$, nodo apoq

Correlación cruzada para predicción de 4 pasos

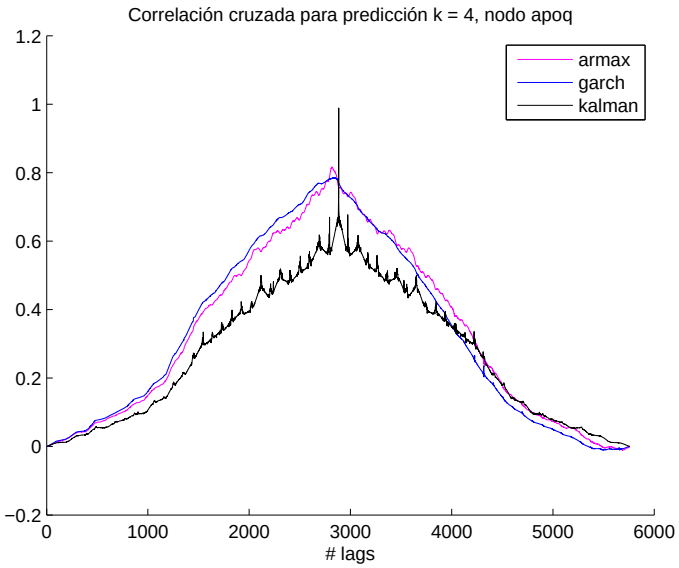


Figura 4.2.: Correlación cruzada de serie de validación y estimada, $k = 4$, nodo apoq

Las correlaciones cruzadas muestran que en las proyecciones de un paso las correlaciones de las series de validación y estimada son similares, para los modelos ARMAX y filtros de Kalman. Las proyecciones de 4 pasos muestran baja correlación (entre series de validación y estimada) en el caso de los modelos ARMA, tanto ARMAX como GARCH.

4.1.5. Test de Kolmogorov-Smirnov

Los tests de hipótesis permiten decidir entre distintos supuestos. Si se considera una serie $X_1, \ldots, X_n$ con una distribución $\mathbb{P}$, se busca evaluar la hipótesis si es igual a una distribución $\mathbb{P}_0$, es decir, evaluando entre las hipótesis

$$H_0 : \mathbb{P} = \mathbb{P}_0, \quad H_1 : \mathbb{P} \neq \mathbb{P}_0$$

Si $F(x) = \mathbb{P}(X_1 \leq x)$ es la distribución de las observaciones, se define una distribución empírica, c.d.f, por

$$F_n(x) = \mathbb{P}_n(X \leq x) = \frac{1}{n} \sum_k^n I(X_k \leq x)$$

que da cuenta de la proporción de las muestras bajo x. Para cualquier punto $x \in \mathbb{R}$, y de la ley de los grandes números se tiene

$$F_n(x) = \frac{1}{n} \sum_k^n I(X_k \leq x) \to \mathbb{E}\{I(X_1 \leq x)\} = \mathbb{P}(X_1 \leq x) = F(x)$$

donde la proporción de muestras en el conjunto $(-\infty, x]$ aproxima las probabilidades del conjunto. Si la aproximación es uniforme para todo $x \in \mathbb{R}$, entonces

$$\sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \to 0 \tag{4.1}$$

la mayor diferencia entre F_n y F tiende a 0 en probabilidad.

El test de Kolmogorov-Smirnov se basa en que la distribución del supremo no depende de la distribución de las muestras $\mathbb{P}$, que no se conoce, cuando $\mathbb{P}$ es continua [17].

Test de Kolmogorov-Smirnov para dos series

Para las serie $X_1, \ldots, X_m$ de tamaño m con c.d.f. $F(x)$ y $Y_1, \ldots, Y_n$ de tamaño n con c.d.f. $G(x)$, se evalua

$$H_0 : \mathbb{F} = \mathbb{G}, \quad \text{vs.} \quad H_1 : \mathbb{F} \neq \mathbb{G}$$

Si $F_m(x)$ y $G_n(x)$ son las distribuciones empíricas, c.d.f.s, entonces la estadística

$$D_{mn} = \left(\frac{mn}{m+n} \right)^{1/2} \sup_x |F_m(x) - G_n(x)|$$

también satisface la aproximación entre las series, en probabilidad, como antes para F_n y F (ecuación 4.1). La descripción de los teoremas del test en Anexos C.2

Los test de Kolmogorov-Smirnov se aplican para las series de validación y de proyecciones, para análisis de que modelos permiten aproximaciones de distribuciones empíricas, desde las proyecciones a las observaciones de validación [14].

4.1.6. Test Kolmogorov-Smirnov para proyecciones, por modelo

Esquema del test

Con $h = 0$ si se acepta la hipótesis de que ambas series, de validación y de proyecciones, siguen una distribución empírica similar. Si $h = 1$ la hipótesis se rechaza.

Los valores p-value y k-test corresponden a las estadísticas del test

- p-value: valor de referencia del test, para aceptación o rechazo, dado umbral de referencia, de 0,05.
- k-test: valor de la distribución de Kolmogorov, que representa la mayor diferencia, normalizada, entre series.

Tabla resumen para predicción de un paso

Los resultados del test Kolmogorov-Smirnov, entre las series de validación y de proyecciones, por modelo, son:

nodo	modelo	h	p-value	k-stat
apoq	armax	1	0	0,76
	garch	1	0	0,95
	kalman	0	0,09	0,03
dehesa	armax	1	0	0,18
	garch	1	0	0,51
	kalman	1	$5{,}74{\cdot}10^{-5}$	0,06
pcoya	armax	1	0	0,95
	garch	1	0	0,96
	kalman	0	0,37	0,02
slucia	armax	1	0	0,96
	garch	1	0	0,93
	kalman	1	$2{,}91{\cdot}10^{-4}$	0,06

Tabla 4.3.: Test Kolmogorov-Smirnov, predicción $k = 1$

Tabla resumen para predicción de 4 pasos

nodo	modelo	h	p-value	k-stat
apoq	armax	1	0	0,47
	garch	1	0	0,93
	kalman	0	0,08	0,03
dehesa	armax	1	0	0,19
	garch	1	0	0,54
	kalman	1	0	0,22
pcoya	armax	1	0	0,58
	garch	1	0	0,73
	kalman	0	0,33	0,03
slucia	armax	1	0	0,58
	garch	1	0	0,92
	kalman	1	$2{,}91 \cdot 10^{-4}$	0,06

Tabla 4.4.: Test Kolmogorov-Smirnov, predicción $k = 4$

Los filtros de Kalman presentan la menor distancia entre las distribuciones empíricas de las series de validación y estimada. En 4 casos, que corresponden a los nodos apoq y pcoya, de uno o 4 pasos de predicción, se acepta la hipótesis de similitud de las distribuciones empíricas de las series estimada y de validación.

Los resultados del test Kolmogorov-Smirnov para los modelos ARMA, en todos los casos, son de rechazo de la hipótesis en análisis, es decir, que las series de validación y estimada corresponden a distribuciones empíricas similares.

4.2. Discusión de resultados

4.2.1. Comparación de modelos

Coeficientes y grados de los modelos

- En ARMAX, sólo se tiene disponible AIC, en cambio en GARCH (las funciones garchfit en conjunto con aicbic), se tienen disponibles AIC y BIC. Las referencias indican [3] que AIC, sobrestima (p,q), en comparación a BIC, como los resultados descritos en tabla 3.4, al comparar cantidad de grados de los modelos.
- Los modelos GARCH se construyen con más restricciones que los modelos ARMAX
  - Se explícita una varianza constante nula, GARCH(0,0).
  - Requieren más de tiempo, en iteraciones, al incluir más variables, en cada iteración. La cantidad de coeficientes son enumerados con la función *garchcount*.

Representación como filtros de tiempo discreto

- Para poder comparar, con idénticas secuencias de validación, los modelos ARMA se representan en forma común utilizando la función *filter*, en dominio *Z*.
- Las ventajas de la representación con filtros son
  - Análisis de la estabilidad de los modelos ARMA a través de la representación de filtros, al revisar si los polos y ceros de los filtros se encuentran dentro del círculo unitario. La estabilidad de los filtros y su respuesta al impulso sirve para descartar modelos ARMA inestables, que realizan predicciones incorrectas, como se verifica para los modelos GARCH.
  - Análisis en dominio de frecuencia, de manera directa.
  - Comparación directa de las distintas salidas de los filtros, correspondiente a los distintos modelos, en cuanto a estadísticas como:
    - Desviación de la señal de salida v/s señal de entrada, con serie de validación, idéntica para los distintos modelos: ARMAX, GARCH y filtro de Kalman.
    - Otras mediciones, como correlaciones, varianzas, etc., entre señal de salida y señal de validación.
  - Entre otros posibles ítems.

4.2.2. Proyecciones

Modelos ARMA

- Cada toolbox, tanto de Identificación de Sistemas (IS), como de Econometría, tienen salidas de distintos objetos, además de distintas funciones para realizar forecasting
  - En ARMAX (IS) se tienen las funciones *predict* y *compare*.
  - En GARCH+ARMAX se tiene la función *garchpred* con foco en predicción de modelos GARCH, no ARMA, o derivados (con funciones armax), descartándose el uso de esta función.
  - Se utiliza el framework de ARMAX (IS) para generar los objetos idpoly desde los modelos GARCH, con el objetivo de realizar proyecciones de m pasos utilizando la función *predict* basada en modelos de espacio de estado [2.7], en formato de filtros de Kalman.

Filtros de Kalman

- Las métricas de errores absolutos de las proyecciones con filtros de Kalman, de uno o 4 pasos, son menores que los modelos ARMA.
- Los modelos ARMA no son dinámicos en comparación a los filtros de Kalman.
- Para los test Kolmogorov-Smirnov, entre series de verificación y estimaciones, se obtiene en algunos casos, aceptación de hipótesis de similitud de las series.
- La implementación de los filtros de Kalman, en comparación con ARMA, utiliza menos restricciones, en cuanto a inicialización del sistema.
- El análisis de las proyecciones, de uno ó 4 pasos, muestra que al incrementar los horizontes de predicción, los errores absolutos aumentan y el orden de los errores relativos se mantiene similar a los pasos de predicción previos.
- Las características del filtro de Kalman son:
  - Utilización de modelo de espacio de estado.
  - Estructura recursiva.
  - Para los errores de observación se asume una distribución normal.
  - El modelo de espacio de estado es estacionario si la matriz de estado (transición) del filtro (ecuación [2.14]) tiene sus eigenvalores dentro del círculo unitario.

4.3. Extensiones del trabajo

La combinación de filtros adaptivos con modelos de series de tiempo de procesos espaciales (STARMA) pueden utilizarse para la implementación de estadísticas espacio-temporales como:

- Proyecciones multivariadas, en espacio-tiempo, por nodo de red.
- Modelos de tráfico por zona geográfica.

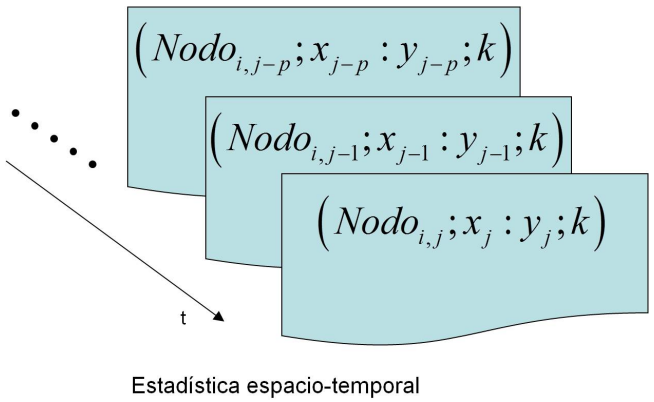


Figura 4.3.: Estadística espacio-temporal para proyecciones de k pasos

En las siguientes secciones se describen los filtros adaptivos y modelos STARMA como extensiones de este trabajo desde los filtros de Kalman y modelos de series de tiempo discreto.

4.3.1. Filtros adaptivos

En general, las series de tiempo, y su representación de filtros, de manera recursiva, pueden describirse como filtros adaptivos recursivos de mínimos cuadrados, Recursive Least-Squares RLS, así como en otros formatos de filtros adaptivos. Los filtros RLS permiten la resolución de problemas de minimización.

Los filtros de Kalman pueden representarse como filtros RLS [21]. La equivalencia de un problema de estimación lineal de mínimos cuadrados, con filtros RLS, es bi-direccional y se basa en la forma estocástica y determinística de los sistemas lineales.

Sistema estocástico	Sistema determinístico
Variables aleatorias {x,y}	Variables determinísticas {x,y}
Modelo $y = Hx + v$	Modelo $y = Hx + v$
Matriz de covarianza R_x	Matriz de regularización inversa Π^{-1}
Covarianza del ruido R_v	Matriz de pesos inversa W^{-1}
$\hat{x}$	$\hat{x}$
min $E(x - Ky)(x - Ky)^*$	min $[x^*\Pi x + \parallel y - Hx \parallel_w^2]$
$\hat{x} = [R_x^{-1} + H^* R_v^{-1} H]^{-1} H^* R_v^{-1} y$	$\hat{x} = [\Pi + H^* W H]^{-1} H^* W y$
m.m.s.e $= [R_x^{-1} + H^* R_v^{-1} H]^{-1}$	min costo $= y^* [W^{-1} + H\Pi^{-1} H^*]^{-1} y$

Tabla 4.5.: Equivalencia de sistemas estocásticos y determinísticos

Los filtros RLS pueden expresarse como un caso especial de filtros de Kalman, además versiones extendidas de los filtros RLS que permiten una equivalencia más general.

Una colección de variables aleatorias escaladas de media cero $\{y(k), 0 \leq k \leq N\}$ con modelo de espacio de estado

$$\begin{aligned} x_{k+1} &= \lambda^{-1/2} x_k \\ y(k) &= u_k x_k + v(k) \end{aligned}$$

con $E\{x_0\} = 0$, $E\{x_0 x_0^*\} = H_0$, $E\{v(k)v^*(i)\} = \delta_{ki}$ y $E\{x_0 v^*(k)\} = 0$ representan un caso específico de un modelo de espacio de estado general, de variables

$$F_k = \lambda^{-1/2} I, \quad G_k = 0, \quad R_k = 1 \quad \text{y } H_k = u_k \text{ (vector fila)}$$

donde λ es un escalar positivo, tal que $0 \ll \lambda \leq 1$. Al aplicar las expresiones de filtro de Kalman al modelo de espacio de estado para minimizar el sistema lineal, se obtiene una equivalencia entre los filtros Kalman y RLS.

Ambos filtros, Kalman y RLS, pueden presentar problemas de estabilidad [11], en estimación de matrices de error de covarianza, P_k, por ello se recomienda factorización en matrices triangulares y diagonales.

La siguiente tabla muestra las correspondencias entre ambos sistemas de filtros.

Nombre variables	Variable Kalman	Variable RLS	Descripción
Mediciones	$y(k)$	$\frac{d(k)}{\left(\sqrt{\lambda}\right)^k}$	Señal de referencia
Vector de estado	x_k	$\frac{w}{\left(\sqrt{\lambda}\right)^k}$	Vector entrada
Estimador de estado	$\hat{x}_{k+1\|k}$	$\frac{w_k}{\left(\sqrt{\lambda}\right)^{k+1}}$	Vector estimado
Error de covariancia	$P_{k+1\|k}$	$\lambda^{-1}P_k$	Inversa matriz de coeficientes
Vector ganancia	$\frac{k_{i(k)}}{r_e(k)}$	$\lambda^{-1/2}g_k$	Vector ganancia
Innovaciones	$v(k)$	$\frac{e(k)}{\left(\sqrt{\lambda}\right)^k}$	Error a priori
Innovaciones	$v(k)$	$\frac{r(k)r_e(k)}{\left(\sqrt{\lambda}\right)^k}$	Error a posteriori
Varianza de innovaciones	$r_e(k)$	$\gamma^{-1}(k)$	Inversa factor de conversión
Condición inicial	$\hat{x}_{0\|-1}$	w_{-1}	Condición inicial
Covarianza inicial	Π_0	$\lambda^{-1}\Pi^{-1}$	Matriz de regularización

Tabla 4.6.: Correspondencias entre variables de filtros RLS y Kalman

4.3.2. Estadísticas espacio-temporales

Modelos STARMA, series de tiempo de procesos espaciales

Se utiliza D_k como conjunto de índice valores de $Y(\cdot)$, proceso espacial, con $\triangle_k$ incrementos discretos de tiempo [33]

$$\begin{aligned} Y_0(\cdot) &\equiv \{Y(s;0):\ s \in D_s\} \\ Y_1(\cdot) &\equiv \{Y(s;1):\ s \in D_s\} \\ &\cdots \\ Y_k(\cdot) &\equiv \{Y(s;k):\ s \in D_s\} \end{aligned}$$

con $D_k = \{0, 1, 2, \ldots\}$, en unidades de $\triangle_k$.

De esta forma, una serie de tiempo de un proceso espacial, dados D_s dependiente de k, es

$$\{Y_k(\cdot):\ k = 0, 1, \ldots\}$$

El proceso espacio-temporal, se describe como un proceso espacial mutivariable, de esta manera, se describen n-variadas series de tiempo

$$\{Y_k(s_1) : \ i = 1, \ldots, n \, ; \ k = 0, 1, \ldots\}$$

Modelos Vector AutoRegresivos, VAR

Una serie multivariada, para $k = 1, ..., T$

$$Y_0, Y_1, ..., Y_T$$

puede presentar dependencia temporal, siguendo un modelo de Markov, esto es

$$[Y_k|Y_0, ..., Y_{k-1}] = [Y_k|Y_{k-1}] \qquad k = 1, 2, ...$$

que no muestra dependencia espacial. Para reconocer dependencia espacial, pueden utilizarse relaciones de vecindad espacial. El modelo autoregresivo, AR, de dimensión n

$$Y_k = M \cdot Y_{k-1} + \eta_k$$

contempla M de n^2 parámetros y $\sum_\eta = var(\eta_k)$ de $O(n^2)$ parámetros.

El contexto espacial puede utilizarse para reducir la cantidad de parámetros, con relaciones de vecindad, por ejemplo, utilizando el supuesto que (i, j) de M, $M_{ij} = 0$, salvo para $||s_i - s_j|| \leq h$, donde h representa un radio de cercanía espacial, para los tiempos $k - 1$ y $k+1$, en los valores pasados de s_i y cercanos en s_j, en radio h. El modelo de Y_k representa un localidad, pero no como es afectado por $(n+1)$ localidades s, por lo que no podría utilizarse para predicciones fuera de la localidad s.

La generalización del modelo vector autoregresivo correponde a modelos espacio-temporal autoregresivos moving-average, STARMA. Los modelos STARMA representan series multivariadas espacio-temporales, de cantidades relevantes de parámetros.

Modelos STARMA

Los procesos estocásticos espacio-temporales discretizados se describen como series multivariadas, $\{Y_k\}$, que siguen modelos VAR, que se generalizan en modelos STARMA, descritos como

$$Y_k = \sum_{k=0}^{p} \left(\sum_{j=1}^{\lambda_i} f_{ij} \cdot U_{ij} \right) Y_{k-i} + \sum_{\ell=0}^{q} \left(\sum_{j=1}^{\mu_\ell} g_{\ell j} \cdot V_{\ell j} \right) W_{k-\ell}$$

con $\{U_{ij}\}$ y $\{V_{\ell j}\}$ matrices conocidas de ponderación, p y q representan los grados de los modelos componentes, autoregresivo y moving-average, respectivamente. $\{f_{ij}\}$ y $\{g_{\ell j}\}$ representan los parámetros del modelo, y $\{W_k\}$ son vectores aleatorios iid, con media 0 y matriz de covarianza $\sum_w$. Para simplificar los modelos, el índice j puede ser re-parametrizado, la nueva estructura es

$$Y_k = \sum_{k=0}^{p} B_i \cdot Y_{k-i} + \sum_{\ell=0}^{q} E_\ell \cdot W_{k-\ell}$$

donde $\sum_w = \sigma_w^2 \cdot I$, y B_0 ceros bajo diagonal, con supuesto $(I - B_0)$ invertible.

El número de parámetros de Y_k puede reducirse considerando los siguientes casos simples.

Primer caso $p = 0$ y $q = 0$ reduce la secuencia a un proceso espacial, sin dependencias temporales

$$Y_k = B_0 \cdot Y_k + E_0 \cdot W_k \qquad k = 0, 1, \ldots$$

esta expresión puede re-escribirse como

$$Y_k = (I - B_0)^{-1} \cdot E \cdot W_k \qquad k = 0, 1, \ldots$$

donde $\{W_0, W_1, \ldots\}$ son mutuamente independientes, en este caso Y_k no tiene dependencia temporal

Segundo caso

$p = 1$ y $q = 0$ y B_0 matriz diagonal de ceros, así

$$Y_k = B_0 \cdot Y_k + B_1 \cdot Y_{k-1} + E_0 \cdot W_k \qquad k = 0, 1, \ldots$$

dado Y_{k-1}, Y_k tiene dependencia estadística espacial, que puede expresarse como un campo aleatorio de Markov, de dependencia condicional simple, como

$$\begin{aligned} Y_k &= (I - B_0)^{-1} \cdot B_1 \cdot Y_{k-1} + (I - B_0)^{-1} \cdot E_0 \cdot W_k \\ &\equiv MY_{k-1} + \eta_k \end{aligned}$$

con $M \equiv (I - B_0)^{-1} \cdot B_1$ y $\{\eta_k\}$ iid, media cero y $var(\eta_k) = \sigma_w^2 \sum_\eta (I - B_0)^{-1} E_0 E_0' (I - B_0')^{-1}$. La matriz B_0 representa una dependencia espacial 'instántanea'.

Tercer caso

$p = 1$, $q = 1$ y $B_0 \equiv 0$, entonces

$$Y_k = B_1 \cdot Y_{k-1} + E_0 \cdot W_k \qquad k = 0, 1, \ldots$$

que equivale a

$$Y_k = MY_{k-1} + \eta_k$$

donde $M \equiv B_1$ y $\{\eta_k\}$ son iid con media cero y $var(\eta_k) = \sigma_w^2 E_0 E_0'$. La expresión de Y_k corresponde a un modelo VAR(1).

5. Conclusiones

5.1. Aportes del trabajo

Los aportes de este trabajo son:

- Comparación de distintas metodologías de construcción de modelos de tráfico, con técnicas de distintos dominios que no se habían utilizado en un problema común como modelos para pronósticos de tráfico de redes de datos.
- Se implementa una nueva revisión de distintos modelos de series de tiempo en el contexto de planificación de sistemas de comunicaciones, que permite tener distintas alternativas de evaluación de modelos de tráfico.
- En contraste a las restricciones actuales de evaluación de sistemas de comunicaciones, como construcción de parámetros de desempeño, la metodología propuesta permite utilizar métricas de evaluación de los modelos desde dominios de análisis de señales y series de tiempo.
- La presentación de los modelos ARMA como combinación de filtros IIR y FIR permite analizar convergencia y estabilidad de los distintos modelos, además de la construcción de una metodología común para proyecciones. La metodología de pronósticos de tráfico utilizando filtros de tiempo discreto también permite la comparación de desempeño con filtros de Kalman.

5.2. Resumen de modelos ARMA y filtros de Kalman

La construcción de los modelos ARMA y filtros de Kalman permite:

1. Las estimaciones de tráfico con los modelos ARMA [A.4] en formatos ARMAX y GARCH y filtros de Kalman [3.8], permiten la proyección de X_{k+m} a partir de las observaciones X_k.
2. Los modelos de las series de observaciones de tráfico, X_k, se describen como filtros de tiempo discreto [A.3.1].
3. Las series de observaciones de tráfico, X_k, son filtradas a partir de los distintos modelos ARMA. La utilización de filtros permite generar un método común para proyecciones.
4. Los distintos modelos para proyecciones de tráfico, tanto ARMA como filtros de Kalman, se evaluan y comparan con las siguientes métricas:

a) Métricas de errores de predicción [4.1.3].

b) Correlación cruzada, entre serie de validación y serie estimada.

c) Test Kolmogorov-Smirnov, para evaluación de hipótesis de similitud, entre serie de validación y serie estimada [4.1.6].

5.3. Desarrollo

Las conclusiones del desarrollo son:

- El procesamiento de las observaciones de tráfico, en cuanto a: filtrado, normalización temporal y regularización de observaciones nulas con información de más series de tráfico, del mismo nodo, facilita las pruebas de distintos modelos, aún no siendo actividades de corto plazo.
- La iteración de distintos modelos y validación con proyecciones, permite la introducción de filtros para análisis de los modelos, junto a la verificación de las proyecciones. Se plantea como alternativa de nuevos trabajos la generalización de esta aproximación vía filtros adaptivos [4.3.1], para la generación de modelos de tráfico de redes.
- Se utiliza la misma metodología para las proyecciones de los distintos modelos ARMA con independencia de su formulación. Las proyecciones se basan en sistemas dinámicos lineales, en diversos formatos de filtros.

5.4. Resultados y discusión

Los resultados permiten señalar que la selección de los parámetros, de los modelos de proyección, afectan los desempeños de los distintos modelos, de las series de tráfico utilizadas.

5.4.1. Errores de predicción

La cuantificación de los errores entre los modelos y las series de validación puede clasificarse como:

- Errores en los parámetros de los modelos.
- Errores en las salidas de los modelos y las series de validación.
- Error a señales de prueba entrada/salida, por ejemplo los errores a la respuesta al impulso.

Los errores pueden evaluarse como:

- Errores absolutos.
- Errores relativos.
- Valores medios, ejemplos: lineales o cuadráticos.

Los resultados comprenden los errores para las pruebas de los modelos, con series de verificación, para proyecciones de uno o 4 pasos.

Para los 4 nodos del estudio, con sus respectivas series de validación, se tiene que los filtros de Kalman generan los menores errores absolutos y que para los errores relativos, en comparación a los modelos ARMAX, hay casos en que son mayores. En cuanto a los modelos GARCH, en todos los casos, los 4 nodos tienen los mayores errores entre las series de validación y de salidas de los modelos.

Estabilidad

La respuesta al impulso, en el caso de errores con señales de validación, además de la verificación de si polos y ceros se encuentran dentro del círculo unitario, para estabilidad de los filtros, permite a priori descartar los modelos GARCH que presentan los mayores errores de predicción, al tener polos y ceros fuera del círculo unitario y respuesta infinita al impulso.

5.4.2. Grados de los modelos

La determinación de los grados de los modelos ARMA puede realizarse con distintos criterios, tales como:

- Función de costos.
- Errores residuales.
- Polos y ceros.
- Métodos de test de hipótesis.
- Criterios de información.

Los métodos automatizados de estimación de los grados de los modelos ARMA, para ARMAX y GARCH, corresponden a los criterios de información AIC y BIC, los cuales equivalen asintóticamente a otros métodos como test de hipótesis y funciones de error. De esta manera, con BIC y AIC se estiman los grados de modelos ARMA que posteriormente se utilizan en la representación de filtros de los modelos. Las series AR que componen los modelos GARCH pueden utilizarse para realizar proyeciones con menores errores de proyección.

5.4.3. Correlación

Si la auto-correlación de la serie de error de predicción es nula, para $k \neq 0$, entonces la señal de error es independiente de las series de validación. La utilización de series de verificación distintas a las series de construcción de los modelos permite la *validación cruzada* al evaluarse la correlación cruzada entre series de validación y salidas de los modelos.

La correlación cruzada es otra alternativa de validación de los modelos ARMA y filtros de Kalman. Se verifica que las salidas de los filtros de Kalman aproximan relativamente a las series de validación, en comparación a los modelos ARMA, para proyecciones de un paso y que para proyecciones de 4 pasos, los modelos tienen menor relación con las series de verificación.

5.4.4. Test de distribuciones empíricas

Las hipótesis de similitud de las distribuciones empíricas de las series de validación y proyecciones, en formato de test de Kolmogorov-Smirnov, representan otra vía de evaluación y comparación de los distintos modelos ARMA y filtros de Kalman.

- Los resultados del test Kolmogorov-Smirnov confirman para los filtros de Kalman, en pocos casos (4 de 24), aceptación de la hipótesis de distribución similar a las series de validación y rechazan la hipótesis para los modelos ARMA.
- La distancia entre las series de proyecciones y de comprobación es mayor, en la mayoría de las evaluaciones, en los modelos GARCH, y menor en el caso de los filtros de Kalman.

5.4.5. Descripción de próximos trabajos

Las propuestas de nuevos trabajos con uso de filtros adaptivos y modelos STARMA, secciones [4.3.1] y [4.3.2] respectivamente, se enumeran a continuación:

- Utilización de filtros adaptivos para proyecciones de tráfico de nodos de red de m pasos.
- Extensión de los modelos de series de tiempo para inclusión de variables espaciales, modelos STARMA.
- Para una región geográfica y con nodos de tecnología de distintos tipos de accesos (inalámbricos, por cable, etc.), realizar modelos y proyecciones de tráfico espacio-temporales. La zona geográfica puede ser atendida, en forma conjunta, por nodos de red de distintas tecnologías.

5.5. Epílogo

El uso de las metodologías de series de tiempo y filtros digitales para modelamiento de tráfico en nodos de redes de datos son generales y no presentan dependencia de las distintas tecnologías de redes de comunicaciones, como accesos de fibra, accesos inalámbricos, xDSL, etc. Las observaciones de tráfico utilizadas en el trabajo corresponden a nodos DSLAM (accesos xDSL), aunque pueden utilizarse datos de tráfico de otros tipos de nodos de redes, no sólo de acceso, sino también de transporte, de core, etc.

Los resultados comprenden la comparación de los modelos ARMAX, GARCH y filtros de Kalman para predicciones de tráfico de uno o cuatro pasos. Las métricas de desempeño, como errores de predicción y test de hipótesis de correlación, entre la serie estimada y la de verificación, se aplican a los modelos de tráfico de los nodos de red de banda ancha. Los filtros de Kalman, en la mayoría de las comparaciones, presentan mejores parámetros de desempeño que los modelos ARMA, y entre los modelos ARMA, la construcción ARMAX tiene los menores índices de error.

Bibliografía

[1] A. Adhikari, L. Denby, J. M. Landwehr y J. Meloche. Using data network metrics, graphics, and topology to explore network characteristics. *Networks*, 54:62–75, 2007.

[2] Peter Bloomfield. *Fourier Analysis of Time Series: An Introduction (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2da edición, Febrero 2000.

[3] Jean-Yves Le Boudec. *Performance Evaluation of Computer and Communication Systems*. EFPL Press, 1era edición, Febrero 2011.

[4] Petrus M.T. Broersen. *Automatic Autocorrelation and Spectral Analysis*. Springer, re-impresión de hardcover 2006, 1era edición, Octubre 2010.

[5] Rizwan Butt. *Applied Linear Algebra and Optimization Using MATLAB (Mathematics)*. Mercury Learning & Information, 1era edición, Agosto 2011.

[6] César Hernández Suarez, Octavio Salcedo y Andrés Escobar. An ARIMA model for forecasting Wi-Fi data network traffic values. *Revista de Ingeniería e Investigación*, 29(29):65–69, 2009.

[7] César Hernández Suarez y Luis Pedraza Martínez. Modelo de tráfico WIMAX basado en series de tiempo para pronosticar valores futuro de tráfico. *Journal of Information Systems and Technology Management*, 5(3):505–525, 2008.

[8] Luis Chaparro. *Signals and Systems using MATLAB*. Academic Press, 1era edición, Octubre 2010.

[9] Chris Chatfield. *The Analysis of Time Series: An Introduction, Sixth Edition (Chapman & Hall/CRC Texts in Statistical Science)*. Chapman and Hall/CRC, 6ta edición, Julio 2003.

[10] Christian Sauer y Matthias Gries. Modular Reference Implementation of an IP-DSLAM. *Proceedings, 10th IEEE Symposium on Computers and Communications*, (ISCC), 2005.

[11] Paulo S. R. Diniz. *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, re-impresión de hardcover 2008, 3ra edición, Octubre 2010.

[12] Greg Welch y Gary Bishop. An Introduction to the Kalman Filter. Technical report, UNC-Chapel Hill, 2006.

[13] James Haught, Kenneth Hopkinson, Nathan Stuckey, Michael Dop y Alexander Stirling. A Kalman Filter-Based Prediction system for better network context awarenesss. In *Winter Simulation Conference*. IEEE, 2010.

[14] Matthew S. Allen, John Brevik y Rich Wolski. Comparing network bandwidth time-series. In *Proceedings of the first international conference on Networks for grid applications*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2007.

[15] Piet Van Mieghem. *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, 1era edición, Abril 2009.

[16] Mohammed Olama, Seddik Djouadi y Charalambos Charalambous. Position and Velocity Tracking in Cellular Networks Using the Kalman Filter. *Advances*, (April), 2009.

[17] Dmitry Panchenko. Curso: Statistics for Applications, sección 13: Kolmogorov-Smirnov test. Technical report, MIT OpenCourseWare, 2006.

[18] Paulo Cortez, Miguel Rio, Miguel Rocha y Pedro Sousa. Multi-scale Internet traffic forecasting using Neural networks and Time Series methods. *Expert Systems*, 29(2):143–155, 2010.

[19] Thomas G. Robertazzi. *Planning Telecommunication Networks*. Wiley-IEEE Press, 1era edición, Diciembre 1998.

[20] S. Gowrishankar. A Time Series modeling and Prediction of Wireless Network Traffic. *Computer Science and Telecommunications*, 2(2):40–52, 2008.

[21] Ali H. Sayed. *Adaptive Filters*. Wiley-IEEE Press, 1era edición, Abril 2008.

[22] Galit Shmueli. *Practical Time Series Forecasting: A Hands-On Guide*. CreateSpace, 2da edición, Diciembre 2011.

[23] RATS software. Descripción de RATS. http://en.wikipedia.org/wiki/Regression_Analysis_of_Time_Series. Acceso Mayo 2012.

[24] Subsecretaría de Telecomunicaciones de Chile - Subtel. Información Estadística anual. http://ref.playsip.org/subtel_junio2012. Acceso Junio 2012.

[25] Cisco Systems. Cisco IOS NetFlow. http://ref.playsip.org/netflow_cisco-ios. Acceso Mayo 2012.

[26] T. Anjali, C. Bruni, D. Iacoviello, G. Koch y C. Scoglio. Filtering and forecasting problems for aggregate traffic in Internet links. *Performance Evaluation*, 58(1):25–42, 2004.

[27] Vassilios C. Moussas, Marios Daglis y Eva Kolega. Network traffic modeling and Prediction using Multiplicative Seasonal ARIMA models. *1st International Conference on Experiments/Process/System/Modeling/Simulation/Optimization*, (1):6–9, 2005.

[28] Wei Ke y Lenan Wu. Mobile Location with NLOS Identification and Mitigation Based on Modified Kalman Filtering. 11(2):1641–56, 2011.

[29] Xue Lianqing y Cheng Guang. Network Traffic Forecast Based on Seasonal Neural Network Model. *Science And Technology*, (2), 2004.

[30] Mohsen Guizani, Ammar Rayes, Bilal Khan y Ala Al-Fuqaha. *Network Modeling and Simulation: A Practical Perspective*. Wiley-Interscience, 1era edición, Abril 2010.

[31] Mohinder S. Grewal y Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-IEEE Press, 3era edición, Septiembre 2008.

[32] Bruce L. Bowerman, Richard O'Connell y Anne Koehler. *Forecasting, Time Series, and Regression (with CD-ROM) (Forecasting, Time Series, & Regression)*. South-Western College Pub, 4ta edición, Abril 2004.

[33] Noel Cressie y Christopher K. Wikle. *Statistics for Spatio-Temporal Data (Wiley Series in Probability and Statistics)*. Wiley, 1era edición, Mayo 2011.

[34] Robert H. Shumway y David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 3era edición, Noviembre 2010.

[35] Scott Miller y Donald Childers. *Probability and Random Processes, Second Edition: With Applications to Signal Processing and Communications*. Academic Press, 2da edición, Enero 2004.

[36] Christiaan Heij, André C.M. Ran y F. van Schagen. *Introduction to Mathematical Systems Theory: Linear Systems, Identification and Control*. Birkhäuser Basel, 1era edición, Noviembre 2006.

[37] George E. P. Box, Gwilym M. Jenkins y Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control (Wiley Series in Probability and Statistics)*. Wiley, 4ta edición, Junio 2008.

[38] Klaus Wehrle, Mesut Günes y James Gross, editor. *Modeling and Tools for Network Simulation*. Springer, 1era edición, Septiembre 2010.

[39] Jyh-ying Peng y John A. D. Aston. State Space Models : A MATLAB Software Implementation for Time Series Analysis by State Space Methods. 2007.

[40] Gunter Bolch, Stefan Greiner, Hermann de Meer y Kishor Shridharbhai Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Interscience, 2da edición, Abril 2006.

[41] Stephen Boyd y Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2da edición, Marzo 2004.

[42] Rolf Isermann y Marco Münchhof. *Identification of Dynamic Systems: An Introduction with Applications (Advanced Textbooks in Control)*. Springer, 1era edición, Diciembre 2010.

[43] Spyros G. Makridakis, Steven C. Wheelwright y Rob J. Hyndman. *Forecasting: Methods and Applications*. Wiley, 3era edición, Diciembre 1997.

[44] Yantai Shu, Minfang Yu, Oliver Yang, Jiakun Liu y Huifang Feng. Wireless traffic modeling and Prediction using Seasonal ARIMA models. *IEICE TRANS.COMMUN*, E88:3992–3999, 2005.

[45] Eric Zivot. State Space Models and the Kalman Filter. Technical report, Foster School of Business, University of Washington, 2006.

A. Anexo A: Modelos ARMA y filtros de Kalman

A.1. Código con función armax

```
% url = http://ref.playsip.org/matlab_codes_arima
% Loading Data %
load data;
% Inputs %
Input = data;
WNLagPeriod = 15;
% White Noise Lag Period %
MaxI = 15;          % Maximum integrand period %
MaxAR = 20;         % Maximum AR lookback %
MaxMA = 20;         % Maximum MA lookback %

% Checking for I(n) process %
VarianceMatrix = [];
for I = 1: MaxI
        VarianceMatrix=[VarianceMatrix; var(diff(Input,I))];
end
IMatrix=find(ismember(VarianceMatrix, min(VarianceMatrix))==1);
if var(Input) < min(VarianceMatrix)
        I = 0;
else
        I = IMatrix;
end

% Making Data Stationary %
if I > 0
        StationaryInput = diff(Input,I);
else
        StationaryInput = Input;
end;

% Testing for White-Noise %
[LBTestH,LBTestValue,LBTeststat,LBTestCriticalValue]=
lbqtest(StationaryInput, WNLagPeriod,[],[]);
clear LBTestValue LBTeststat LBTestCriticalValue WNLagPeriod

if LBTestH == 0
```

```
        disp(['The process is white noise at I =',num2str(I)]);
else
% Selecting AR & MA Process based on AIC %
    Y = [];
        for AR = 1: MaxAR
                for MA = 0: MaxMA
              Y = [Y; aic(armax(StationaryInput,[AR,MA])),AR,MA];
                end
        end
        clear AR MA
        AIC = Y(:,1);
        AR = Y(:,2);
        MA = Y(:,3);
        clear Y;
        RowNumber = find(ismember(AIC, min(AIC))==1);
        ARp = AR(RowNumber);
        MAq = MA(RowNumber);
model = armax(StationaryInput,[ARp,MAq]);
end;

clear LBTestH VarianceMatrix IMatrix MaxAR MaxMA MaxI AIC AR MA
Input RowNumber StationaryInput;
```

A.2. Código con función garch

```
% url = http://ref.playsip.org/matlab_codes_arima
% Loading Data %
load data;
% Inputs %
Input = data;
WNLagPeriod = 15;
% White Noise Lag Period %
MaxI = 15;          % Maximum integrand period %
MaxAR = 20;         % Maximum AR lookback %
MaxMA = 20;         % Maximum MA lookback %

% Checking for I(n) process %
VarianceMatrix = zeros(MaxI,1);
for I = 1: MaxI
VarianceMatrix(I,1) = var(diff(Input,I));
end
IMatrix = find(ismember(VarianceMatrix, min(VarianceMatrix))==1);
if var(Input) < min(VarianceMatrix)
    I = 0;
else
    I = IMatrix;
end
% Making Data Stationary %
```

```
if I > 0
    StationaryInput = diff(Input,I);
else
    StationaryInput = Input;
end;
[row1,~] = size(StationaryInput);

% Testing for White-Noise %
[LBTestH,~,~,~] = lbqtest(StationaryInput, WNLagPeriod,[],[]);

clear LBTestValue LBTeststat LBTestCriticalValue WNLagPeriod;

% Start Iterations %
if LBTestH == 0
    disp(['The process is white noise at I = ', num2str(I)]);
else
    % Selecting AR & MA Process based on BIC %
    Z = zeros(MaxAR + 1,MaxMA+1);
        for MA = 0: MaxMA
                for AR = 0: MaxAR
            i = AR + 1;
            j = MA + 1;
   spec = garchset('R', AR, 'M', MA, 'P', 0, 'Q', 0, 'Display','off');
            [Coeff,~,LLF,~,~,~] = garchfit(spec,StationaryInput);
            Parameters = garchcount(Coeff)
            [BIC,~] = aicbic(LLF,Parameters,row1);
            Z(i,j) = BIC;
                end
        end

clear AR MA spec Coeff LLF AIC CSize i j row1;

[row,column] = find(ismember(Z, min(min(Z),[],2)));
ARp = row - 1; MAq = column - 1;
spec = garchset('R',ARp,'M',MAq,'P',0,'Q',0, 'Display','off');
[FinalCoeff,FinalErrors,~,FinalInnovations,~,FinalSummary]=
garchfit(spec,StationaryInput); garchdisp(FinalCoeff, FinalErrors)
RSquared = 1 - corr(StationaryInput,FinalInnovations)

end;

clear row column  Input RowNumber StationaryInput Parameters LBTestH;
clear VarianceMatrix IMatrix MaxAR MaxMA MaxI BIC AIC AR MA;
```

A.3. Modelos ARMA, desde funciones armax y garch

A.3.1. Expresiones de los modelos con coeficientes

Discrete-time idpoly model

$$
\begin{aligned}
B(z) = {} & 1 - 0{,}2573z^{-1} - 0{,}01851z^{-2} - 4{,}4 \cdot 10^{-5}z^{-3} + 1{,}033 \cdot 10^{-4}z^{-4} \\
& -1{,}931 \cdot 10^{-5}z^{-5} + 1{,}522 \cdot 10^{-6}z^{-6} + 2{,}235 \cdot 10^{-8}z^{-7} \\
& -8{,}902 \cdot 10^{-9}z^{-8} + 9{,}824 \cdot 10^{-10}z^{-9} - 1{,}415 \cdot 10^{-10}z^{-10} \\
& +8{,}129 \cdot 10^{-12}z^{-11} - 1{,}583 \cdot 10^{-13}z^{-12} + 8{,}632 \cdot 10^{-15}z^{-13} \\
& +1{,}158 \cdot 10^{-14}z^{-14} - 1{,}766 \cdot 10^{-15}z^{-15} + 6{,}642 \cdot 10^{-17}z^{-16}
\end{aligned}
$$

$$
\begin{aligned}
A(z) = {} & 1 - 0{,}3394z^{-1} - 0{,}03984z^{-2} + 0{,}001569z^{-3} + 9{,}283 \cdot 10^{-5}z^{-4} \\
& -2{,}729 \cdot 10^{-5}z^{-5} + 3{,}215 \cdot 10^{-6}z^{-6} - 1{,}256 \cdot 10^{-7}z^{-7} \\
& -8{,}204 \cdot 10^{-9}z^{-8} + 1{,}599 \cdot 10^{-9}z^{-9} - 2{,}298 \cdot 10^{-10}z^{-10} \\
& +2{,}141 \cdot 10^{-11}z^{-11} - 9{,}667 \cdot 10^{-13}z^{-12} + 2{,}457 \cdot 10^{-14}z^{-13} \\
& +1{,}191 \cdot 10^{-14}z^{-14} - 2{,}875 \cdot 10^{-15}z^{-15} + 2{,}266 \cdot 10^{-16}z^{-16} \\
& -6{,}801 \cdot 10^{-18}z^{-17} + 7{,}591 \cdot 10^{-20}z^{-18} - 1{,}612 \cdot 10^{-21}z^{-19}
\end{aligned}
$$

Tabla A.1.: Modelo nodo apoq desde función armax

Discrete-time garch model

$$
\begin{aligned}
B(z) = {} & -0,0224z^{-1} + 0,917z^{-2} \\
\\
A(z) = {} & -0,763z^{-1} - 0,914z^{-2} + 0,687z^{-3} - 0,029z^{-4} \\
& +0,046z^{-5} - 0,034z^{-6} + 0,020z^{-7} + 0,034z^{-8} \\
& -0,025z^{-9} + 4,2 \cdot 10^{-4}z^{-10} - 0,007z^{-11} - 0,013z^{-12}
\end{aligned}
$$

Tabla A.2.: Modelo nodo apoq desde función garch

Discrete-time idpoly model

$$
\begin{aligned}
B(z) &= 1 - 2{,}088z^{-1} + 1{,}532z^{-2} - 0{,}4994z^{-3} - 1{,}719z^{-4} \\
&\quad +3{,}606z^{-5} - 2{,}241z^{-6} + 0{,}4522z^{-7} + 0{,}8153z^{-8} \\
&\quad -1{,}575z^{-9} + 0{,}7343z^{-10} + 0{,}004804z^{-11} - 0{,}102z^{-12} \\
&\quad +0{,}06581z^{-13} - 0{,}009874z^{-14} + 0{,}03826z^{-15} \\
&\quad +0{,}005369z^{-16} - 0{,}01238z^{-17} - 0{,}007534z^{-18}
\end{aligned}
$$

$$
\begin{aligned}
A(z) &= 1 - 2{,}174z^{-1} + 1{,}627z^{-2} - 0{,}5488z^{-3} - 1{,}694z^{-4} \\
&\quad +3{,}821z^{-5} - 2{,}479z^{-6} + 0{,}5484z^{-7} + 0{,}694z^{-8} \\
&\quad -1{,}647z^{-9} + 0{,}8527z^{-10}
\end{aligned}
$$

Tabla A.3.: Modelo nodo dehesa desde función armax

Discrete-time garch model

$$
\begin{aligned}
B(z) &= 0,122z^{-1} - 0,966z^{-2} - 0,003z^{-3} + 0,996z^{-4} - 0,125z^{-5} + 0,962z^{-6}
\end{aligned}
$$

$$
\begin{aligned}
A(z) &= -0,026z^{-1} + 1,059z^{-2} + 0,207z^{-3} - 0,808z^{-4} + 0,205z^{-5} \\
&\quad -0,923z^{-6} - 0,079z^{-7} - 0,056z^{-8} - 0,054z^{-9}
\end{aligned}
$$

Tabla A.4.: Modelo nodo dehesa desde función garch

Discrete-time idpoly model

$$
\begin{aligned}
B(z) &= 1 - 1{,}402z^{-1} - 0{,}2037z^{-2} - 0{,}09071z^{-3} + 0{,}697z^{-4} \\
&\quad +0{,}7544z^{-5} + 0{,}234z^{-6} - 1{,}554z^{-7} + 0{,}1243z^{-8} \\
&\quad -0{,}2735z^{-9} + 0{,}9002z^{-10} + 0{,}1851z^{-11} - 0{,}1954z^{-12} - 0{,}1748z^{-13}
\end{aligned}
$$

$$
\begin{aligned}
A(z) &= 1 - 0{,}6518z^{-1} - 0{,}677z^{-2} - 0{,}6204z^{-3} + 0{,}2559z^{-4} \\
&\quad +0{,}928z^{-5} + 0{,}941z^{-6} - 0{,}8618z^{-7} - 0{,}5244z^{-8} \\
&\quad -0{,}6492z^{-9} + 0{,}3979z^{-10} + 0{,}5013z^{-11} + 0{,}1608z^{-12} \\
&\quad -0{,}06674z^{-13} - 0{,}04015z^{-14} - 0{,}02862z^{-15} - 0{,}01102z^{-16} \\
&\quad +0{,}02488z^{-17} - 0{,}06715z^{-18}
\end{aligned}
$$

Tabla A.5.: Modelo nodo slucia desde función armax

Discrete-time garch model

$$\begin{aligned} B(z) &= -0,090z^{-1} + 0,293z^{-2} + 0,225z^{-3} + 0,139z^{-4} + 0,093z^{-5} + 0,053z^{-6} \\ &\quad +0,042z^{-7} + 0,038z^{-8} + 0,003z^{-9} + 0,027z^{-10} - 0,004z^{-11} \end{aligned}$$

$$A(z) = -0,626z^{-1} - 0,369z^{-2}$$

Tabla A.6.: Modelo nodo slucia desde función garch

Discrete-time idpoly model

$$\begin{aligned} B(z) &= 1 - 1{,}992z^{-1} - 0{,}2504z^{-2} + 1{,}815z^{-3} + 0{,}6583z^{-4} \\ &\quad -1{,}051z^{-5} - 0{,}5362z^{-6} - 0{,}03096z^{-7} - 0{,}03834z^{-8} \\ &\quad +0{,}6245z^{-9} - 0{,}1863z^{-10} + 0{,}1022z^{-11} + 0{,}3016z^{-12} \\ &\quad -0{,}4515z^{-13} + 0{,}1017z^{-14} - 0{,}6163z^{-15} + 0{,}6204z^{-16} \\ &\quad -0{,}07717z^{-17} + 0{,}1626z^{-18} - 0{,}1556z^{-19} \end{aligned}$$

$$\begin{aligned} A(z) &= 1 - 1{,}188z^{-1} - 1{,}162z^{-2} + 0{,}8772z^{-3} + 1{,}258z^{-4} \\ &\quad -0{,}07382z^{-5} - 0{,}4668z^{-6} - 0{,}3344z^{-7} - 0{,}4152z^{-8} \\ &\quad +0{,}3063z^{-9} + 0{,}05085z^{-10} + 0{,}08996z^{-11} + 0{,}4093z^{-12} \\ &\quad -0{,}121z^{-13} + 0{,}06781z^{-14} - 0{,}5138z^{-15} + 0{,}07231z^{-16} \\ &\quad -0{,}09442z^{-17} + 0{,}2701z^{-18} + 0{,}05313z^{-19} - 0{,}07578z^{-20} \end{aligned}$$

Tabla A.7.: Modelo nodo pcoya desde función armax

Discrete-time garch model

$$\begin{aligned} B(z) &= +0,213z^{-1} + 0,127z^{-2} + 0,063z^{-3} + 0,060z^{-4} + 0,073z^{-5} \\ &\quad +0,041z^{-6} + 0,028z^{-7} + 0,056z^{-8} + 0,013z^{-9} + 0,009z^{-10} \\ &\quad +0,038z^{-11} + 0,009z^{-12} + 0,071z^{-13} \end{aligned}$$

$$A(z) = -0,993z^{-1}$$

Tabla A.8.: Modelo nodo pcoya desde función garch

A.4. Predicción *m* pasos con modelos ARMA

A.4.1. Predicciones de un paso

Para nodo dehesa, para un paso, con uso de los modelos ARMA, se tiene

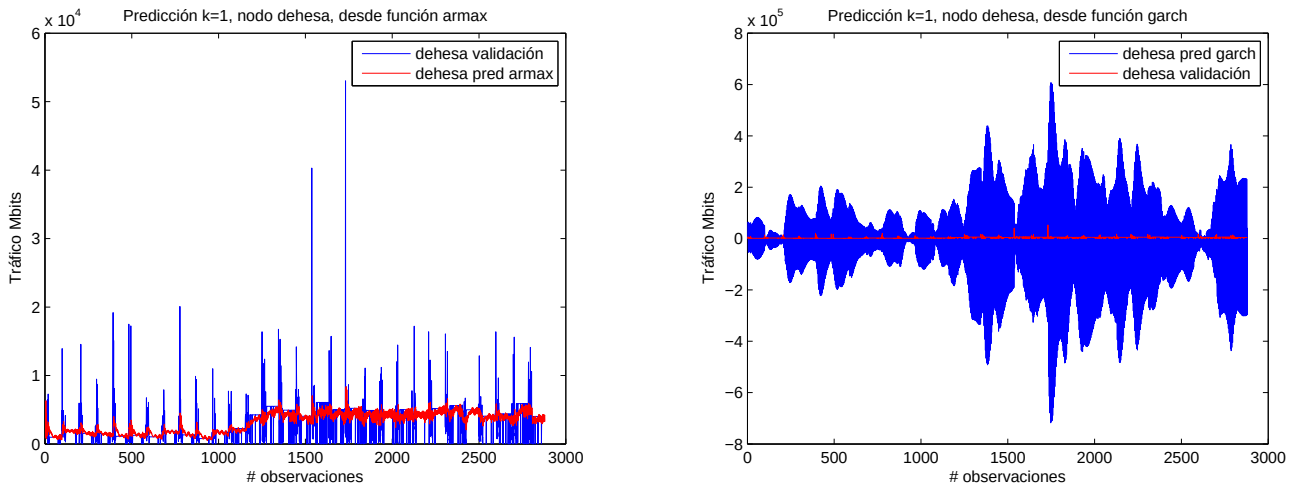


Figura A.1.: Predicción nodo dehesa, un paso, desde funciones armax y garch

Para nodo slucia, para un paso, con uso de los modelos ARMA, se tiene

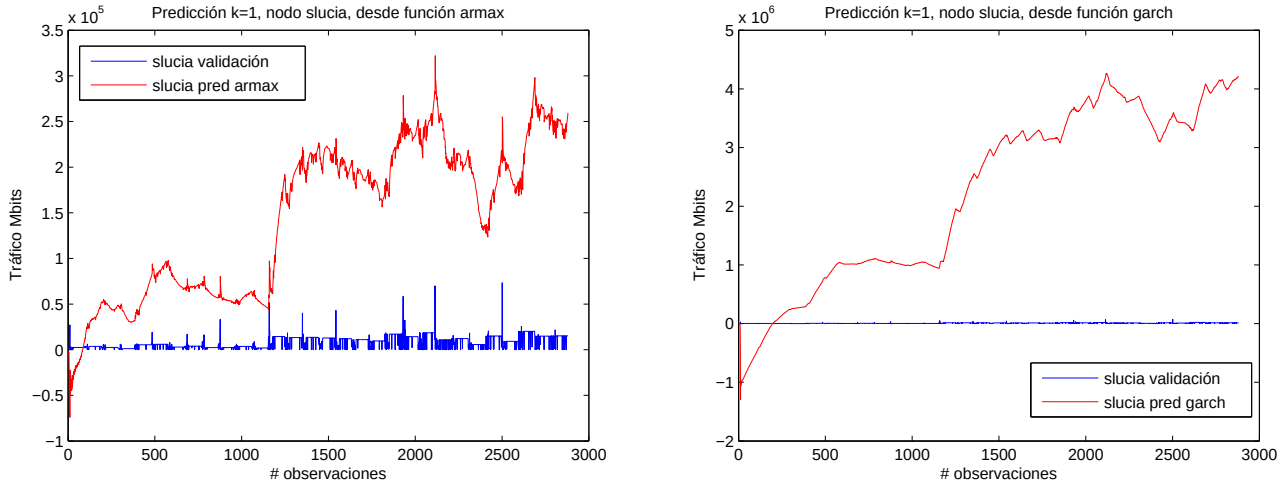


Figura A.2.: Predicción nodo slucia, un paso, desde funciones armax y garch

Para nodo pcoya, para un paso, con uso de los modelos ARMA, se tiene

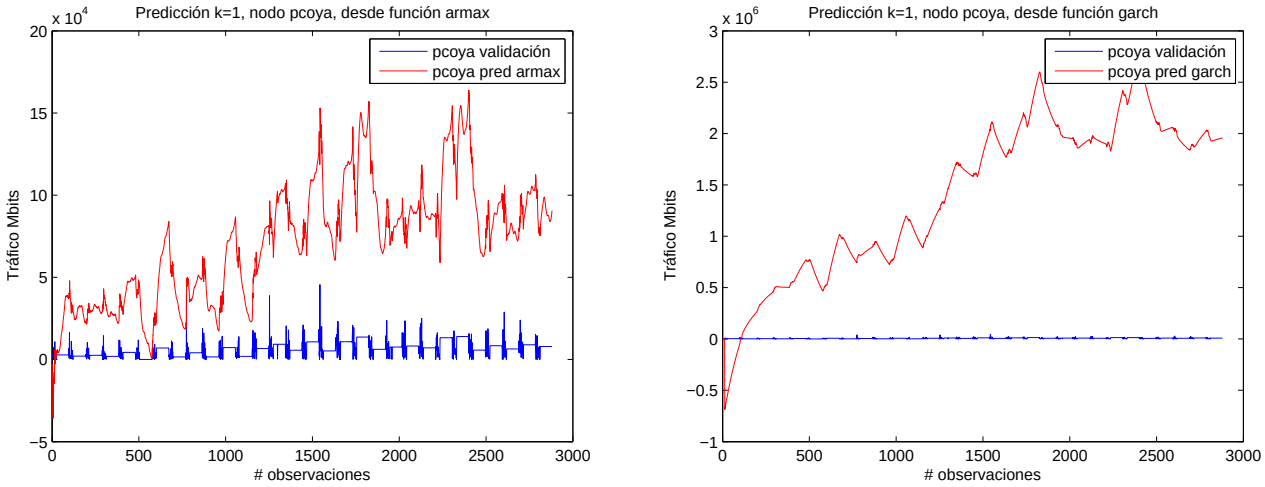


Figura A.3.: Predicción nodo pcoya, un paso, desde funciones armax y garch

A.4.2. Predicciones de 4 pasos

Para nodo dehesa, para 4 pasos, con uso de los modelos ARMA, se tiene

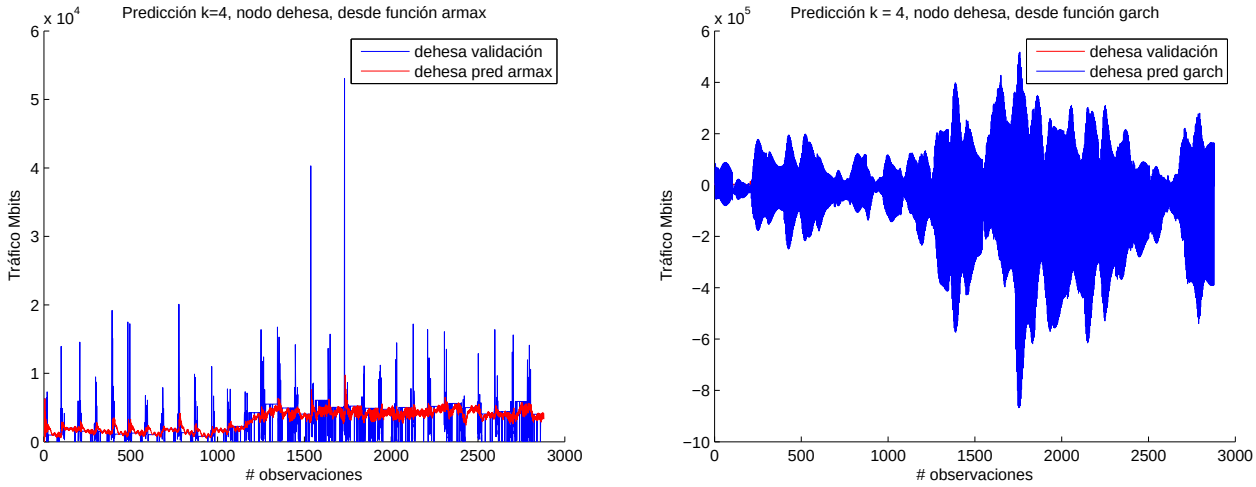


Figura A.4.: Predicción nodo dehesa, 4 pasos, desde funciones armax y garch

Para nodo slucia, para 4 pasos, con uso de los modelos ARMA, se tiene

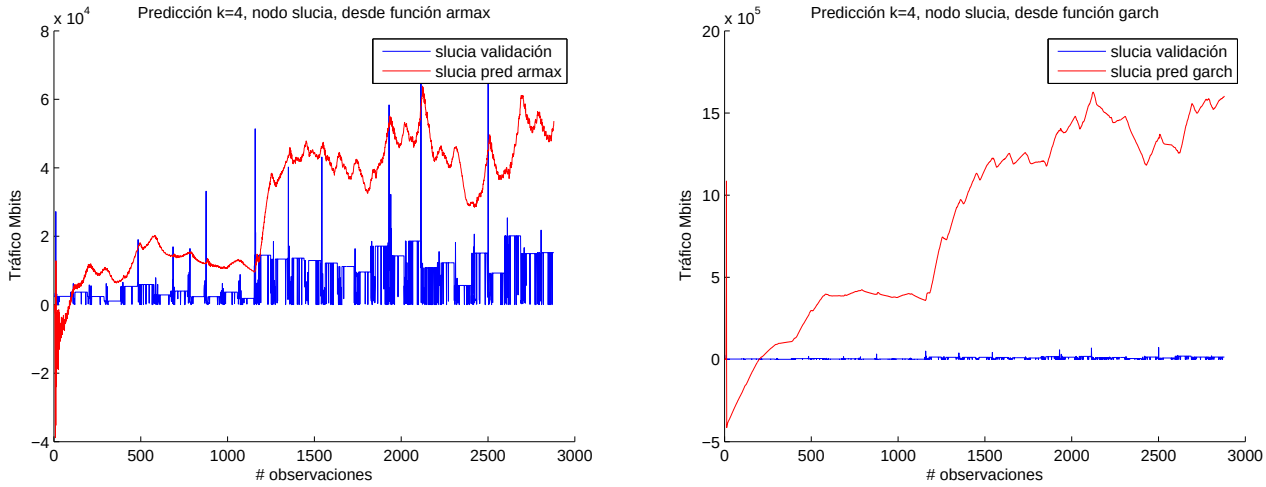


Figura A.5.: Predicción nodo slucia, 4 pasos, desde funciones armax y garch

Para nodo pcoya, para 4 pasos, con uso de los modelos ARMA, se tiene

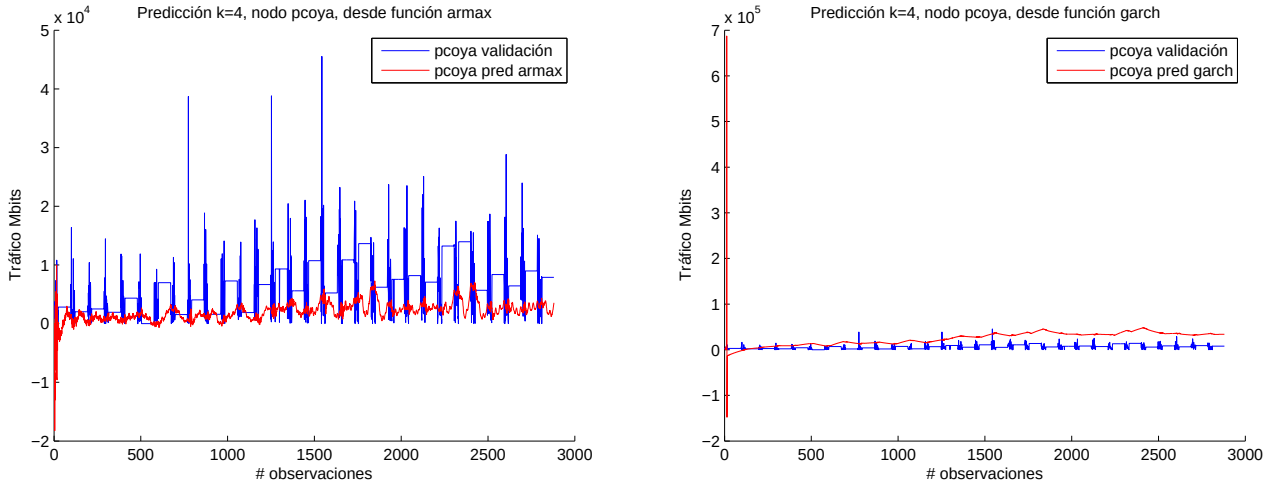


Figura A.6.: Predicción nodo pcoya, 4 pasos, desde funciones armax y garch

A.5. Filtros de Kalman

A.5.1. Código filtro de Kalman

Código de filtro de Kalman, para proyecciones de uno y 4 pasos, ejemplo de nodo apoq.

```
% Copyright 2002-2003 Federal Reserve Bank of Atlanta
% Revisión: 1.2 Date: 19 Marzo 2003
% Autor: Iskander Karibzhanov
% Master of Science in Computational Finance
```

```
% Georgia Institute of Technology
% url = http://www.econ.umn.edu/~karib003/

% nodo apoq %
y = apoq_p';
T = size(y,2);
lead = 0;
Nz = 2;
a = zeros(Nz,1);
F = [1 1; 0 1];
Ny = 1;
b = zeros(Ny,1);
H = [1 0];
var = eye(Ny+Nz)*1e-3;
logl = kalcvf(y, lead, a, F, b, H, var);
z0 = a;
vz0 = eye(Nz)*1e-3;
logl = kalcvf(y, lead, a, F, b, H, var, z0, vz0);
lead = 4;
z0 = a;
vz0 = eye(Nz)*10;
[logl, pred, vpred, filt, vfilt] = kalcvf(y, lead,
a, F, b, H, var, z0, vz0);
y=y';
pred = pred';
pred_ver = pred (1:size(pred));
pred_ver = pred_ver';
filt = filt';
filt_ver = filt (1:size(filt));
filt_ver=filt_ver';
apoq_kalman4=pred_ver;
apoq_kalman4_abs=abs(apoq_kalman4);
figure;
hold on;
ho=plot(apoq_p,'m-');
hp=plot(apoq_kalman4_abs,'b-');
legend([ho hp],'apoq validación','pred kalman');
title('Aplicación filtro Kalman, para predicción,
k = 4, nodo apoq');
xlabel('# observaciones');
ylabel('Tráfico Mbits');
hold off;
```

A.5.2. Predicciones filtro de Kalman, un paso

Aplicación de filtro de Kalman a las secuencias de validación de tráfico de los nodos apoq, dehesa, slucia y pcoya, para predicción de un paso ($k = 1$)

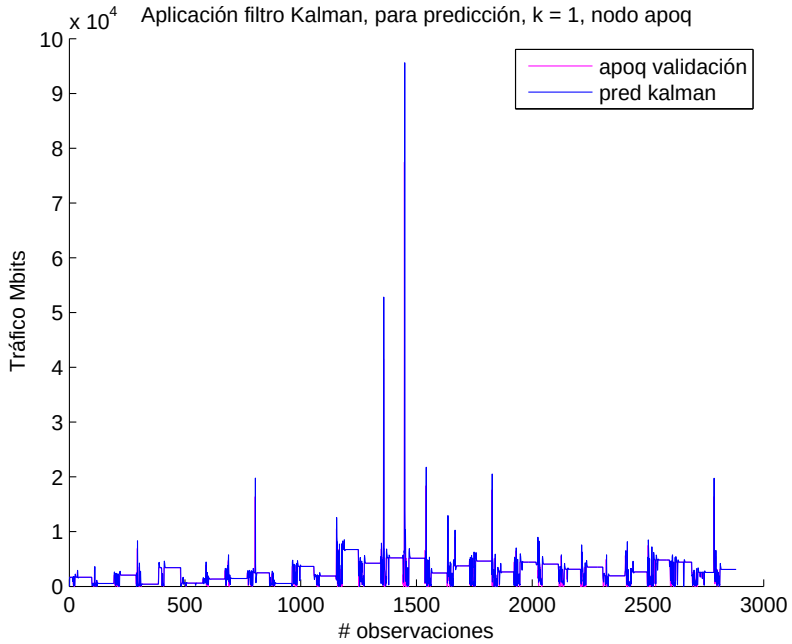


Figura A.7.: Filtro de Kalman para predicción $k = 1$, nodo apoq

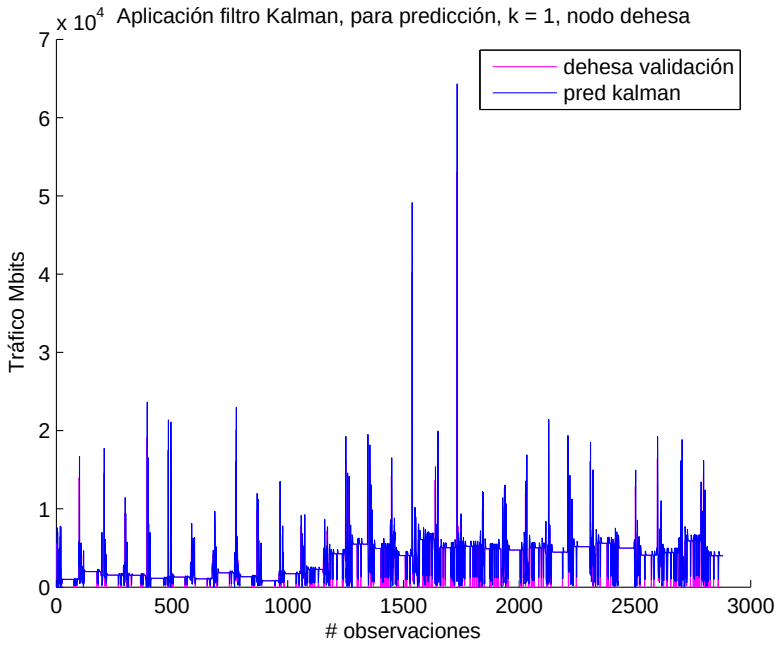


Figura A.8.: Filtro de Kalman para predicción $k = 1$, nodo dehesa

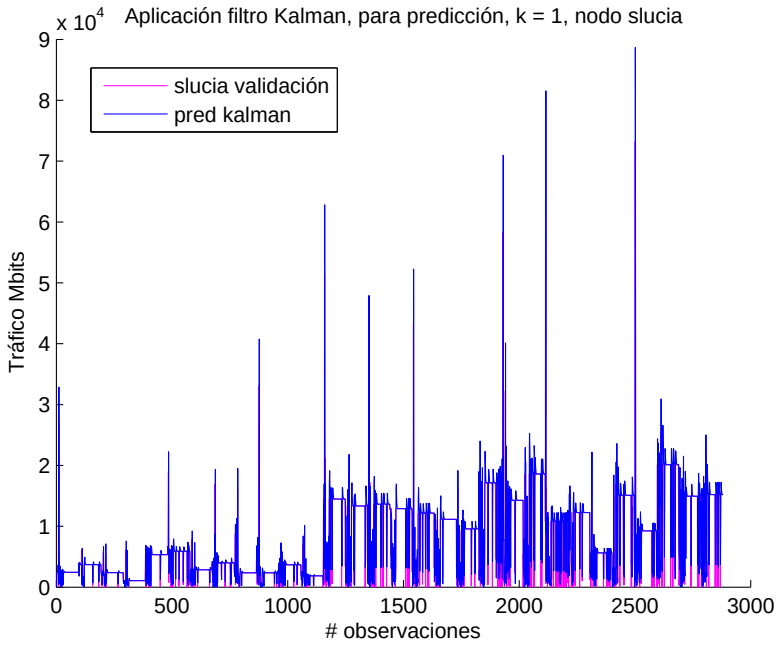


Figura A.9.: Filtro de Kalman para predicción $k = 1$, nodo slucia

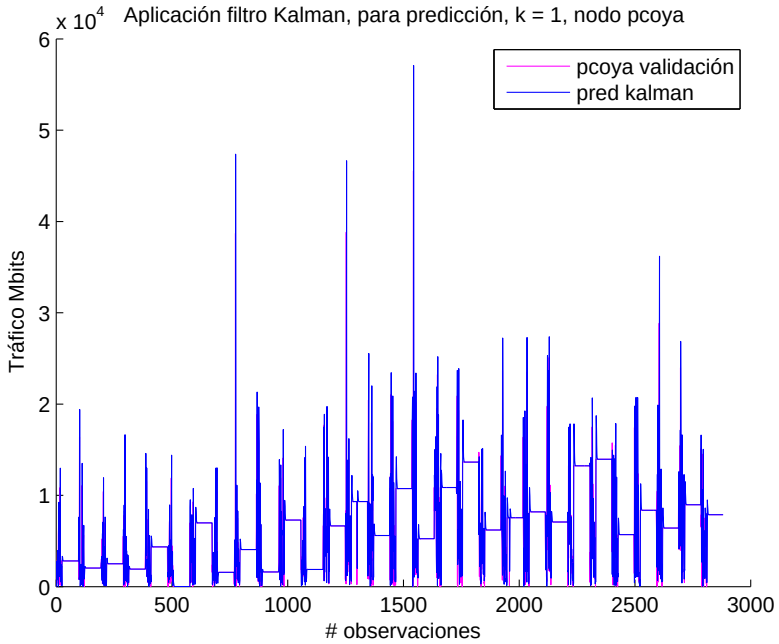


Figura A.10.: Filtro de Kalman para predicción $k = 1$, nodo pcoya

A.5.3. Predicciones filtro de Kalman, 4 pasos

Aplicación de filtro de Kalman a las secuencias de validación de tráfico de los nodos apoq, dehesa, slucia y pcoya, para predicción de 4 pasos ($k = 4$)

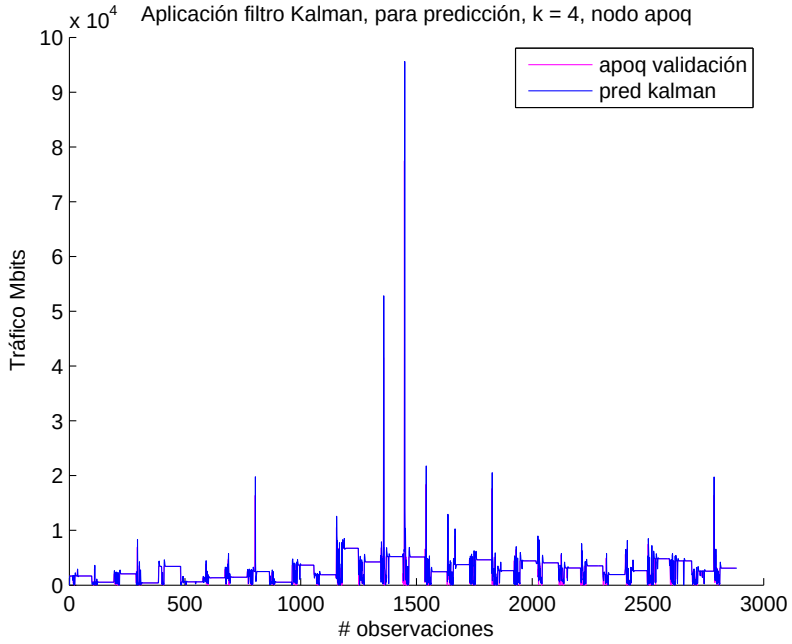


Figura A.11.: Filtro de Kalman para predicción $k = 4$, nodo apoq

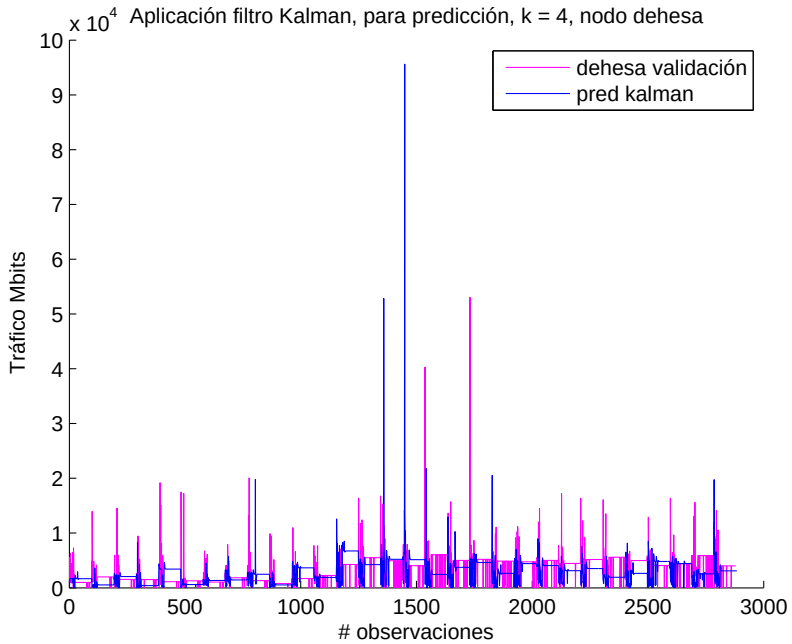


Figura A.12.: Filtro de Kalman para predicción $k = 4$, nodo dehesa

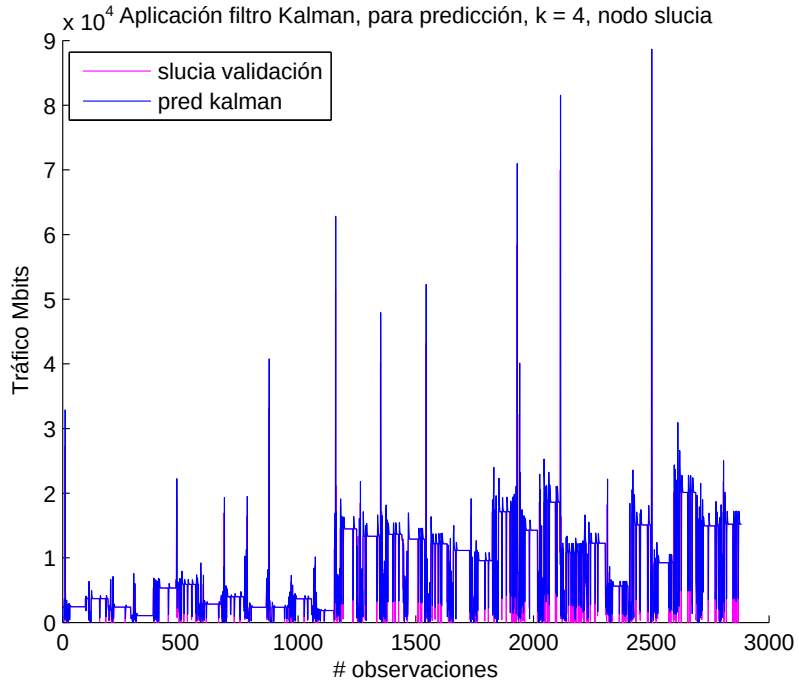


Figura A.13.: Filtro de Kalman para predicción $k = 4$, nodo slucia

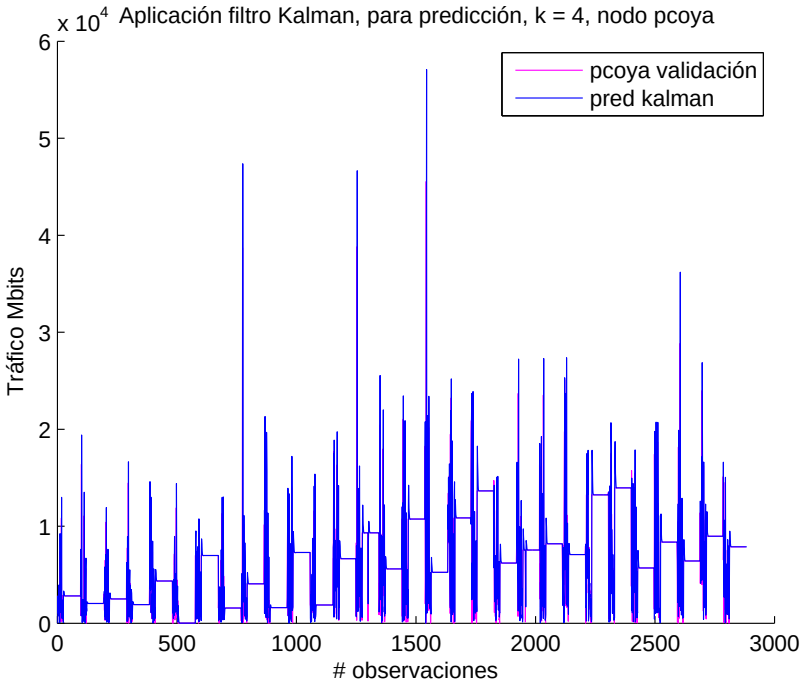


Figura A.14.: Filtro de Kalman para predicción $k = 4$, nodo pcoya

B. Anexo B: Procesamiento y obtención de mediciones de tráfico

Para obtener los reportes de consultas de tráfico por DSLAM, el proceso consta de 4 partes, que a su vez están compuestas de varios pasos o consultas como se muestra en el siguiente esquema general.

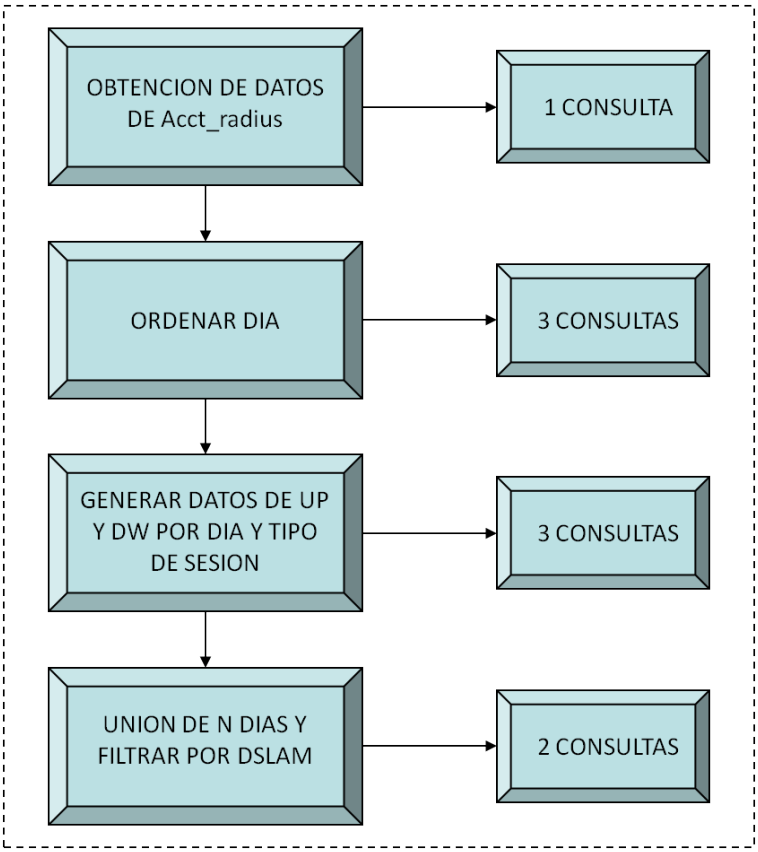


Figura B.1.: Esquema general obtención Reporte de Tráfico de DSLAM

B.1. Obtención de datos de Acct_radius

Los datos de tráfico se encuentran en servidores locales del operador de telecomunicaciones, para acceder a la base de datos Radius, denominada Acct_radius. En esta base se encuentran una serie de tablas que incluye los Radius para cada día. Las tablas por días son:

acct_radius_fecha, acct_radius_fecha_rapel, acct_radius_fecha_teno, acct_radius_fecha_maipo y acct_radius_fecha_puangue.

Nota: algunos días se encuentran en los servidores de bases de datos 'rapel' y 'teno' solamente y hay otros días que se agregan a los servidores 'maipo' y 'puangue'. Los nombres de estos servidores se enumeran como referencia al incluirse en las consultas después listadas [B.2].

Cuando se quiere obtener los datos para un día cualquiera se hace la consulta para el día en cuestión y el día siguiente. Con la consulta 1 se obtienen los datos de los siguientes campos:

Session_End_Dt;Session_End_Tm;Session_Duration_Cnt;Acct_Input_Packets;
Acct_Output_Packets;Acct_Input_Gigawords;Acct_Input_Octets;
Acct_Output_Gigawords;Acct_Output_Octets;End_User_IP_Address_Val;
Session_User_Name;Vpi_Address_Num;Vci_Address_Num;
ISP_Provider_Code_Val;Session_Disconn_Reason_Cd;Session_Identification_Val;
Session_NAS_Port_Val;Session_NAS_Ip_Address_Val;Session_Server_Source_Cd;
Aggregator_Name;Dslam_Name;Dslam_Port_Name

Tabla B.1.: Consulta 1

Siendo los campos utilizados, los que indican: la fecha (*Session_End_Dt*), el termino de la sesión en formato de hora (*Session_End_Tm*), la duración de sesión (*Session_Duration_Cnt*), tráfico de entrada y salida, nas_Port (*Session_NAS_Port_Val*) y nas_IP (*Session_NAS_Ip_Address_Val*).

La consulta 1 entrega en una tabla los datos de dos días mezclados y el campo *Session_End_Tm*, en desorden.

B.1.1. Ordenar día

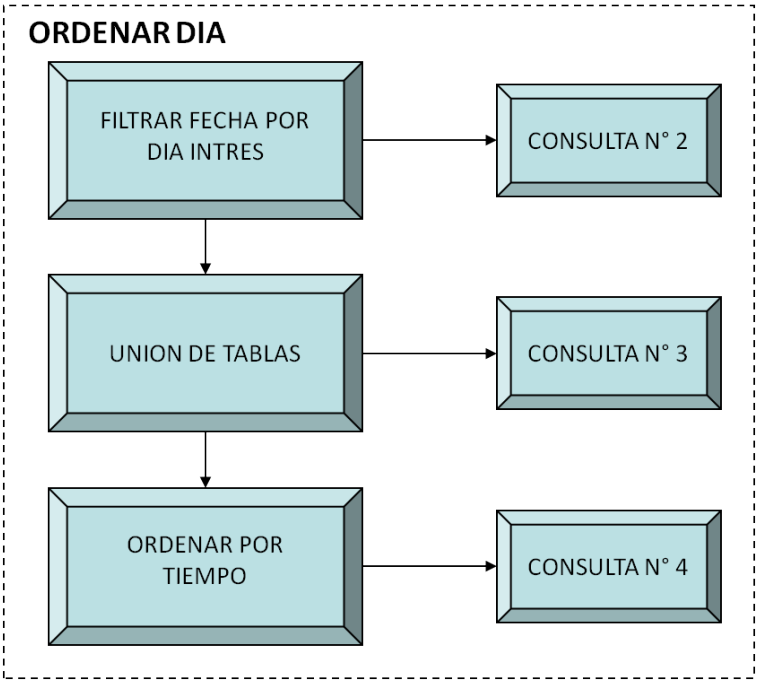


Figura B.2.: Esquema ordenar día

En esta parte se crea un archivo que tiene sólo un día y ordenado por el campo *Session_End_Tm*.

Para obtener un día se considera el día anterior y el día actual de la parte previa, obtención de datos de Acct_radius, para filtrar por la fecha día en consulta (campo *Session_End_Dt*). Esto lo realiza la consulta 2.

Luego se tienen que unir estas dos consultas (consulta 3) para posteriormente ordenar esta unión por el campo *Session_End_Tm* (consulta 4).

B.1.2. Generar datos de up y dw por día y tipo de sesión

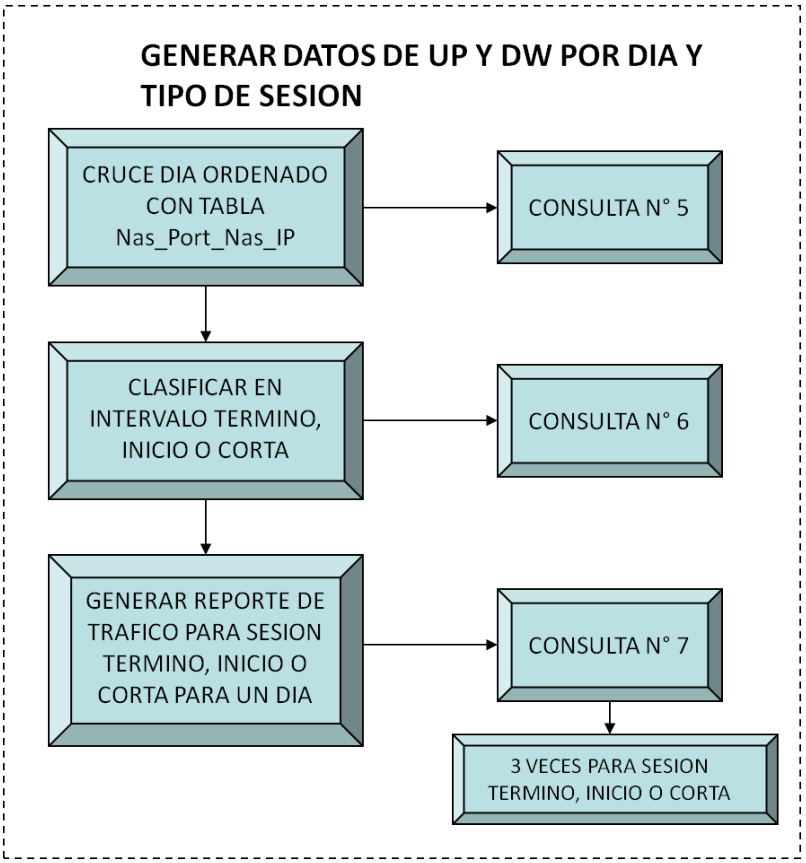


Figura B.3.: Generar datos de up y dw por día y tipo de sesión

Se cruza la tabla *3755_nas_port_nas_ip_trafico* que contiene los NAS_PORT y NAS_IP de los 4 DSLAM en consulta, con la tabla de un día ordenado obtenido de la parte 2. La consulta 5 realiza esto y se crea la tabla *Cruce_radius_dslam*.

El paso siguiente es clasificar en sesión de termino, inicio o corta (consulta 6). Para esto se crea el campo *Stop_seg* que es la conversión del campo *Session_End_Tm* en segundos.

Un sesión es clasificada corta si Session_Duration_Cnt <= 900 y el intervalo donde ocurre se determina como (Int(Stop_seg/900)+1)

Un sesión es clasificada como de termino si Session_Duration_Cnt > 900 y el intervalo donde termina se determina como (Int(Stop_seg/900)+1)

Un sesión es clasificada como de inicio si Session_Duration_Cnt > 900 y además si Stop_seg-Session_Duration_Cnt > 0 y su intervalo de inicio es (Stop_seg-Session_Duration_Cnt)/900)-1.

Luego se genera un reporte de Bit_UP, MBit_UP, Bit_DW y MBit_DW para la sesión de término, inicio o corta para cada día (consulta 7).

B.1.3. Unión de días y filtro por DSLAM

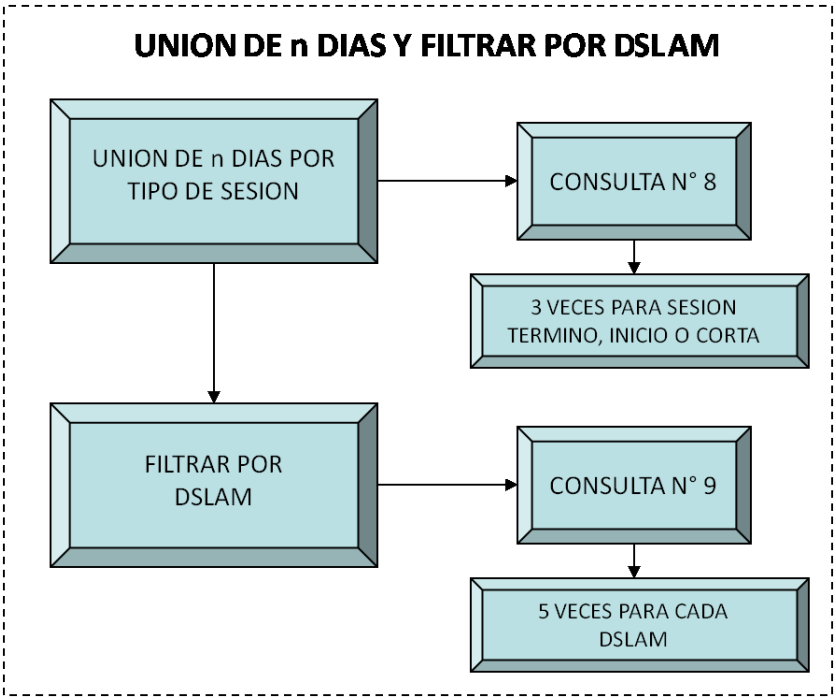


Figura B.4.: Unión de días y filtro por DSLAM

Todo lo anterior se realiza para un día. Una vez que se tienen los N días (N = 117) para cada tipo de sesiones ya procesadas hasta la parte 3 se procede a la unión de los datos (consulta 8).

Finalmente se filtra por DSLAM cada uno de los 3 archivos generados de la parte anterior (consulta 9).

B.2. Consultas

Obtención de datos de Acct_radius

Consulta 1

```
SELECT Replace(Mid(timestamp,1,Instr(timestamp," ")),"-","/")       AS Session_End_Dt, Mid(
    timestamp,reverse(InStr(timestamp," ")))                        AS Session_End_Tm,
sesiontime    As  Session_Duration_Cnt, 1             As Acct_Input_Packets, 1             As
    Acct_Output_Packets, input_giga    As  Acct_Input_Gigawords, input          As
    Acct_Input_Octets, output_giga  As  Acct_Output_Gigawords, output        As
    Acct_Output_Octets, Framedip      As  End_User_IP_Address_Val, login          As
    Session_User_Name,
If(nas_type="virtual",      If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet" Or
     Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,InStr
    (calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,":")+1,
    InStr(calling_id,"-")-1-InStr(calling_id,":")),          Mid(Mid(calling_id,InStr(
    calling_id,":")+1),1,InStr(Mid(calling_id,InStr(calling_id,":")+1),".")-1)          ) ,
       If(nas_type="xDSL",          Substring_Index(Substring_Index(calling_id,"#",-2),"#",1)
    ,          ''          )       )                                                    AS Vpi_Address_Num
    ,
If(nas_type="virtual",       If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet"
    Or Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,
    InStr(calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,"-")
    +1,InStr(calling_id,"#")-1-InStr(calling_id,"-")),          SUBSTRING_INDEX(Mid(Mid(
    calling_id,InStr(calling_id,":")+1),InStr(Mid(calling_id,InStr(calling_id,":")+1),".")
    +1),'#',1)          ),       If(nas_type="xDSL",          Substring_Index(calling_id
    ,"#",-1),          ''          )    )                                                AS
    Vci_Address_Num,
If ( instr(login ,".") > 0 ,       Substring_Index( login,".",-1) ,       Substring_Index(
    login,"@",-1)      )                                                 As ISP_Provider_Code_Val,
terminate       As Session_Disconn_Reason_Cd, session_id      As Session_Identification_Val,
    Nasport         As Session_NAS_Port_Val, NasIp          As Session_NAS_Ip_Address_Val,
    radius          As Session_Server_Source_Cd,
If(nas_type<>"virtual",  mID(Mid(calling_id, 2),1,Instr(Mid(calling_id, 2),"#")-1) ,
    session_id)          As Aggregator_Name,
nas_type                 As Dslam_Name, calling_id                 As Dslam_Port_Name
FROM acct_radius_20100820_teno where status ="Stop" and  timestamp >= "2010/08/20 00:00:00"
union all
SELECT Replace(Mid(timestamp,1,Instr(timestamp," ")),"-","/")       AS Session_End_Dt, Mid(
    timestamp,reverse(InStr(timestamp," ")))                        AS Session_End_Tm,
sesiontime    As  Session_Duration_Cnt, 1             As Acct_Input_Packets, 1             As
    Acct_Output_Packets, input_giga    As  Acct_Input_Gigawords, input          As
    Acct_Input_Octets, output_giga  As  Acct_Output_Gigawords, output        As
    Acct_Output_Octets, Framedip      As  End_User_IP_Address_Val, login          As
    Session_User_Name,
If(nas_type="virtual",      If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet" Or
     Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,InStr
    (calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,":")+1,
    InStr(calling_id,"-")-1-InStr(calling_id,":")),          Mid(Mid(calling_id,InStr(
    calling_id,":")+1),1,InStr(Mid(calling_id,InStr(calling_id,":")+1),".")-1)          ) ,
       If(nas_type="xDSL",          Substring_Index(Substring_Index(calling_id,"#",-2),"#",1)
    ,          ''          )       )                                                    AS Vpi_Address_Num
    ,
If(nas_type="virtual",       If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet"
    Or Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,
    InStr(calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,"-")
    +1,InStr(calling_id,"#")-1-InStr(calling_id,"-")),          SUBSTRING_INDEX(Mid(Mid(
    calling_id,InStr(calling_id,":")+1),InStr(Mid(calling_id,InStr(calling_id,":")+1),".")
    +1),'#',1)          ),       If(nas_type="xDSL",          Substring_Index(calling_id
    ,"#",-1),          ''          )    )                                                AS
    Vci_Address_Num,
If ( instr(login ,".") > 0 ,       Substring_Index( login,".",-1) ,       Substring_Index(
    login,"@",-1)      )                                                 As ISP_Provider_Code_Val,
terminate       As Session_Disconn_Reason_Cd, session_id      As Session_Identification_Val,
    Nasport         As Session_NAS_Port_Val, NasIp          As Session_NAS_Ip_Address_Val,
    radius          As Session_Server_Source_Cd,
```

```
If(nas_type<>"virtual", mID(Mid(calling_id, 2),1,Instr(Mid(calling_id, 2),"#")-1),
    session_id)          As Aggregator_Name,
nas_type                 As Dslam_Name, calling_id              As Dslam_Port_Name
FROM acct_radius_20100820_rapel where status ="Stop" and  timestamp >= "2010/08/20
    00:00:00"
union all
SELECT Replace(Mid(timestamp,1,Instr(timestamp," ")),"-","/")      AS Session_End_Dt, Mid(
    timestamp,reverse(InStr(timestamp," ")))                  AS Session_End_Tm,
sesiontime   As  Session_Duration_Cnt, 1          As Acct_Input_Packets, 1          As
    Acct_Output_Packets, input_giga   As  Acct_Input_Gigawords, input       As
    Acct_Input_Octets, output_giga  As  Acct_Output_Gigawords, output       As
    Acct_Output_Octets, Framedip     As  End_User_IP_Address_Val, login       As
    Session_User_Name,
If(nas_type="virtual",     If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet" Or
     Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,InStr
    (calling_id," ")-1)="FastEthernet",         Mid(calling_id,InStr(calling_id,":")+1,
    InStr(calling_id,"-")-1-InStr(calling_id,":")),         Mid(Mid(calling_id,InStr(
    calling_id,":")+1),1,InStr(Mid(calling_id,InStr(calling_id,":")+1),".")-1)       ),
       If(nas_type="xDSL",       Substring_Index(Substring_Index(calling_id,"#",-2),"#",1)
    ,       ''       )     )                                    AS Vpi_Address_Num,
If(nas_type="virtual",       If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet"
    Or Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,
    InStr(calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,"-")
    +1,InStr(calling_id,"#")-1-InStr(calling_id,"-")),          SUBSTRING_INDEX(Mid(Mid(
    calling_id,InStr(calling_id,":")+1),InStr(Mid(calling_id,InStr(calling_id,":")+1),".")
    +1),'#',1)       ),       If(nas_type="xDSL",          Substring_Index(calling_id
    ,"#",-1),       ''       )   )                                  AS
    Vci_Address_Num,
If ( instr(login ,".") > 0 ,      Substring_Index( login,".",-1) ,      Substring_Index(
    login,"@",-1)     )                                    As ISP_Provider_Code_Val,
terminate      As Session_Disconn_Reason_Cd, session_id      As Session_Identification_Val,
    Nasport        As Session_NAS_Port_Val, NasIp          As Session_NAS_Ip_Address_Val,
    radius         As Session_Server_Source_Cd,
If(nas_type<>"virtual", mID(Mid(calling_id, 2),1,Instr(Mid(calling_id, 2),"#")-1),
    session_id)          As Aggregator_Name,
nas_type                 As Dslam_Name, calling_id              As Dslam_Port_Name
FROM acct_radius_20100821_teno where status ="Stop" and  timestamp < "2010/08/21 00:00:00"
union all SELECT Replace(Mid(timestamp,1,Instr(timestamp," ")),"-","/")      AS
    Session_End_Dt, Mid(timestamp,reverse(InStr(timestamp," ")))              AS
    Session_End_Tm,
sesiontime   As  Session_Duration_Cnt, 1          As Acct_Input_Packets, 1          As
    Acct_Output_Packets, input_giga   As  Acct_Input_Gigawords, input       As
    Acct_Input_Octets, output_giga  As  Acct_Output_Gigawords, output       As
    Acct_Output_Octets, Framedip     As  End_User_IP_Address_Val, login       As
    Session_User_Name,
If(nas_type="virtual",     If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet" Or
     Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,InStr
    (calling_id," ")-1)="FastEthernet",         Mid(calling_id,InStr(calling_id,":")+1,
    InStr(calling_id,"-")-1-InStr(calling_id,":")),         Mid(Mid(calling_id,InStr(
    calling_id,":")+1),1,InStr(Mid(calling_id,InStr(calling_id,":")+1),".")-1)       ),
       If(nas_type="xDSL",       Substring_Index(Substring_Index(calling_id,"#",-2),"#",1)
    ,       ''       )     )                                    AS Vpi_Address_Num,
If(nas_type="virtual",       If(Left(calling_id,InStr(calling_id," ")-1)="GigabitEthernet"
    Or Left(calling_id,InStr(calling_id," ")-1)="TenGigabitEthernet" Or Left(calling_id,
    InStr(calling_id," ")-1)="FastEthernet",          Mid(calling_id,InStr(calling_id,"-")
    +1,InStr(calling_id,"#")-1-InStr(calling_id,"-")),          SUBSTRING_INDEX(Mid(Mid(
    calling_id,InStr(calling_id,":")+1),InStr(Mid(calling_id,InStr(calling_id,":")+1),".")
    +1),'#',1)       ),       If(nas_type="xDSL",          Substring_Index(calling_id
    ,"#",-1),       ''       )   )                                  AS
    Vci_Address_Num,
If ( instr(login ,".") > 0 ,      Substring_Index( login,".",-1) ,      Substring_Index(
    login,"@",-1)     )                                    As ISP_Provider_Code_Val,
terminate      As Session_Disconn_Reason_Cd, session_id      As Session_Identification_Val,
    Nasport        As Session_NAS_Port_Val, NasIp          As Session_NAS_Ip_Address_Val,
    radius         As Session_Server_Source_Cd,
If(nas_type<>"virtual", mID(Mid(calling_id, 2),1,Instr(Mid(calling_id, 2),"#")-1),
    session_id)          As Aggregator_Name,
nas_type                 As Dslam_Name, calling_id              As Dslam_Port_Name
FROM acct_radius_20100821_rapel where status ="Stop" and  timestamp < "2010/08/21 00:00:00"
    ;
```

Ordenar día

Consulta 2: Filtrar por fecha de interés

```
SELECT_____RADIUS_____20110102.Session_End_Dt, RADIUS_____20110102.Session_End_Tm,
    RADIUS_____20110102.Session_Duration_Cnt, RADIUS_____20110102.Acct_Input_Packets,
    RADIUS_____20110102.Acct_Output_Packets, RADIUS_____20110102.Acct_Input_Gigawords,
    RADIUS_____20110102.Acct_Input_Octets, RADIUS_____20110102.Acct_Output_Gigawords,
    RADIUS_____20110102.Acct_Output_Octets, RADIUS_____20110102.End_User_IP_Address_Val,
    RADIUS_____20110102.Session_User_Name, RADIUS_____20110102.Vpi_Address_Num,
    RADIUS_____20110102.Vci_Address_Num, RADIUS_____20110102.ISP_Provider_Code_Val,
    RADIUS_____20110102.Session_Disconn_Reason_Cd, RADIUS_____20110102.
    Session_Identification_Val, RADIUS_____20110102.Session_NAS_Port_Val, RADIUS_____20110102
    .Session_NAS_Ip_Address_Val, RADIUS_____20110102.Session_Server_Source_Cd,
    RADIUS_____20110102.Aggregator_Name, RADIUS_____20110102.Dslam_Name, RADIUS_____20110102.
    Dslam_Port_Name FROM RADIUS_____20110102 WHERE (((RADIUS_____20110102.Session_End_Dt)
    ="2011/01/03"));
```

Consulta 3: Unión de tablas

```
SELECT [Radius_20110301_parte1].* FROM Radius_20110301_parte1 UNION ALL SELECT [
    Radius_20110301_parte2].* FROM Radius_20110301_parte2;
```

Consulta 4: Ordenar por tiempo

```
SELECT Radius_20110303_d.*
FROM Radius_20110303_d ORDER BY Radius_20110303_d.Session_End_Tm;
```

Generar datos de UP y DW por día y tipo de sesión

Consulta 5: Cruce Radius_dia_ordenado con tabla Nas_Port_IP

```
SELECT Radius_06022011.Session_End_Dt, Radius_06022011.Session_End_Tm, Radius_06022011.
    Session_Duration_Cnt, Radius_06022011.Acct_Input_Packets, Radius_06022011.
    Acct_Output_Packets, Radius_06022011.Acct_Input_Gigawords, Radius_06022011.
    Acct_Input_Octets, Radius_06022011.Acct_Output_Gigawords, Radius_06022011.
    Acct_Output_Octets, Radius_06022011.End_User_IP_Address_Val, Radius_06022011.
    Session_User_Name, Radius_06022011.Vpi_Address_Num, Radius_06022011.Vci_Address_Num,
    Radius_06022011.ISP_Provider_Code_Val, Radius_06022011.Session_Disconn_Reason_Cd,
    Radius_06022011.Session_Identification_Val, Radius_06022011.Session_NAS_Port_Val,
    Radius_06022011.Session_NAS_Ip_Address_Val, Radius_06022011.Session_Server_Source_Cd,
    Radius_06022011.Aggregator_Name, Radius_06022011.Dslam_Name, Radius_06022011.
    Dslam_Port_Name, [3755_nas_port_nas_ip_trafico].DSLAM, [3755_nas_port_nas_ip_trafico].
    IP_DSLAM, [3755_nas_port_nas_ip_trafico].Tipo_de_Puerto, [3755_nas_port_nas_ip_trafico
    ].PLANTA, [3755_nas_port_nas_ip_trafico].AGREGADOR, [3755_nas_port_nas_ip_trafico].
    NAS_SESSION_ID, [3755_nas_port_nas_ip_trafico].ID_PUERTO FROM 3755
    _nas_port_nas_ip_trafico INNER JOIN Radius_06022011 ON ([3755_nas_port_nas_ip_trafico].
    NAS_IP = Radius_06022011.Session_NAS_Ip_Address_Val) AND ([3755_nas_port_nas_ip_trafico
    ].NAS_PORT = Radius_06022011.Session_NAS_Port_Val);
```

Consulta 6: Clasificar en intervalo Termino, Inicio o Corta

```
SELECT Cruce_radius_dslam.Session_End_Dt, Cruce_radius_dslam.Session_End_Tm,
    Cruce_radius_dslam.Session_Duration_Cnt, Cruce_radius_dslam.Acct_Input_Packets,
    Cruce_radius_dslam.Acct_Output_Packets, Cruce_radius_dslam.Acct_Input_Gigawords,
    Cruce_radius_dslam.Acct_Input_Octets, Cruce_radius_dslam.Acct_Output_Gigawords,
    Cruce_radius_dslam.Acct_Output_Octets, Cruce_radius_dslam.End_User_IP_Address_Val,
    Cruce_radius_dslam.Session_User_Name, Cruce_radius_dslam.Vpi_Address_Num,
    Cruce_radius_dslam.Vci_Address_Num, Cruce_radius_dslam.ISP_Provider_Code_Val,
    Cruce_radius_dslam.Session_Disconn_Reason_Cd, Cruce_radius_dslam.
    Session_Identification_Val, Cruce_radius_dslam.Session_NAS_Port_Val, Cruce_radius_dslam
    .Session_NAS_Ip_Address_Val, Cruce_radius_dslam.Session_Server_Source_Cd,
    Cruce_radius_dslam.Aggregator_Name, Cruce_radius_dslam.Dslam_Name, Cruce_radius_dslam.
    Dslam_Port_Name, Cruce_radius_dslam.DSLAM, IIF(Acct_Input_Gigawords>0,(
    Acct_Input_Gigawords)*4294967295+Acct_Input_Octets,Acct_Input_Octets) AS
    Acct_Input_Octet_Gigawords, IIF(Acct_Output_Gigawords>0,(Acct_Output_Gigawords)
    *4294967295+Acct_Output_Octets,Acct_Output_Octets) AS Acct_Output_Octet_Gigawords,
    Cruce_radius_dslam.Session_Duration_Cnt, 3600*CInt(Left([Session_End_Tm],2))+60*CInt(
    Left(Mid([Session_End_Tm],4),2))+CInt(Right([Session_End_Tm],2)) AS Stop_seg, IIF(
    Session_Duration_Cnt>900,Int(Stop_seg/900)+1,0) AS Intervalo_Termino, IIF(
    Session_Duration_Cnt>900,IIF(Stop_seg-Session_Duration_Cnt>0,Int((Stop_seg-
    Session_Duration_Cnt)/900)-1,0),0) AS Intervalo_Inicio, IIF(Session_Duration_Cnt<=900,
    Int(Stop_seg/900)+1,0) AS Intervalo_Corta FROM Cruce_radius_dslam;
```

Consulta 7: Genera reporte de Bit_UP, MBit_UP, Bit_DW y MBit_DW para sesión de Término, Inicio o Corta para cada día

```
SELECT
IIf([Session_End_Dt]="2011/02/07","2011/02/07","2011/02/07") AS Fecha,
Rango_DSlam.DSLAM, Rango_DSlam.Rango,
IIf((Sum(8*[Acct_Input_Octet_Gigawords]))>0,Sum(8*[Acct_Input_Octet_Gigawords]),0) AS
    Bit_UP,
(IIf((Sum(8*[Acct_Input_Octet_Gigawords]))>0,Sum(8*[Acct_Input_Octet_Gigawords]),0))
    /1048576 AS MBit_UP,
IIf((Sum(8*[Acct_Output_Octet_Gigawords]))>0,Sum(8*[Acct_Output_Octet_Gigawords]),0) AS
    Bit_DW,
(IIf((Sum(8*[Acct_Output_Octet_Gigawords]))>0,Sum(8*[Acct_Output_Octet_Gigawords]),0))
    /1048576 AS MBit_DW
FROM Rango_DSlam LEFT JOIN Cruce_radius_dslam_3_rangos ON (Rango_DSlam.Rango =
    Cruce_radius_dslam_3_rangos.Intervalo_Termino) AND (Rango_DSlam.DSLAM =
    Cruce_radius_dslam_3_rangos.DSLAM) GROUP BY Cruce_radius_dslam_3_rangos.Session_End_Dt,
     Rango_DSlam.DSLAM, Rango_DSlam.Rango ORDER BY Rango_DSlam.DSLAM, Rango_DSlam.Rango;
```

Unión de días y filtro por DSLAM

Consulta 8: Unir los 60 días, para corta, termine e inicio

```
select * from Reporte_Corta_02012011
union
select * from Reporte_Corta_03012011
union
select * from Reporte_Corta_04012011
union
.
.
.
select * from Reporte_Corta_26022011
union
select * from Reporte_Corta_27022011
union
select * from Reporte_Corta_28022011
```

Consulta 9: Filtrar por DSLAM

```
SELECT Reporte_Cortas_todas.Fecha, Reporte_Cortas_todas.DSLAM, Reporte_Cortas_todas.Rango,
    Reporte_Cortas_todas.Bit_UP, Reporte_Cortas_todas.MBit_UP, Reporte_Cortas_todas.Bit_DW,
     Reporte_Cortas_todas.MBit_DW FROM Reporte_Cortas_todas WHERE (((Reporte_Cortas_todas.
    DSLAM)="APOQUINDO-18"));
```

C. Anexo C: Complementos

C.1. Ejemplo minimización de una función con Karush–Kuhn–Tucker

Minimizar la función

$$z = (x_1 - 2)^2 + (x_2 - 3)^2 + (x_3 - 3)^2$$

sujeto a las siguientes restricciones de desigualdad

$$\begin{aligned} x_1 + x_2 - x_3 &\leq 1 \\ 2x_1 - x_2 - 2x_3 &\leq 2 \end{aligned}$$

Desarrollar

- Las condiciones KT del problema
- Las soluciones que cumplen las condiciones KT del problema
- Los puntos mínimos

Condiciones KT del problema

Dados

$$f(x_1, x_2, x_3) = (x_1 - 2)^2 + (x_2 - 3)^2 + (x_3 - 3)^2$$

con las restricciones

$$\begin{aligned} g_1(x_1, x_2, x_3) &= x_1 + x_2 - x_3 - 1 \leq 0 \\ g_2(x_1, x_2, x_3) &= 2x_1 - x_2 - 2x_3 - 2 \leq 0 \end{aligned}$$

La forma de la función de Lagrange es

$$L(x, \lambda) = (x_1 - 2)^2 + (x_2 - 3)^2 + (x_3 - 3) + \lambda_1 [x_1 + x_2 - x_3 - 1] + \lambda_2 [2x_1 - x_2 - 2x_3 - 2]$$

De esta manera, las condiciones KT son

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= 2(x_1-2)+\lambda_1+2\lambda_2=0 \\
\frac{\partial L}{\partial x_2} &= 2(x_2-3)+\lambda_1-\lambda_2=0 \\
\frac{\partial L}{\partial x_3} &= 2(x_3-3)-\lambda_1-2\lambda_2=0 \\
\frac{\partial L}{\partial \lambda_1} &= g_1(x)=x_1+x_2-x_3-1\leq 0 \\
\frac{\partial L}{\partial \lambda_2} &= g_2(x)=2x_1-x_2-2x_3-2\leq 0 \\
\lambda_1\left[x_1+x_2-x_3-1\right] &= 0 \\
\lambda_2\left[2x_1-x_2-2x_3-2\right] &= 0 \\
\lambda_1\geq 0 \qquad & \lambda_2\geq 0
\end{aligned}$$

Posibles soluciones que cumplen las condiciones KT

Considerar los 4 casos posibles

$\lambda_1=0$	$\lambda_2=0$
$\lambda_1\neq 0$	$\lambda_2\neq 0$
$\lambda_1=0$	$\lambda_2\neq 0$
$\lambda_1\neq 0$	$\lambda_2=0$

Tabla C.1.: 4 alternativas de minimización de una función de ejemplo

Primer caso

Cuando $\lambda_1=0$ y $\lambda_2=0$, las ecuaciones de la condición de gradiente son

$$\begin{aligned}
2(x_1-2)=0 \quad &\text{dados} \quad x_1=2 \\
2(x_2-3)=0 \quad &\text{dados} \quad x_2=3 \\
2(x_3-3)=0 \quad &\text{dados} \quad x_3=3
\end{aligned}$$

Usando los valores x_1, x_2 y x_3, en la primera restricción, se tiene

$$x_1+x_2-x_3-1=2+3-3-1=1\nleq 0$$

que no se cumple

Segundo caso

Cuando $\lambda_1\neq 0$ y $\lambda_2\neq 0$, las ecuaciones de la condición de gradiente son

$$2(x_1-2)+\lambda_1+2\lambda_2=0 \text{ dados } x_1=\frac{1}{2}[4-\lambda_1-2\lambda_2]$$

$$2(x_2-3)+\lambda_1-\lambda_2=0 \text{ dados } x_2=\frac{1}{2}[6-\lambda_1+\lambda_2]$$

$$2(x_3 - 3) - \lambda_1 - 2\lambda_2 = 0 \text{ dados } x_3 = \frac{1}{2}[6 + \lambda_1 + 2\lambda_2]$$

Donde $\lambda_1 \neq 0$ y $\lambda_2 \neq 0$, de las condiciones de ortogonalidad, se tiene

$$\begin{aligned} x_1 + x_2 - x_3 - 1 &= 0 \\ 2x_1 - x_2 - 2x_3 - 2 &= 0 \end{aligned}$$

Usando los valores de x_1, x_2 y x_3, en el sistema de ecuaciones

$$\begin{aligned} 3\lambda_1 + 9\lambda_2 &= -14 \\ 3\lambda_1 + 3\lambda_2 &= 2 \end{aligned}$$

La solución del sistema es $\lambda_1 = \dfrac{10}{3}$ y $\lambda_2 = -\dfrac{8}{3}$, pero de la condición de no negatividad, $\lambda_2 \geq 0$, descarta la solución.

Tercer caso

Cuando $\lambda_1 = 0$ y $\lambda_2 \neq 0$, las ecuaciones de la condición de gradiente son

$$\begin{aligned} 2(x_1 - 2) + 2\lambda_2 = 0 \quad &\text{dados} \quad x_1 = 2 - \lambda_2 \\ 2(x_2 - 3) - \lambda_2 = 0 \quad &\text{dados} \quad x_2 = \frac{1}{2}[6 + \lambda_2] \\ 2(x_3 - 3) - 2\lambda_2 = 0 \quad &\text{dados} \quad x_3 = 3 + \lambda_2 \end{aligned}$$

Donde $\lambda_2 \neq 0$, de las condiciones de ortogonalidad, se tiene

$$2x_1 - x_2 - 2x_3 - 2 = 0$$

Usando los valores de x_1, x_2 y x_3, tenemos

$$2(2 - \lambda_2) - \frac{1}{2}(6 + \lambda_2) - 2(3 + \lambda_2) - 2 = 0$$

Con $\lambda_2 = -\dfrac{8}{7}$, por lo que la solución también se descarta.

Cuarto caso

Cuando $\lambda_1 \neq 0$ y $\lambda_2 \neq 0$, las ecuaciones de la condición de gradiente son

$$\begin{aligned} 2(x_1 - 2) + \lambda_1 = 0 \quad &\text{dados} \quad x_1 = \frac{1}{2}[4 - \lambda_1] \\ 2(x_2 - 3) + \lambda_1 = 0 \quad &\text{dados} \quad x_2 = \frac{1}{2}[6 - \lambda_1] \\ 2(x_3 - 3) - \lambda_1 = 0 \quad &\text{dados} \quad x_3 = \frac{1}{2}[6 + \lambda_1] \end{aligned}$$

Donde $\lambda_1 \neq 0$, de las condiciones de ortogonalidad, se tiene

$$x_1 + x_2 - x_3 - 1 = 0$$

Usando los valores de x_1, x_2 y x_3, tenemos

$$\frac{1}{2}(4-\lambda_1)+\frac{1}{2}(6-\lambda_1)-\frac{1}{2}(6+\lambda_1)-1=0, \quad \lambda_1=\frac{2}{3}$$

Se tiene entonces que

$$\overline{x}=\left[\frac{5}{3},\frac{8}{3},\frac{10}{3}\right]^{\mathrm{T}} \qquad \text{y} \qquad \overline{\lambda}=\left[\frac{2}{3},0\right]^{\mathrm{T}}$$

es un punto KT.

Los puntos mínimos

Se revisa si las funciones f, g_1 y g_2 son convexas o no. Primero se analiza si la función objetivo es convexa, de la siguiente forma

$$f(x_1,x_2,x_3)=(x_1-2)^2+(x_2-3)^2+(x_3-3)^2$$

$$\frac{\partial f}{\partial x_1}=2(x_2-3); \quad \frac{\partial f^2}{\partial x_2^2}=2$$
$$\frac{\partial f}{\partial x_2}=2(x_2-2); \quad \frac{\partial f^2}{\partial x_1^2}=2$$
$$\frac{\partial f}{\partial x_3}=2(x_3-2); \quad \frac{\partial f^2}{\partial x_3^2}=2$$

$$\frac{\partial f^2}{\partial x_1 \partial x_2}=\frac{\partial f^2}{\partial x_1 \partial x_3}=\frac{\partial f^2}{\partial x_2 \partial x_3}=0$$

De esta forma la matriz Hessian de la función objetivo es

$$H=\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

Al borrar la filas y columnas 1 y 2 de la matriz Hessian, se obtiene el principal primer-orden menor $2>0$. Borrando las filas y columnas 1 y 3, se obtiene el principal primer-orden menor $2>0$. Al borrar la filas y columnas 2 y 3 de la matriz Hessian, se obtiene el principal primer-orden menor $2>0$.

Borrando la fila 1 y columna 1 de la matriz Hessian, se obtiene el principal segundo-orden menor

$$|H_2|=\left|\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right|=4-0=4>0$$

Borrando la fila 2 y columna 2 de la matriz Hessian, se obtiene el principal segundo-orden menor

$$|H_2|=\left|\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\right|=4-0=4>0$$

Borrando la fila 3 y columna 3 de la matriz Hessian, se obtiene el principal segundo-orden menor

$$|H_2| = \left| \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \right| = 4 - 0 = 4 > 0$$

El principal tercer-orden menor es el determinante de la matriz Hessian, es decir

$$|H_3| = 8 > 0$$

Porque $\forall\ (x_1, x_2, x_3)$ todos los principales menores de la matriz Hessian son no negativos, se tiene que $f(x_1, x_2, x_3)$ es una función convexa.

También, dados que las funciones g_1 y g_2 son lineales y convexas, todas las funciones son convexas, por ello el punto $\overline{x} = \left[\frac{5}{3}, \frac{8}{3}, \frac{10}{3}\right]^{\mathrm{T}}$ es un mínimo global.

Para este ejemplo se tiene que la restricción activa es $\overline{g}_1(x) = 0$ con $\lambda_1 \neq 0$, y la restricción inactiva es $\overline{g}_2(x) < 0$ con $\lambda_2 \neq 0$,

C.2. Test de Kolmogorov-Smirnov

Teorema 1

Si $F(x)$ es continua, entonces la distribución de

$$\sup_{x \in \mathbb{R}} |F_n(x) - F(x)|$$

no depende de F.

Demostración

Definiendo la inversa de F por

$$F^{-1}(y) = \text{mín}\{x : F(x) \geq y\}$$

y realizando cambio de variables $y = F(x)$ o $x = F^{-1}(y)$, se tiene

$$\mathbb{P}\left(\sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \leq k\right) = \mathbb{P}\left(\sup_{0 \leq y \leq 1} \left|F_n(F^{-1}(y)) - y\right| \leq k\right)$$

La definición de distribución empírica c.d.f. F_n

$$F_n(F^{-1}(y)) = \frac{1}{n}\sum_{i}^{n} I(X_i \leq F^{-1}(y)) = \frac{1}{n}\sum_{i}^{n} I(F(X_i) \leq y)$$

se aplica en

$$\mathbb{P}\left(\sup_{0 \leq y \leq 1} \left|F_n(F^{-1}(y)) - y\right| \leq k\right) = \mathbb{P}\left(\sup_{0 \leq y \leq 1} \left|\frac{1}{n}\sum_{i}^{n} I(F(X_i) \leq y) - y\right| \leq k\right)$$

La distribución $F(X_i)$ es uniforme en el intervalo [0,1] debido a que c.d.f. de $F(X_1)$ es

$$\mathbb{P}(F(X_1 \leq k) = \mathbb{P}(X_1 \leq F^{-1}(k)) = F(F^{-1}(k)) = k$$

Las variables aleatorias

$$U_i = F(X_i) \quad \text{para } i \leq n$$

son independientes, con distribución uniforme en [0,1], por lo que

$$\mathbb{P}\left(\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| \leq k\right) = \mathbb{P}\left(\sup_{0\leq y\leq 1} \left|\frac{1}{n}\sum_i^n I(U_i \leq y) - y\right| \leq k\right)$$

es independiente de F.

Si x, cumple con el teorema del límite central, entonces

$$\sqrt{n}(F_n(x) - F(x)) \to {}^d N\left(0, F(x)(1 - F(x))\right.$$

porque $F(x)(1 - F(x))$ es la varianza de $I(X_1 \leq x)$, así

$$\sqrt{n}\sup_{x\in\mathbb{R}} |F_n(x) - F(x)|$$

converge en distribución también.

Teorema 2

$$\mathbb{P}\left(\sqrt{n}\sup_{x\in\mathbb{R}} |F_n(x) - F(x)| \leq k\right) \to H(k) = 1 - 2\sum_i^\infty (-1)^{i-1} e^{-2i^2k}$$

donde $H(k)$ es la distribución, c.d.f., de Kolmogorov-Smirnov

Esquema de test

Considerando la siguiente estadística

$$D_n = \sqrt{n}\sup_{x\in\mathbb{R}} |F_n(x) - F_0(x)|$$

y el test de hipótesis

$$H_0 : F = F_0 \quad \text{vs.} \quad H_1 : F \neq F_0$$

siendo F_0 la distribución, c.d.f., de $\mathbb{P}_0$.

Si la hipótesis es cierta, entonces, por Teorema 1, la distribución D_n sólo depende de n. Además, y n es grande, entonces la distribución D_n se aproxima a una distribución de Kolmogorov-Smirnov, por Teorema 2.

Si la hipótesis es falsa, $F \neq F_0$, como F es la distribución, c.d.f., de la serie, por ley de los grandes números, la distribución empírica, c.d.f. F_n, converge a F, y no a F_0, es decir, para n grande, se tiene

$$\sup_x |F_n(x) - F_0(x)| > \delta$$

para δ pequeño, multiplicando la expresión por $\sqrt{n}$

$$D_n = \sqrt{n} \sup_{x \in \mathbb{R}} |F_n(x) - F_0(x)| > \sqrt{n}\delta$$

Si H_0 falla, entonces $D_n > \sqrt{n}\delta \to +\infty$ cuando $n \to \infty$, así, el test H_0 puede considerarse como una referencia de decisión

$$\delta = \begin{cases} H_0 : & D_n \leq c \\ H_1 : & D_n > c \end{cases}$$

El umbral c depende del nivel de significancia α y se remite de la condición

$$\alpha = \mathbb{P}(\delta \neq H_0 | H_0) = \mathbb{P}(D_n \geq c | H_0)$$

ya que bajo H_0, la distribución de D_n y el umbral $c = c_\alpha$, pueden tabularse, en cada n.

Cuando n es grande, se puede utilizar la distribución Kolmogorov-Smirnov, $H(\cdot)$, para encontrar c, por

$$\alpha = \mathbb{P}(D_n \geq c \,|H_0) \approx 1 - H(c)$$

dados los valores tabulados de $H(\cdot)$ se encuentra c.