INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Aplicación de cómputo no convencional en la predicción de contaminantes ambientales

TESIS

que para obtener el grado de
Maestro en Ciencias de la Computación

PRESENTA:

Erick Nicolás Cabrera Álvarez

Directores de Tesis

Dr. Cornelio Yáñez Márquez
Dr. Oscar Camacho Nieto

México, D. F. Agosto de 2012

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 25 del mes de junio de 2012 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Aplicación de cómputo no convencional en la predicción de contaminantes ambientales"

Presentada por el alumno:

CABRERA	**ÁLVAREZ**	**ERICK NICOLÁS**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	0	1	6	3	5

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Cornelio Yáñez Márquez

Dr. Oscar Camacho Nieto

Dr. René Luna García

Dr. Miguel Jesús Torres Ruiz

Dr. Luis Octavio López Leyva

Dr. Amadeo José Argüelles Cruz

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de **México DF** el día **27** del mes **Junio** del año **2012**, el que suscribe **Erick Nicolás Cabrera Álvarez** alumno del Programa de **Maestría en Ciencias de la Computación** con número de registro **B101635**, adscrito al **Centro de Investigación en Computación**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección de **Dr. Cornelio Yáñez Márquez y Dr. Oscar Camacho Nieto** y cede los derechos del trabajo intitulado **Aplicación de cómputo no convencional en la predicción de contaminantes ambientales**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **cabrerick@hotmail.com**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Erick Nicolás Cabrera Álvarez

Agradecimientos

A mis padres por ser la fuente principal de felicidad, comprensión, cariño y esfuerzo, sin los cuales no habría podido alcanzar una meta tan importante de mi vida.

A mis Directores de tesis Dr. Cornelio Yáñez Márquez y Dr. Oscar Camacho Nieto sin duda unos excelentes guías, que con su incondicional apoyo, paciencia y dedicación ayudaron a que el presente trabajo de tesis fuera posible.

A mis sinodales Dr. René Luna García, Dr. Miguel Jesús Torres Ruiz, Dr. Amadeo José Argüelles Cruz y Dr. Luis Octavio López Leyva por sus invaluables contribuciones y por sus atinadas críticas las cuales ayudaron a mejorar esta tesis.

Al Centro de Investigación en Computación del IPN, por acogerme y brindarme un lugar de estudio digno, así como a cada uno de los que laboran dentro del mismo, que con profesionalismo y dedicación atendieron cada una de mis solicitudes.

Al Instituto Politécnico Nacional del cual me siento muy orgulloso por ser mi *Alma Mater* desde nivel superior y que ha dejado profundas huellas con sus colores guinda y blanco brillando en mi corazón.

Al CONACyT y a la Secretaría de Investigación y Posgrado por su apoyo económico, factor importante para el desarrollo del presente trabajo de tesis.

Resumen

En este trabajo de tesis se desarrolla y se propone el Método de Predicción por Aproximación de Cadenas (MPAC), el cual se aplica en la predicción de contaminantes atmosféricos presentes en la Ciudad de México.

El MPAC se fundamenta teóricamente en la teoría estadística y en las bondades que exhiben los algoritmos de correspondencia de cadenas, los cuales son algoritmos de búsqueda de subcadenas, por lo que su objetivo es buscar la existencia de subcadenas dentro de una cadena dada.

Específicamente, el MPAC se sustenta sobre la base de algunas modificaciones pertinentes que se realizaron a uno de los algoritmos de correspondencia de cadenas más eficaces y eficientes: el algoritmo de Knuth-Morris-Pratt, el cual ha mostrando un buen desempeño en el tiempo de correspondencia de cadenas, al compararlo con otros algoritmos relevantes en esta rama de la investigación científica, como lo es el algoritmo de Rabin-Karp.

En la parte experimental de esta tesis se muestra la potencia y competitividad del MPAC, al aplicarlo en la predicción de contaminantes incluidos en los bancos de datos del Sistema de Monitoreo Atmosférico de la Ciudad de México (SIMAT), específicamente en la Red Automática de Monitoreo Atmosférico (RAMA), la cual cuenta con 34 estaciones de monitoreo ubicadas en lugares estratégicos de la Ciudad de México.

Dado que en la literatura científica actual (a mayo de 2012) no se reportan trabajos que utilicen algoritmos de correspondencia de cadenas para la predicción de contaminantes atmosféricos, el MPAC puede ubicarse como un método de Cómputo No Convencional. Así, el principal objetivo de este trabajo de tesis es desarrollar un método no convencional para la predicción de contaminantes atmosféricos.

Con el desarrollo de este trabajo de tesis se muestra el advenimiento de

los métodos de cómputo no convencional en la tarea de predicción de contaminantes atmosféricos, como una viable alternativa a los métodos convencionales como las redes neuronales, los métodos de regresión, las máquinas de soporte vectorial, los métodos híbridos basados en lógica difusa, entre otros.

Abstract

In this thesis the Prediction Method by Approach Chains (in Spanish MPAC), is developed and set as a proposal, which is applied in the prediction of air pollutants in Mexico City.

The MPAC is based theoretically on statistical theory and the kindness exhibited by string-matching algorithms, which are substring searching algorithms, so its goal is to find whether the pattern exists within a string as a substring.

Specifically, the MPAC is supported on the basis of some relevant changes, that were made to one of the string-matching algorithms more effective and efficient: the Knuth-Morris-Pratt algorithm, which has shown good performance in the matching time, compared to other relevant algorithms in this area of scientific research, such as the Rabin-Karp algorithm.

In the experimental part of this thesis the power and competitiveness of the MPAC is shown, when applied in the prediction of pollutants contained in the databases Atmospheric Monitoring System of Mexico City (acronym in Spanish SIMAT), specifically in the Automatic Environmental Monitoring Network (acronym in Spanish RAMA), which has 34 monitoring stations located in strategic locations in Mexico City.

Since in the current scientific literature (to May 2012) we can not find any reported work using string-matching algorithms for the prediction of air pollutants, the MPAC can be placed as a method of Unconventional Computing. Thus, the main aim of this thesis is to develop an unconventional method for prediction of air pollutants.

With the development of this thesis the coming of the unconventional computational methods in the prediction task of air pollutants are shown, as a viable alternative to conventional methods such as neural networks,

regression methods, support vector machines, hybrid methods based on fuzzy logic, among others.

Índice general

Índice de figuras

Capítulo 1

Introducción

En este trabajo de tesis se desarrolla una propuesta, la cual se basa en el uso de algoritmos de correspondencia de cadenas y algunas bases de la teoría estadística. Dicha idea, se aplica de manera útil en la predicción de contaminantes atmosféricos.

1.1. Antecedentes

La contaminación atmosférica es un problema fundamental e importante en muchas partes del mundo, en particular en megaciudades como la ciudad de México, donde existe un alta concentración de población, así como de unidades industriales y de transporte, factores que originan un alta concentración de emisiones y que además son el resultado de un crecimiento económico e industrial [1].

La predicción de contaminantes atmosféricos no es una área nueva, sin embargo, es una área con pocos desarrollos [2]. En la literatura científica indexada se encuentran obras desde los años sesenta [3, 4], donde se aplican métodos estadísticos y matemáticos. Posteriormente se marca un crecimiento en los años setenta [5, 6, 7, 8, 9, 10, 11], donde dichos trabajos en su mayoría utilizan modelos estadísticos. Continuando de manera cronológica en los años ochenta es todavía mayor el interés por la predicción de contaminantes atmosféricos [12, 13, 14, 15, 16, 17, 18, 19, 20], mismo que ha seguido creciendo hasta la fecha, donde se han hecho presentes diversas técnicas de inteligencia

artificial.

En la actualidad, la preocupación por la salud humana y la correcta orientación de las políticas públicas que conducen el desarrollo de estándares para minimizar los efectos de la contaminación del aire en la salud humana, ha llevado a distintos investigadores a desarrollar técnicas de cómputo inteligente, que les permitan generar predicciones confiables sobre el índice de calidad del aire o bien sobre los niveles de concentración en ciertos contaminantes atmosféricos como el ozono(O_3), el dióxido de nitrógeno(NO_2), el monóxido de carbono(CO), entre algunos otros.

Varias técnicas de inteligencia artificial se han aplicado a la predicción de contaminantes atmosféricos, entre las cuales se encuentran:

1. Métodos de Regresión [21],[22],[23],[24],[25],[26]
2. Lógica Difusa [27],[28],[29]
3. Programación Genética [30]
4. Sistemas Expertos [31]
5. Redes Neuronales Artificiales [32],[33],[23],[34],[35],[28],[36], [37],[38],[24],[39],[31],[25], [26],[40],[41],[42],[43],[44]
6. Máquinas de Soporte Vectorial [45],[46],[47]
7. Clasificador Gamma [48],[49]

Sin embargo es importante hacer notar que a pesar de que existen diversas técnicas de inteligencia artificial para la predicción de contaminantes atmosféricos, hasta la fecha no existe trabajo alguno de investigación científica que utilice los algoritmos de correspondencia de cadenas para la predicción de contaminantes atmosféricos.

1.2. Justificación

Los algoritmos de correspondencia de cadenas son algoritmos de búsqueda de subcadenas y por lo tanto su objetivo es buscar la existencia de subcadenas

dentro de una cadena. En particular el algoritmo de Knuth-Morris-Pratt (KMP) [50] localiza todas las ocurrencias de una subcadena dentro de una cadena en un tiempo de ejecución proporcional a la suma de la longitud de las cadenas. El algoritmo KMP en comparación con algoritmos como el de *Fuerza Bruta* o el de *Rabin-Karp*, resulta ser más eficiente que ambos [51].

De acuerdo con la búsqueda exhaustiva realizada hasta el mes de Mayo de 2012, en las revistas científicas de prestigio, como las publicadas por (Elsevier, Springer, IEEE) no se reportan trabajos que utilicen los algoritmos de correspondencia de cadenas para la predicción de contaminantes atmosféricos.

El desarrollo en este trabajo de tesis, responde a lo anterior, mediante la propuesta de un método para la predicción de contaminantes atmosféricos que desde el punto de vista de Toffoli [52] esta idea puede ubicarse dentro de lo no-convencional.

1.3. Objetivo

En los últimos años, los algoritmos de optimización global que imitan ciertos principios de la naturaleza han demostrado su utilidad en diversos campos de aplicación. El principal objetivo de este trabajo de tesis es desarrollar un método no convencional para la predicción de contaminantes atmosféricos que nos permita enriquecer o ir más allá de los modelos estándar dentro de la inteligencia artificial.

1.4. Contribuciones

Las contribuciones de este trabajo de tesis, son las siguientes:

- El desarrollo de un método para la predicción de contaminates atmosféricos, basado en correspondencia de cadenas y herramientas básicas de la teoría estadística.
- El desarrollo de una aplicación para la predicción de contaminates atmosféricos.

1.5. Organización del documento

- En este Capítulo se han presentado: los antecedentes, la justificación, el objetivo y las contribuciones de este trabajo de tesis . El resto del documento está organizado como se describe a continuación.

- En el Capítulo 2 se presentan los conceptos básicos y el estado del arte de las principales técnicas que se utilizan en la predicción de contaminantes atmosféricos.

- El capítulo 3 inicia con los tópicos más importantes de las series de tiempo, posteriormente se presentan la notaciones asintóticas que se utilizan para describir el tiempo de ejecución asintótico en un algoritmo, continúa con la formalización del problema de correspondencia de cadenas y por último se describe el algoritmo Knuth-Morris-Pratt, herramienta fundamental en el desarrollo de nuestro método.

- El Capítulo 4 es la parte más relevante de este documento. En el mismo, se introducen las modificaciones al algoritmo Knuth-Morris-Pratt, junto con el algoritmo propuesto que le da sustento a este trabajo de tesis.

- Los resultados experimentales, así como la discusión de los mismos, se presentan en el Capítulo 5; y en el Capítulo final, el 6, se exponen las conclusiones y recomendaciones para trabajo futuro.

- Como parte final en este trabajo de tesis, se presentan los Apéndices A,B y C. En el Apéndice A se presenta la simbología y su correspondiente descripción. En el Apéndice B, se presentan las gráficas de tres métodos de interpolación que se utilizan en MATLAB, posteriormente se presentan gráficamente las concentraciones del CO y del $O3$ a en su forma original y después de haber aplicado la interpolación *pchip* de MATLAB. Finalmente, en el Apéndice C se muestran las imágenes de la distintas modalidades del software desarrollado.

Capítulo 2

Estado del Arte

En el presente capítulo se abordan algunas de las principales técnicas que se utilizan en la predicción de contaminantes atmosféricos, mismas que integran el estado del arte de la Inteligencia Computacional en su tarea de regresión.

En el capítulo previo se menciona que existen varias técnicas que se aplican en la predicción de contaminantes atmosféricos. Sin embargo, dentro de las principales técnicas de inteligencia artificial se encuentran las siguientes [47]:

- Lógica Difusa
- Redes Neuronales
- Máquinas de Soporte Vectorial

2.1. Inteligencia computacional

Dentro de la sociedad de la inteligencia artificial, el término *inteligencia computacional* se entiende en gran medida como un conjunto de metodologías computacionales inteligentes, tales como la computación basada en lógica difusa, la neuro computación y la computación evolutiva, la cual no se describe en este trabajo de tesis. Estas metodologías ayudan a solucionar problemas computacionales complejos en la ciencia y en la tecnología que no se pueden

resolver o al menos no se resuelven fácilmente mediante el uso de los métodos matemáticos convencionales.

De acuerdo con las publicaciones en curso, el término *inteligencia computacional* fue definido por Bezdek [53] en su intento por estudiar la relación entre las redes neuronales, el reconocimiento de patrones y la inteligencia. En su publicación Bezdek establece que la inteligencia computacional utiliza datos numéricos los cuales se obtienen por medio de sensores y que no se trata de conocimiento. Además establece que la Inteligencia Artificial es muy diferente, puesto que trata principalmente con conocimiento no numérico o bien cuando se tiene una clara identificación de una componente no numérica.

Bezdek también intenta clasificar los dos tipos de inteligencia, donde considera a la inteligencia artificial como un *cómputo de medio nivel al estilo de la mente* y a la inteligencia computacional como un *cómputo de bajo nivel al estilo de la mente*. Sin embargo, esta clasificación de los dos tipos de inteligencia vista más o menos desde el aspecto de reconocimiento de patrones y de redes neuronales permanece más como un punto de vista personal del autor que una opinion general.

Después Poole [54] presenta un punto de vista diferente sobre la inteligencia computacional donde considera a la inteligencia computacional como un campo de estudio donde se pueden diseñar agentes inteligentes, es decir sistemas que son capaces de aprender de la experiencia, flexibles al cambio de ambiente así como al cambio de objetivos.

La inteligencia computacional, a pesar de llevar un poco más de una década, ha encontrado su camino en importantes aplicaciones de ingeniería, tales como el modelado, la identificación, la optimización y la predicción necesaria para la toma de decisiones. Esto se debe gracias a los esfuerzos de la investigación en la ampliación de los fundamentos teóricos de las tecnologías de cómputo inteligente, explotando sus posibilidades de aplicación y a la enorme expansión de sus capacidades para hacer frente a los problemas reales.

2.2. Cómputo suave

La primeros avances de la investigación en el ámbito de la aplicación conjunta de tecnologías de cómputo inteligente se debe gracias a Zadeh [55], quien dio origen al término *cómputo suave*, al que definió como un conjunto de

metodologías que tienen como objetivo explotar la tolerancia a la imprecisión y a la incertidumbre para conseguir manejabilidad, robustez y un bajo costo en la solución. Según Zadeh, los principales componentes del cómputo suave son la lógica difusa, la neuro computación y el razonamiento probabilístico.

En opinion de Zadeh la razón de la necesidad del cómputo suave era que vivimos en un mundo impreciso e incierto y que la precisión y la certidumbre requieren de un costo. En adelante, el cómputo suave será visto como una asociación entre distintos métodos y no como un cuerpo homogéneo de conceptos y técnicas.

Se puede decir que la lógica difusa es la parte más importante del cómputo suave la cual cierra la brecha entre la información cuantitativa y la información cualitativa que puede ser procesada conjuntamente utilizando la computación difusa.

2.2.1. Lógica Difusa

La lógica difusa desarrollada por Lotfi Zadeh en 1965 es considerada una generalización de la lógica clásica que desde sus inicios ha venido creciendo y se encuentra en varias áreas de aplicación, además se utiliza para modelar sistemas bajo condiciones de incertidumbre o de imprecisión. En la lógica difusa podemos decir que existe más de una alternativa es decir existe todo un continuo de valores de verdad para las proposiciones lógicas [56].

Cuando realizamos ciertas pruebas y obtenemos una salida siempre es deseable saber el significado o la interpretación de la misma para lo cual es recomendable utilizar los métodos de la lógica difusa debido a que son adecuados para el razonamiento incierto o aproximado, ya que de alguna manera existen casos en donde es difícil de obtener un modelo matemático. Principalmente la lógica difusa permite tomar decisiones con valores estimados en información incompleta o imprecisa.

Al referimos a un conjunto difuso estamos hablando de sus objetos y sus respectivos grados de pertenencia al conjunto, donde el grado de pertenencia de un objeto en el conjunto difuso se define por medio de una función de pertenencia y el valor del grado de pertenencia de un objeto puede ser un valor entre 0 y 1 donde el valor de 1 denota una completa pertenencia al conjunto difuso, pero si dicho valor se aproxima al 0 el grado de pertenencia

al conjunto difuso será cada vez menor [57].

La teoría de conjuntos difusos nos permite expresar en forma numérica que tanta imprecisión o ambigüedad existe en el pensamiento humano o bien al tomar decisiones. Cuando tenemos datos difusos es muy común realizar una codificación o un procesamiento lo cual mejora el conocimiento y de esta manera los procedimientos ambiguos resultan fáciles e intuitivos, ya que como se menciona previamente, el valor de verdad de un enunciado es una cuestión de grado. Esencialmente, la ambigüedad es un tipo de imprecisión que se deriva de una agrupación de elementos en clases que no tienen límites bien definidos. Estas clases llamadas conjuntos difusos, se originan, por ejemplo, cada vez que se describe la ambigüedad, la indefinición y la ambivalencia en los modelos matemáticos de los fenómenos empíricos.

Cuando se trata de describir sistemas en el mundo real, no siempre es adecuado el uso del enfoque binario incluso el ternario para el tratamiento de los fenómenos físicos, ya que es muy común que en los modelos matemáticos se escapen ciertos aspectos de la realidad. En consecuencia los atributos de las variables del sistema pueden surgir como el resultado de causas tales como; una confusión alusiva, un reajuste de contexto, o de la imprecisión humana.

Como uno de sus objetivos de la teoría de conjuntos difusos está el desarrollar una metodología para formular y solucionar problemas que son complicados o que están mal definidos para ser tratados por técnicas convencionales. De acuerdo con algunos autores la lógica difusa tiene una débil conexión con la teoría de la probabilidad. Los métodos probabilísticos que tienen que ver con un conocimiento impreciso se formulan en el esquema bayesiano, pero la lógica difusa [58] no tiene que justificarse mediante un enfoque probabilístico. El camino usual es generalizar los resultados de varios valores lógicos de tal manera que se preserve parte de la estructura algebraica.

Algunos destacados autores han demostrado que existe un fuerte vínculo entre la teoría de conjuntos, la lógica y la geometría [56]. La teoría de conjuntos difusa está en relación con la lógica difusa y la semántica de los operadores difusos, además puede entenderse mediante un modelo geométrico. La visualización geométrica de la lógica difusa es importante porque nos da una pista sobre la posible conexión con las redes neuronales.

La lógica difusa se puede utilizar como un modelo de interpretación de las propiedades de las redes neuronales, así como para dar una descripción más precisa de su desempeño. Además, algunos autores demuestran que los ope-

radores difusos pueden ser concebidos como funciones de salida generalizada de las unidades de cómputo [56]. La lógica difusa también se puede utilizar para especificar ciertas redes directamente sin tener que aplicar un algoritmo de aprendizaje. En algunos trabajos se crean modelos para clasificar los vectores de datos en conjunto de datos multivariados y después dividen el espacio de entrada en un número de subespacios para formar reglas difusas [29].

Un experto en un determinado campo a veces puede producir un simple conjunto de normas de control de un sistema dinámico con menos esfuerzo que el trabajo que implica la formación de una red neuronal. Un ejemplo clásico propuesto por Zadeh, está en desarrollar un sistema de red neuronal para estacionar un coche. Es sencillo de formular un conjunto de reglas difusas para esta tarea, pero no es inmediatamente obvio cómo construir una red para hacer lo mismo, ni la forma de entrenar. La lógica difusa es que ahora se utiliza en muchos productos de electrónica industrial y de consumo para que sea suficiente un buen sistema de control y donde la cuestión del control óptimo no necesariamente surge.

2.2.2. Redes Neuronales Artificiales

El inicio de las redes neuronales artificiales se produjo en 1943 cuando Warren McCulloch, un neurofisiologo, y un joven matemático, Walter Pitts, dieron origen a un tema que trata sobre el funcionamiento de las neuronas [59], donde inicialmente se utilizaron circuitos eléctricos para modelar las redes neuronales simples.

Las redes neuronales están inspiradas en el conocimiento fisiológico de la organización del cerebro. Se estructuran como un conjunto de unidades idénticas interconectadas, conocidas como neuronas. Las interconexiones se utilizan para enviar señales de una neurona a las demás, ya sea en una forma incrementada o inhibida. Esta inhibición o incremento se obtienen mediante el ajuste de pesos. Las redes neuronales pueden llevar a cabo las tareas de clasificación y regresión, ya sea en una forma supervisada o no supervisada. Logran esto mediante métodos apropiados de ajuste de pesos, por medio del cual las salidas de la red optimistamente convergen.

Contrariamente a la clasificación estadística, las redes neuronales tienen la ventaja de ser máquinas de modelo libre, comportándose como aproximadores

universales, capaces de adaptarse a cualquier salida deseada o topología de clases.

Una de las desventajas de las redes neuronales en comparación con la clasificación estadística es que su matemática es más complicada y de acuerdo con la experiencia, frecuentemente el diseñador tiene poca orientación teórica para tomar algunas decisiones importantes y tiene que confiar en el análisis de ensayo y error. Otra desventaja, que puede ser importante en algunas circunstancias, es que prácticamente no hay información semántica disponible en una red neuronal. Para apreciar este último punto, imaginemos que un médico realiza una tarea de diagnóstico con la ayuda de una red neuronal y de un clasificador estadístico, ambos se alimentan con la misma entrada de valores (los síntomas) y proporcionan la respuesta correcta tal vez en contra del conocimiento o la intuición del médico. En el caso del clasificador estadístico, el médico es probablemente capaz de percibir cómo se llegó a la salida, teniendo en cuenta los modelos de distribución. En el caso de la red neuronal, esta percepción suele ser imposible. Sin embargo, las redes neuronales son preferibles a los modelos clásicos de estadística, especialmente cuando el tamaño del conjunto de entrenamiento es pequeño comparado con la dimensión del problema a resolver. Un modelo libre de enfoques, ya sea basado en la clasificación estadística clásica o en las redes neuronales, tienen un cuerpo común de análisis proporcionado por la teoría del aprendizaje estadístico [60]

Como ya se ha dicho una red neuronal está catalogada como un aproximador universal [61], lo que significa que una red neuronal se puede comportar como cualquier función deseada en teoría, aunque lo difícil es encontrar una red y un entrenamiento adecuado, además, no es posible determinar si dicha arquitectura es óptima para el problema en cuestión, porque debido a su estructura, puede darse el caso de que exista una mejor red que nos reduzca dicho error. Se puede decir que prácticamente una red neuronal realiza un mapeo de un espacio de entrada a uno de salida.

Existen diversos teoremas que muestran que una red neuronal, con una sola capa oculta y un número suficiente de neuronas, es suficiente para aproximar cualquier función [62]. Pero también existe una pregunta que la mayoría de los investigadores comparten y se debe a cuántas neuronas son suficientes, podemos afirmar que la respuesta no es simple ya que depende de los datos y del problema. Dependiendo del problema que se trate es como se

tiene que variar el número de neuronas, pero no existe algo que nos indique hasta qué punto debemos de aumentar las neuronas. En el peor de los casos uno podría agregar demasiadas, pero esto originaria que la red no tuviera un buen rendimiento, por lo que es recomendable agregar más capas ocultas y disminuir drásticamente el número de nodos en la capa oculta. De acuerdo con ciertos trabajos se tiene el conocimiento de que con dos capas ocultas se puede resolver la mayoría de los problemas, incluyendo la predicción [62], [61].

Según el tipo de problema, es el tipo de representación que se utiliza, para lo cual existen dos grandes grupos:

- Redes unidireccionales (no tienen ciclos)
- Redes recurrentes (tienen ciclos)

Estos dos tipos de topologías pueden ser clasificadas en: multi-capa, solo una capa, aleatoriamente conectados, localmente conectados, conexiones dispersas, completamente conectados, entre otros. Las redes multicapa son comúnmente usadas para realizar la predicción y debido a que no hay ningún teorema para su construcción, muchos autores utilizan desde 1 capa hasta 3 o 4 para para tratar el problema de predicción [62].

En las redes neuronales el proceso de aprendizaje puede tomar lugar en *modo supervisado* por ejemplo cuando se utilizan las redes de retropropagacion o en *modo no supervisado* cuando se usan redes recurrentes o redes de Kohonen. Esto es debido al perceptron [63], la base teórica que fue desarrollada por Minsky y Papert [64]. Es el perceptron multicapa el cual es capaz de emular el comportamiento del cerebro humano en aprendizaje y cognición. La capacidad de aprendizaje del perceptron multicapa, según lo propuesto por Werbos [65], debería obtenerse a través de un proceso de entrenamiento adaptativo.

2.3. Máquinas de soporte vectorial

Como un método alternativo a las redes neuronales, las máquinas de soporte vectorial desarrolladas por Vapnik, se han utilizado en tareas de clasificación pero en particular para proporcionar métodos para la predicción en series de tiempo [45],[46],[47].

Históricamente, las máquinas de soporte vectorial han sido en gran parte motivadas por un marco teórico conocido como la teoría del aprendizaje computacional, también a veces nombrado como la teoría del aprendizaje estadístico [66]. Esto tiene sus orígenes con Valiant [67] quien formuló el marco de aprendizaje *probablemente aproximadamente correcto* (PAC). El objetivo del PAC es entender qué tan grande debe ser un conjunto de datos a fin de dar una buena generalización.

El problema básico de las máquinas de soporte vectorial, es separar linealmente por medio de un hiperplano los vectores que se mapean de un espacio vectorial formado por los vectores de entrada, a otro espacio vectorial con una mayor dimensión. Sin embargo, existe una complicación debido a que más de un hiperplano puede separar a dichos vectores. Por tal motivo el objetivo principal de las máquinas de soporte vectorial es encontrar el hiperplano óptimo que mejor generalice la clasificación. Para ello, se utiliza el concepto de *vector de soporte*, que se refiere a los patrones más cercanos al hiperplano buscado, donde dichos patrones se encuentran en la frontera de las clases. Para encontrar ese hiperplano óptimo, se maximiza la distancia a los vectores de soporte.

Es importante mencionar que puede darse el caso donde existan patrones cercanos a la frontera de las clases, que se comportan más como excepciones. Si dichos patrones se tomaran como vectores de soporte, degradarían la generalización de la clasificación. Para solucionar este problema, se incluye cierto margen de error, con lo cual se admite que ciertos patrones cercanos a la frontera no sean tomados como vectores de soporte, siempre y cuando se mantengan a una distancia menor o igual al margen de error establecido. Por consiguiente, el problema de encontrar el hiperplano óptimo que divide a ambas clases, se resuelve maximizando la distancia entre dicho hiperplano y los vectores de soporte, a la vez que se minimiza el error [66],[68],[69]

La figura 1 muestra un ejemplo de una máquina de soporte vectorial, donde los vectores de soporte están indicados con un recuadro mientras que el hiperplano óptimo aparece como una línea de color negro y los hiperplanos que pasan por los vectores de soporte de cada una de las clases están de color verde y rojo respectivamente.

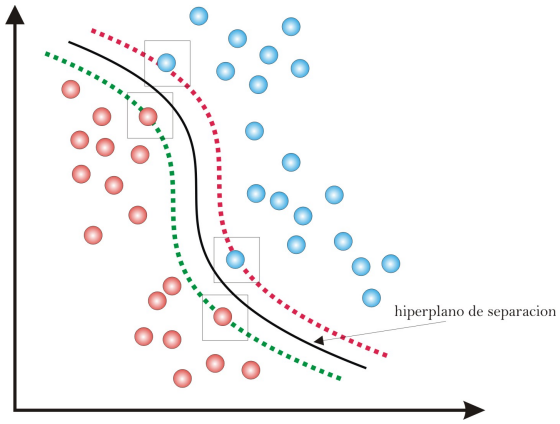


Figura 1. *El entrenamiento de una máquina de soporte vectorial consiste en encontrar el hiperplano óptimo.*

Capítulo 3

Materiales y métodos

Este capítulo se compone de 5 secciones. En la sección 3.1, se hace una breve descripción de las series de tiempo, además, se describen la clasificación, el análisis y los elementos característicos, así como los diferentes modelos de las series de tiempo. En la sección 3.2, se explican las 5 etapas necesarias para la construcción de un modelo ARIMA, dentro de las cuales se incluye la etapa de predicción. En la sección 3.3, se definen las diferentes notaciones asintóticas, que se utilizan para describir el tiempo de ejecución en un algoritmo. En la sección 3.4, se describe y formaliza el problema de correspondencia de cadenas. Por último, en la sección 3.5, se describe el algoritmo Knuth-Morris-Pratt, parte fundamental de nuestro método.

Los conceptos presentados en las secciones 3.1 y 3.2 se han tomado de las referencias que, a nuestro juicio, son las más representativas [70, 71, 72, 73]. Para el caso de las secciones 3.3, 3.4 y 3.5, las referencias más representativas son: [74, 50, 75]

3.1. Introducción a las series de tiempo

Una serie de tiempo es un conjunto de observaciones medidas secuencialmente a través del tiempo [70]. Estas mediciones se pueden realizar de manera continua a través del tiempo o se pueden tomar como un conjunto discreto de puntos en el tiempo. Por convención a estos dos tipos de series se les denomina; serie continua y serie discreta respectivamente. Aunque en

ambos casos la variable medida puede ser discreta o continua. Por ejemplo el eje del tiempo puede verse como una serie de tiempo discreta.

La importancia sobre el análisis y la predicción de las series de tiempo ha aumentado constantemente y sigue siendo de actual interés para ingenieros y científicos. En varios campos es de particular interés la predicción de las series de tiempo donde con base en datos obtenidos se predice el valor futuro.

Para las series de tiempo continuas la variable observada es típicamente una variable continua que se registra continuamente de una señal tal como los registros de la actividad del cerebro en una máquina de electroencefalograma (EEG).

En la práctica los valores se obtienen de sensores tomando muestras de señales continuas. Cuando nos basamos en valores medidos y corruptos por el ruido, los valores de la series de tiempo por lo general contienen una componente estocástica y una componente determinista la cual representa el ruido de interferencia que causa fluctuaciones estadísticas alrededor de los valores determinísticos.

En primer lugar el análisis de una serie de tiempo tiene como meta estudiar su estructura interna por ejemplo la autocorrelación, la tendencia, la estacionalidad, entre otras, para lograr una mejor comprensión del proceso dinámico por el cual se generan los datos de la serie de tiempo.

El término amplio del análisis de series de tiempo incluye actividades como:

- Definición, clasificación y descripción de las series de tiempo
- Construcción de modelos con datos recopilados de series de tiempo
- Pronóstico o predicción de valores futuros

Para pronosticar los valores futuros de una serie de tiempo hay una amplia variedad de métodos disponibles. Desde el punto de vista sistema-teórico ellos pueden ser:

- *Modelo libre* se utiliza en suavización exponencial y análisis de regresión.

- *Basado en modelo* se utiliza particularmente en el modelado de datos de series de tiempo para capturar la característica del comportamiento a largo plazo del sistema dinámico subyacente.

3.1.1. Elementos característicos

Tradicionalmente el análisis de las series de tiempo está definido como una rama de la estadística que generalmente trata con las dependencias estructurales entre los datos de observación y los parámetros correspondientes de los fenómenos aleatorios.

Básicamente, existen 2 enfoques para el análisis de las series de tiempo:

- *En el dominio del tiempo*, principalmente se basan en el uso de la función de covarianza de las series de tiempo.
- *En el dominio de la frecuencia*, se basa en análisis de Fourier y en el análisis de la función de densidad.

ambos enfoques son apropiadas para aplicarse a un amplio campo de disciplinas, pero el enfoque en el dominio del tiempo es el más utilizado en la práctica. Esto se debe particularmente a la flexibilidad en la metodología de Box-Jenkins [71] para el análisis de series de tiempo, el cual se refiere a la aproximación lineal de fenómenos estacionarios. Sin embargo Box y Jenkins han señalado que su planteamiento también se puede aplicar al análisis de series de tiempo no estacionarias después de su diferenciación (eliminación de la tendencia).

Las principales características de las series de tiempo son la estacionariedad, la linealidad, la tendencia y la estacionalidad. Aunque una serie de tiempo puede presentar una o más de estas características.

Para el análisis y la predicción de las series de tiempo los valores de cada característica se trabajan mejor por separado.

Estacionariedad

La estacionariedad de un proceso aleatorio está relacionada con la media y la varianza de los datos de observación, ambos de los cuales deben ser constantes en el tiempo y la covarianza entre x_t y x_{t-d} debe depender solamente

de la distancia entre las dos observaciones y no cambiar con el tiempo, es decir se deben cumplir las siguientes relaciones:

$$Ex_t = \mu \tag{3.1}$$
$$Var(x_t) = E(x_t - \mu)^2 = k_0 \tag{3.2}$$
$$Cov(x_t, x_{t-d}) = E(x_t - \mu)(x_{t-d} - \mu) = k_d \tag{3.3}$$

Para $t = 1, 2, \ldots,$ y $d = \ldots, -2, -1, 0, 1, 2, \ldots$ donde μ, k_0 y k_d son constantes.

En términos estadísticos una serie de tiempo es estacionaria cuando el proceso estocástico subyacente está en un estado particular de equilibrio estadístico, es decir cuando la distribución conjunta de $X(t)$ y $X(t-\tau)$ dependen solo de τ y no de t. En consecuencia el modelo estacionario de una serie de tiempo se puede construir fácilmente si el proceso permanece en estado de equilibrio para todos los tiempos cercanos a una constante.

Es difícil verificar si al mismo tiempo una determinada serie de tiempo cumple las tres condiciones de estacionariedad formuladas previamente. Una serie de tiempo es reconocida como estacionaria cuando es representada por una muestra de aspecto lineal sin ninguna tendencia o estacionalidad y con autocorrelación y varianza ambas invariantes en el tiempo.

Se puede verificar fácilmente cuando en cierto modelo de serie de tiempo está disponible la estacionalidad del proceso que genera los valores de observación de la serie. Por ejemplo para el proceso auto-regresivo de primer orden.

$$x_t = \theta x_{t-1} + \varepsilon_t \tag{3.4}$$

la condición de estacionalidad requiere que la condición

$$Var(x_t) = Var(x_{t+1}) \tag{3.5}$$

o la igualdad

$$E\{[\theta x_{t-1} + \varepsilon t]^2\} = E\{[\theta x_{t-2} + \varepsilon(t-1)]^2\} \tag{3.6}$$

se cumpla. Por lo tanto debido a la mutua independencia de ε_t y x_t se sigue la igualdad

$$Var(x_t) = \theta^2 Var(x_{t-1}) + Var(\varepsilon_t) \tag{3.7}$$

y finalmente se tiene la igualdad

$$k_0 = \theta^2 k_0 + \sigma^2, |\theta| << 1 \tag{3.8}$$

donde k_0 no depende del tiempo t.

Una serie de tiempo no estacionaria se puede transformar en una serie de tiempo equivalente de tipo estacionario tomando las diferencias entre los valores consecutivos a lo largo de toda la muestra de la serie de tiempo, es decir mediante diferenciación múltiple o simple en los datos de una determinada serie de tiempo. Este método se recomienda generalmente debido a que algunas series de tiempo buscando ser estacionarias pueden llegar a ser no estacionarias. Para resolver experimentalmente el problema de estacionariedad, primero la serie debe ser dividida en dos o más segmentos que son aparentemente estacionarios, entonces la autocorrelación y las propiedades de cada segmento se verifican y se comparan los resultados.

Linealidad

La linealidad en una serie de tiempo indica que la forma de la serie de tiempo depende de su estado, por lo que el estado actual determina el patrón de la serie de tiempo local. Si una serie de tiempo es lineal entonces puede ser representada por una función lineal de valor presente y de valores pasados. Algunos ejemplos de representaciones lineales son autoregresivos (AR), promedios móviles (PM) (*siglas en inglés* MA), autoregresivo con promedios móviles (*acrónimo en inglés* ARMA) y los modelos Autoregresivos Integrados de Promedios Móviles (*acrónimo en inglés* ARIMA) basados en técnicas de regresión o de promedios móviles. Las series de tiempo no lineales pueden representarse por medio de modelos no lineales o bilineales.

En adelante, se utilizaran los acrónimo en ingles ARMA y ARIMA, debido a que es la forma más común en la mayor parte de la literatura que describe la metodología Box-Jenkins.

Las series de tiempo representadas por medio del siguiente modelo lineal

$$X_t = \sum_{i=-\infty}^{\infty} \Psi_i Z_{t-i} \tag{3.9}$$

generalmente describen un proceso lineal, donde Ψ_i es un conjunto de constantes que satisfacen la siguiente condición

$$X_t = \sum_{i=-\infty}^{\infty} |\Psi_i| < \infty \tag{3.10}$$

y $|Z_t|$ es ruido blanco con media cero y varianza σ^2.

La forma multivariable de un proceso lineal está estadísticamente definida por la relación

$$X_t = \sum_{i=-\infty}^{\infty} C_i Z_{t-i} \tag{3.11}$$

Donde $|C_i|$ representa una sucesión de matrices $n \times n$ con elementos que se pueden sumar en valor absoluto y $|Z_t|$ es el ruido blanco con media cero y matriz de covarianza Σ.

Tendencia

La componente de tendencia de una serie de tiempo es su característica a largo plazo que se manifiesta a través del incremento o decremento local o global en los valores de los datos como una consecuencia de superposición de los valores correctos de la serie de tiempo y una perturbación con tendencia ascendente o descendente. La presencia de una componente de perturbación es detectable si se persiguen los cambios en la media para ciertos intervalos de tiempo sucesivos a través de la serie de tiempo patrón.

El análisis de tendencia es importante en la predicción de series de tiempo. En la práctica se lleva a cabo utilizando técnicas de regresión lineal y no lineal que satisfactoriamente ayudan a identificar componentes de tendencia no monótonas en la serie de tiempo. Por ejemplo para identificar el carácter de la tendencia presente en una serie de tiempo, se utiliza la relación lineal,

polinomial o exponencial para el ajuste de los datos recopilados.

$$x_t = \alpha t + \beta + \varepsilon_t$$
$$x_t = \alpha t + \beta t^2 + \gamma + \varepsilon_t$$
$$x_t = \exp(\alpha t + \beta + \varepsilon_t)$$

Estacionalidad

La componente de estacionalidad de una serie de tiempo se muestra a través de su patrón oscilante de forma periódica. Esta característica es más común en las series de tiempo de tipo económico y en las series de tiempo donde las observaciones se obtienen de la vida real, donde la muestra se puede repetir cada hora, cada día, cada semana, cada mes o cada año. Por lo tanto el objetivo del analizar la serie de tiempo estacional se centra en la detección de la naturaleza de sus fluctuaciones periódicas y en su interpretación.

Estimación y eliminación de la tendencia y la estacionalidad

Cuando dos o más series de tiempo con características diferentes se superponen o cuando una serie de tiempo está superpuesta por la componente de estacionalidad o por la componente de tendencia, es necesario un análisis de descomposición para discriminar y separar las componentes individuales que están involucradas. Se utiliza con mayor frecuencia el análisis de descomposición para quitar la tendencia y la estacionalidad en los datos de la serie de tiempo.

3.1.2. Clasificación de las series de tiempo

Dependiendo de la naturaleza de los datos que la serie de tiempo contiene, esta se puede clasificar de la siguiente manera:

1. Estacionaria o no estacionaria
2. Estacional o no estacional
3. Lineal o no lineal

4. Univariable o multivariable

5. Caótica

En la práctica podemos encontrar series de tiempo con dos o más propiedades de las mencionadas arriba. Por ejemplo una serie de tiempo lineal puede ser estacionaria, estacional y puede tener la componente de tendencia incorporada.

3.1.3. Análisis de las series de tiempo

El análisis de las series de tiempo se refiere a los problemas de identificación de características básicas de las series de tiempo así como al descubrimiento de la estructura interna de la serie de tiempo.

Objetivos del análisis

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

- Construir modelos de entrada-salida que representen a las funciones de transferencia equivalentes a los procesos detrás de la serie de tiempo

- Pronosticar los valores futuros a partir de los valores en el pasado de la serie de tiempo utilizando los modelos desarrollados

- Diseñar sistemas de control basados en los resultados del análisis.

Una vez que el modelo de la serie de tiempo se ha desarrollado y probado se puede utilizar para pronosticar los valores futuros en la serie de tiempo para varias distancias de tiempo d. Por supuesto el pronóstico no asesta los valores futuros exactos con respecto a los datos que realmente tiene la serie de tiempo, si no más bien entrega estimaciones.

Funciones FAC y FACP

Los modelos de predicción de Box-Jenkins se identifican en forma tentativa examinando el comportamiento de la función de autocorrelación (FAC) y la función de autocorrelación parcial (FACP) para los valores de una serie temporal estacionaria.

Un instrumento útil en el análisis de series de tiempo es la función de autocorrelación. En una serie de tiempo $x_1, x_2, \ldots, x_n$ si los valores registrados en el tiempo tienen alguna relación entre ellos, se puede pensar en aplicar procedimientos específicos del análisis de series de tiempo para pronosticar valores futuros de la serie. Si no existiera correlación entre los valores de la serie, entonces estaríamos hablando de una serie de valores independientes, para los cuales, el método de inferencia estadística, es diferente.

Para determinar si nuestra serie cumple con esta condición definimos la llamada función de autocorrelación. La FAC como veremos más adelante puede ayudarnos a detectar aspectos de una serie tales como la estacionariedad o la estacionalidad. También la función de autocorrelación parcial nos permite determinar el orden de los llamados modelos autoregresivos, como revisaremos más adelante.

La función de autocorrelación

La función de auto-correlación de la serie de tiempo $x_1, x_2, \ldots, x_n$, con retraso k está dada por:

$$r_k = \frac{\sum_{t=1}^{n-k}(x_t - \mu)(x_{t+k} - \mu)}{\sum_{t=1}^{n}(x_t - \mu)^2} \tag{3.12}$$

Esta cantidad mide la relación lineal entre las observaciones de la serie de tiempo separadas por un desfasamiento de k unidades de tiempo. Se puede demostrar que siempre estará entre -1 y 1 donde un valor cercano a 1 quiere decir que las observaciones separadas por un desfasamiento de k unidades de tiempo tienen una fuerte tendencia a moverse juntas en forma lineal con pendiente positiva. Por otro lado un valor cercano a -1 significa que las observaciones separadas por un desfasamiento de k unidades de tiempo, tienen una fuerte tendencia a desplazarse juntas en forma lineal con pendiente negativa.

Un correlograma es una gráfica de $r-k$ contra k. La inspección visual del correlograma es útil, pues nos permite observar que tanta correlación existe entre los elementos de la serie. Los aspectos principales a considerar son:

- Serie Aleatoria. Si la serie de tiempo es completamente al azar, (hay independencia en los valores observados) entonces, r_k es igual a cero para todos los valores de k.
- Correlación a corto plazo. Una serie estacionaria frecuentemente exhibe correlación a corto plazo, que se caracteriza por un valor alto de r1 seguida de dos o tres coeficientes significativamente mayores a cero y luego los valores de r_k tienen valores cercanos a cero.
- Series no estacionarias. En una serie no estacionaria, los valores de r_k se van a cero lentamente.
- Fluctuaciones Estacionales. Si una serie de tiempo tiene fluctuaciones estacionales, entonces el correlograma exhibe también oscilaciones con la misma frecuencia.

La función de autocorrelación parcial

Se define como el exceso de correlación en el retraso k que no puede ser explicada por un modelo $AR(k-1)$, es una mención útil para reconocer el orden de los modelos AR. La función de autocorrelación parcial es una lista o una gráfica de las autocorrelaciones parciales de la muestra en los desfasamientos $k = 1, 2, \ldots$.

La función de autocorrelación parcial de la muestra en el desfasamiento k es:

$$r_{kk} = \begin{cases} r_1, & \text{si } k = 1 \\ \dfrac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j}, & \text{si } k = 2, 3, \ldots \end{cases} \tag{3.13}$$

En este caso no es precisa la interpretación de la autocorrelación parcial de la muestra en el desfasamiento k. Sin embargo se podría pensar de manera intuitiva que esta cantidad es la función de autocorrelación muestral de las observaciones de la serie de tiempo separadas por un desfasamiento de k unidades de tiempo sin los efectos de las observaciones que intervienen.

En la siguiente tabla se dan algunos lineamientos generales acerca de los patrones típicos para la FAC y la FACP.

Tipo de modelo	Patrón típico de FAC	Patrón típico de FACP
AR(p)	Disminuye exponencialmente o con un patrón sinusoidal decreciente o ambos	Picos grandes a lo largo de los p rezagos
PM(q)	Picos grandes a lo largo de los q rezagos	Decrece exponencialmente
ARMA(p,q)	Decrece exponencialmente	Decrece exponencialmente

3.1.4. Modelos en las series de tiempo

En la estadística, se utilizan dos modelos matemáticos:

- Modelos determinísticos, matemáticamente vistos como modelos analíticos representados por relaciones deterministas como

$$x_t = f(t)$$

o por ecuaciones recurrentes como

$$x_t = f(x_{t-1}, x_{t-2}, x_{t-3}, \ldots)$$

- Modelos estocásticos, estadísticamente vistos como funciones de variables aleatorias

Los modelos matemáticos que se utilizan generalmente para el análisis de las series de tiempo son:

- Modelos de regresión
- Modelos de dominio en el tiempo
- Modelos de dominio en la frecuencia

Mientras que los modelos de dominio en el tiempo pueden ser

- Modelos de función de transferencia
- Modelos de espacio de estado

Modelos de regresión

Los modelos de regresión se construyen utilizando análisis de regresión, el cual se puede ver como una colección de métodos para el estudio de las relaciones existentes entre las variables de predicción y la estimación de valores de una variable, utilizando los valores de otras variables incorporadas en una serie de tiempo unificada.

En las siguientes secciones se hará mención acerca de los modelos de regresión que son la base para la construcción del modelo ARIMA.

Modelo autoregresivo (AR)

Los modelos autoregresivos expresan el valor actual de una serie de tiempo por medio de una sumatoria lineal de valores previos más un término de error μ_t

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + \mu_t \tag{3.14}$$

donde $\alpha_1, \alpha_2, \ldots, \alpha_p$ son parámetros autoregresivos, μ_t es el ruido blanco y p el orden del modelo. La validez de un modelo autoregresivo supone que la serie de tiempo a ser modelada es estacionaria. También debido a ciertos efectos que se acumulan dentro del proceso autoregresivo se puede afirmar que dicho proceso será estable si los valores de los parámetros α están dentro de un cierto rango.

Es común escribir la ecuación autoregresiva en términos de desviaciones $\widetilde{x_t} = x_t - \mu_t$, usando generalmente la variable Z y su desviación $\widetilde{Z} = Z - \mu$. Los términos individuales de la serie de tiempo ahora se convierten en $\widetilde{Z}_t, \widetilde{Z}_{t-1}, \widetilde{Z}_{t-2}, \ldots$, resultando en el modelo autoregresivo

$$\widetilde{Z}_t = \phi_1 \widetilde{Z}_{t-1} + \phi_2 \widetilde{Z}_{t-2} + \ldots + \phi_p \widetilde{Z}_{t-p} + a_t \tag{3.15}$$

donde $\mu, \phi_1, \phi_2, \ldots, \phi_p, \sigma_a^2$ son parámetros desconocidos que serán estimados de los datos de observación. Introduciendo el operador autoregresivo

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3 - \cdots - \phi_p B^p \tag{3.16}$$

el modelo autoregresivo puede ser escrito en la forma compacta

$$\phi(B)\widetilde{Z}_t = a_t \tag{3.17}$$

El modelo contiene $(p+2)$ parámetros desconocidos es decir p parámetros internos y dos parámetros adicionales: la varianza σ_a^2 y el ruido blanco a_t.

Un problema crucial en el modelado de series de tiempo autoregresivas es sin duda la selección del orden del modelo a construir. Un método útil en este caso, es el análisis de la función de autocorrelación parcial y la función de autocorrelación inversa, pero en el caso de los modelos de orden superior el uso de la función de autocorrelación es computacionalmente complicada. Alternativamente ajustando a la forma de la serie de tiempo mediante modelos de orden progresivamente mayor, se puede utilizar el análisis de la suma de cuadrados residuales para cada orden.

Modelo de promedios móviles (PM)

Otro enfoque que frecuentemente se utiliza en el modelado de las series de tiempo univariadas se basa en el modelo de promedios móviles

$$\widetilde{Z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \tag{3.18}$$

el cual expresa $\widetilde{Z}_t$ en términos de una suma infinita de a's. Al introducir el operador de promedio móvil de orden q

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \theta_3 B^3 - \cdots - \theta_q B^q \tag{3.19}$$

el modelo de promedio móvil puede ser escrito de forma compacta como

$$\widetilde{Z}_t = \theta(B) a_t \tag{3.20}$$

Modelo ARMA

La combinación de los modelos AR y PA integran el modelo ARMA (*acrónimo que comúnmente se utiliza*).

$$\widetilde{Z}_t = \phi_1 \widetilde{Z}_{t-1} + \phi_2 \widetilde{Z}_{t-2} + \ldots + \phi_p \widetilde{Z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \tag{3.21}$$

Reescribiendo el modelo como

$$\widetilde{Z}_t - \phi_1 \widetilde{Z}_{t-1} + \phi_2 \widetilde{Z}_{t-2} + \ldots + \phi_p \widetilde{Z}_{t-p} = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q} \tag{3.22}$$

y reordenando nos queda como

$$(1 - \phi_1 B + \phi_2 B^2 - \ldots - \phi_p B^p)\widetilde{Z}_t = (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)a_t \quad (3.23)$$

entonces el modelo puede finalmente escribirse de forma compacta como

$$phi(B)\widetilde{Z}_t = \theta(B)a_t \quad (3.24)$$

donde B es el operador de retardo. El modelo compacto contiene $(p + q + 2)$ parámetros desconocidos $\mu, \phi_1, \phi_2, \ldots, \phi_p$ y $\theta_1, \theta_2, \ldots, \theta_q, \sigma_a^2$ los cuales serán estimados de la serie de datos.

En la práctica para la representar la realidad de lo que ocurre en una serie de tiempo estacionaria, con frecuencia es lo suficientemente adecuado tomar p y q no mayor que 2. La presencia de ambos términos autoregresivo y promedio móvil en el modelo ARIMA permite la representación de series más complejas.

Modelo ARIMA

Esta variante del modelo ARMA está predestinado para aplicarse a series de tiempo no estacionarias que por medio de la diferenciación llegan a convertirse en estacionarias. La diferenciación es una operación por medio de la cual se construye una nueva serie de tiempo tomando las diferencias de manera sucesiva para valores sucesivos tales como $X(t) - X(t-1)$ a lo largo de la serie de tiempo no estacionaria.

3.2. Pronósticos mediante la metodología Box-Jenkins

Para construir un modelo ARIMA que aproxime aceptablemente las características de una serie de tiempo se hace uso de la metodología de Box-Jenkins, la cual puede ser estructurada en cinco etapas.

3.2.1. Análisis exploratorio de la serie

Se grafica la serie a través del tiempo, de manera que se puedan observar a priori sus componentes: tendencia, estacionalidad y ciclos. Podría notarse

también la necesidad de aplicar diferencias, en la parte estacional o regular, para hacer que la media sea constante, así como su varianza homogénea.

3.2.2. Identificación del modelo

Se debe sugerir un conjunto reducido de posibles modelos:

- Selección del conjunto de estimación: conjunto de datos que se usará para la estimación y adecuación del modelo y el conjunto de predicción: conjunto de datos para evaluar las predicciones.
- Determinación de la función de autocorrelación, la función de autocorrelación parcial y sus correspondientes correlogramas.
- Determinación del orden del componente autoregresivo p y promedio móvil q del modelo ARMA (p, q), haciendo uso de los patrones que se observan en los correlogramas simple y parcial.
- Estudio de la estacionariedad. Si la serie no es estacionaria, se debe convertir en estacionaria antes de aplicar la metodología Box-Jenkins.
- Especificación del modelo ARIMA identificado.

3.2.3. Estimación de parámetros

Una vez identificado el modelo, se obtienen los parámetros mediante la minimización de la suma del cuadrado de los errores. En la actualidad, existen diferentes paquetes estadísticos, que nos permiten realizar estimaciones de manera fácil y precisa, tal es el caso del software estadístico MINITAB, cuya posición se encuentra entre los más populares, además de ser utilizado en trabajos científicos publicados en revistas de prestigio [76].

3.2.4. Verificación de diagnóstico

Después de seleccionar un modelo ARIMA particular y de estimar sus parámetros, se trata entonces de ver si el modelo seleccionado se ajusta a los datos en forma razonablemente buena, ya que es posible que exista otro

modelo ARIMA que también lo haga. Es por esto que el diseño de modelos ARIMA de Box-Jenkins se ve algunas veces como arte más que como ciencia; se requiere gran habilidad para seleccionar el modelo ARIMA correcto. Una prueba simple del modelo seleccionado es ver si los residuales estimados a partir de este modelo son de ruido blanco, si lo son, puede aceptarse el ajuste particular, si no lo son, debe empezarse nuevamente. Por tanto, la metodología Box-Jenkins puede llegar a ser un proceso iterativo.

3.2.5. Pronóstico

El pronóstico se basa en el modelo ARIMA seleccionado. Se calculan los errores de predicción. Es importante determinar la adecuación del modelo en función de qué tan bien se pronostica con ciertos valores que no se utilizaron para su estimación. Para evaluar qué tan cercano es el valor pronosticado y el real, se utiliza el error cuadrático medio (ECM).

3.3. Notación asintótica

Para describir el tiempo de ejecución asintótico de un algoritmo se hará referencia a las notaciones Θ, O y Ω las cuales se definen por medio de funciones cuyos dominios son el sistema de los números naturales $N = \{0, 1, 2,...\}$, estas notaciones describen el tiempo de ejecución en el peor de los casos para la función $T(n)$ que depende de entradas enteras. Podemos encontrar que la mayoría de los autores abusan en el uso formal de la notación asintótica por ejemplo, se puede acotar a un subconjunto del dominio de los naturales o bien se puede extender al dominio de los números reales. Sin embargo, basta con solo entender el significado exacto de la notación para darnos cuenta cuando se está abusando [74].

3.3.1. Notación asintótica– Θ

Dada una función $g(n)$, se define al siguiente conjunto $\Theta(g(n))$ como sigue:

$\Theta(g(n)) = \{f(n) :$ existen las constantes positivas c_1, c_2 y n_0 tales que $0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ para todo $n \geq n_0\}$.

Donde una función $f(n)$ se dice que pertenece al conjunto $\Theta(g(n))$ si existen constantes positivas c_1 y c_2 tales que, $c_1 g(n)$ siempre está por debajo de $f(n)$ y $c_2 g(n)$ siempre está por encima de $f(n)$, para n lo suficientemente grande. Siempre que nos referimos a un elemento que pertenece a un conjunto utilizamos el símbolo $\in$, por tal motivo $f(n) \in \Theta(g(n))$ debido a que $\Theta(g(n))$ es un conjunto. En adelante escribiremos $f(n) = \Theta(g(n))$ para expresar la pertenencia al conjunto, aunque puede parecer confuso veremos más adelante que tiene sus ventajas.

Para tener una idea gráfica acerca de las funciones $f(n)$ y $g(n)$ en donde $f(n) = \Theta(g(n))$ veamos la figura 2(a). Observemos que para todos los valores n mayores que n_0, la función $f(n)$ permanece en medio de ambas funciones $c_1 g(n)$ y $c_2 g(n)$. Se dice que $g(n)$ es una cota que se ajusta asintóticamente a $f(n)$.

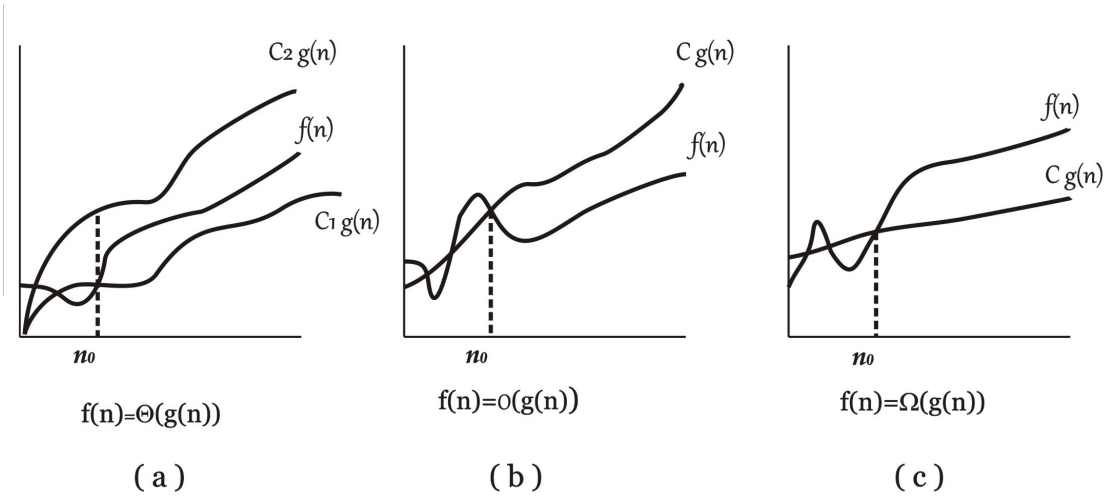


Figura 2. *Ejemplo gráfico de las notaciones θ,O y Ω.*

En necesario que todo elemento $f(n)$ en $\Theta(g(n))$ sea asintóticamente no negativo para definir $\Theta(g(n))$ es decir, la función $f(n)$ debe ser no negativa para n lo suficientemente grande. Por lo tanto, $g(n)$ debe ser asintóticamente

no negativa, o en caso contrario el conjunto $\Theta(g(n))$ sera igual al conjunto vació. En consecuencia toda función que pertenece al conjunto Θ es no negativa. Esta suposición se preserva para las demás notaciones asintóticas.

Ejemplo 1. *Usando la definición formal mostrar que:*

$$\sqrt{n+10} = \theta(\sqrt{n})$$

Solución. *De acuerdo con la definición debemos encontrar c_1, c_2 y n_0 números positivos, tales que se cumple la siguiente desigualdad*

$$c_1\sqrt{n} \leq \sqrt{n+10} \leq c_2\sqrt{n} \text{ para todo } n \geq n_0\}$$

Si tomamos $c_1 = 1, c_2 = \sqrt{2}$ y $n_0 = 10$. Entonces si $n \geq n_0$, por un lado de la desigualdad tenemos:

$$-10 \leq 0 \tag{3.25}$$
$$-10 \leq (1-1) \tag{3.26}$$
$$-10 \leq (1-c_1^2)n \tag{3.27}$$
$$c_1^2 n \leq n+10 \tag{3.28}$$
$$c_1\sqrt{n} \leq \sqrt{n+10} \tag{3.29}$$

ahora considerando el siguiente lado de la desigualdad tenemos:

$$10 \leq n \tag{3.30}$$
$$10 \leq (2-1)n \tag{3.31}$$
$$-10 \leq (c_2^2-1)n \tag{3.32}$$
$$n+10 \leq c_2^2 n \tag{3.33}$$
$$\sqrt{n+10} \leq c_2\sqrt{n} \tag{3.34}$$

por lo tanto de la ecuación 3.29 y 3.34 se obtiene lo deseado.

3.3.2. Notación asintótica– O

Como se explicó anteriormente la notación Θ acota de manera asintótica a una función por arriba y por abajo, pero en el caso donde tenemos sólo una cota superior asintótica, entonces utilizaremos la notación O.

Dada una determinada función $g(n)$ se define al conjunto de funciones $O(g(n))$ como sigue:

$$O(g(n)) = \{f(n) | \exists\, c > 0 \in \mathbb{R}^+ \, y \, n_0 \in \mathbb{N} \pitchfork 0 \leq f(n) \leq cg(n) \, \forall n \geq n_0\}$$

de acuerdo con lo anterior la notación asintótica O define una cota superior para una función $f(n)$ dentro de un intervalo. La figura 2(b) que se mostró en la sección anterior nos da una idea intuitiva donde podemos observar que para todos los valores n mayores que n_0, el valor de la función $f(n)$ está por debajo de $cg(n)$.

Al escribir $f(n) = O(g(n))$ estamos indicando que la función $f(n)$ pertenece al conjunto $O(g(n))$. Debido a que $O(g(n))$ y $\theta(g(n))$ son conjuntos, de manera simple se puede verificar que $\theta(g(n)) \subseteq O(g(n))$ por lo cual si $f(n) = \theta(g(n))$ entonces $f(n) = O(g(n))$, dado que la notación O se define con menos restricciones que la notación θ.

Es muy común en la literatura encontrar errores donde se utiliza de manera informal la notación O para describir una cota que se ajusta asintóticamente, sin embargo cuando escribimos $f(n) = O(g(n))$ estamos diciendo que existe una constante positiva c, tal que $cg(n)$ es una cota superior asintótica para $f(n)$.Cuando queremos describir el tiempo de ejecución de cierto algoritmo es muy frecuente que se utilice la notación O, pues basta con simplemente revisar la estructura completa del algoritmo.

Ejemplo 2. *Usando la definición formal de la notación asintótica O mostrar que:*

$$n = O(e^n)$$

Solución. *Necesitamos mostrar que existen c y n_0 constantes positivas tales que $n \leq ce^n$ para todo $n \geq n_0$. Esto es equivalente a mostrar que la función cociente $q(n) = \frac{n}{e^n}$ está acotada en el intervalo $[n_0, \infty)$ con n_0 apropiado. Para ver esto, observe que la función $q(n)$ es no negativa y continua en el intervalo $[n_0, \infty)$, donde $q(n) = 0$ cuando $n = 0$ y $q(n)$ tiende a 0 cuando $n \to \infty$. Por lo tanto $q(n)$ está acotada en el intervalo $[n_0, \infty)$, ahora basta con solo tomar $n_0 = 0$ y c igual a cualquier valor que sea cota superior de $q(n)$ en el intervalo $[n_0, \infty)$ para que se cumpla la desigualdad deseada.*

3.3.3. Notación asintótica– Ω

Una vez que hemos visto como se describe una cota superior asintótica sobre una función en la definición de la notación O, ahora toca el turno a la notación Ω la cual describe una cota inferior asintótica.

Así, para una determinada función $g(n)$ se denotará al conjunto de funciones $\Omega(g(n))$ como sigue:

$$\Omega(g(n)) = \{f(n) : \exists\, c \in \mathbb{R}^+ \; y \; n_0 \in \mathbb{N} \ni 0 \leq cg(n) \leq f(n) \; \forall \; n \geq n_0\}.$$

Nuevamente para generar una idea intuitiva de la notación Ω hacemos referencia a la figura 2(c), donde podemos ver que los valores de $f(n)$ están por arriba de $g(n)$ para todo n mayor que n_0.

Teorema 1. *Para cualesquiera dos funciones $f(n)$ y $g(n)$, tenemos que $f(n) = \theta(g(n))$ si y sólo si $f(n) = O(g(n))$ y $f(n) = \Omega(g(n))$.*

Demostración:

$\Rightarrow$)Supongamos que $f(n) = \theta(g(n))$ por definición de la notación asintótica θ sabemos que existen c_1 y c_2 tales que

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

esto implicaría que $f(n) = O(g(n))$ y $f(n) = \Omega(g(n))$

$\Leftarrow$)El regreso es análogo. □

La notación asintótica también se puede utilizar en fórmulas matemáticas, por ejemplo, podemos escribir $n = O(n^2)$, o bien $2n^2 + 3n + 1 = 2n^2 + \theta(n)$

¿Pero como se interpretan estas fórmulas?.

En la primer fórmula vemos que la notación asintótica está sólo del lado derecho de la ecuación, decimos que el signo de igualdad indica que $n \in O(n^2)$ en cuanto a la segunda fórmula decimos que $2n^2 + 3n + 1 = 2n^2 + f(n)$ dónde $f(n)$ es una función de $\theta(n)$, en este caso, $f(n) = 3n + 1$, está en $\theta(n)$.

Como otro ejemplo, está la siguiente ecuación $2n^2 + \theta(n) = \theta(n^2)$, interpretamos tal ecuación de la siguiente manera, hay una función $f(n) \in \theta(n)$ y otra función $g(n) \in \theta(n^2)$ tal que $2n^2 + f(n) = g(n) \;\; \forall n$. Entonces la ecuación quedaría de la siguiente forma:

$$2n^2 + 3n + 1 = 2n^2 + \theta(n) = \theta(n^2).$$

Interpretamos cada ecuación por separado. La primer ecuación nos dice que hay una función $f(n) \in \theta(n)$ tal que $2n^2 + 3n + 1 = 2n^2 + f(n) \forall\ n \in \mathbb{N}$.

La segunda ecuación nos dice que para una función $g(n) \in \theta(n)$, hay una función $h(n) \in \theta(n^2)$ tal que $2n^2 + g(n) = h(n)\ \forall\ n$, observe que esta interpretación implica que $2n^2 + 3n + 1 = \theta(n^2)$.

3.4. La correspondencia de cadenas

Las cadenas son evidentemente el centro de los sistemas de edición de textos, tales sistemas procesan cadenas alfanuméricas que pueden definirse en primera aproximación como series de letras, números y caracteres especiales. Donde estos objetos pueden ser demasiado grandes, por lo que es importante disponer de algoritmos eficaces para su manipulación.

Encontrar todas las apariciones de un patrón en un texto es un problema que surge con frecuencia en los programas de edición de textos. Los algoritmos eficientes para este problema pueden ser de gran ayuda a la capacidad de respuesta de los programas de edición de texto. Los algoritmos de correspondencia de cadenas son también utilizados, por ejemplo, para buscar patrones específicos en las secuencias de *ADN*.

Formalizando *el problema de correspondencia de cadenas* podemos suponer que el texto es un vector $T[1 \ldots \mathrm{n}]$ de longitud n y que el patrón es un vector $P[1 \ldots m]$ de longitud $m \leq n$. También podemos suponer que los elementos de P y T son caracteres extraídos de un alfabeto finito Σ. Por ejemplo, tenemos $\Sigma = \{0, 1\}$ o $\Sigma = \{a, b, \ldots, z\}$. Los vectores de caracteres P y T son a menudo llamados cadenas de caracteres.

Diremos que el patrón P se encontró con el desplazamiento s en el texto T, si $0 \leq s \leq n-m$ y $T[s+1 \ldots s+m] = P[1 \ldots m]$ (es decir, si $T[s+j] = P[j]$, para $1 \leq j \leq m$). Si P se encuentra con el desplazamiento s en T, entonces decimos que s es un *desplazamiento válido*; de lo contrario decimos que s es un *desplazamiento no válido*.

El problema de la correspondencia de cadenas, es encontrar todos los desplazamientos válidos con el que un determinado patrón P se encuentra en un texto dado T. La figura 3 ilustra estas definiciones.

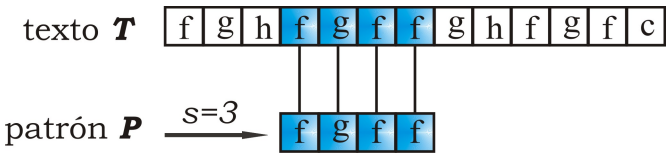


Figura 3. *El problema de correspondencia de cadenas. El objetivo es encontrar todas las apariciones del patrón $P = fgff$ en el texto $T = fghfgffghfgfc$. El patrón se encuentra sólo una vez en el texto, en la posición $s = 3$. El desplazamiento $s = 3$ se dice que es un desplazamiento válido. Cada carácter del patrón está conectado por una línea vertical con el carácter correspondiente en el texto, y todos los caracteres que coinciden se muestran sombreados de azul.*

Excepto para el algoritmo de fuerza bruta, cada algoritmo de correspondencia de cadenas, realiza cierto preprocesamiento basado en el patrón y luego encuentra todos los desplazamientos válidos; llamaremos a esta última fase: *Correspondencia*.

Algoritmo	**Tiempo de preprocesamiento**	**Tiempo de correspondencia**
Fuerza Bruta	0	$O((n-m+1)m)$
Rabin - Karp	$\Theta(m)$	$O((n-m+1)m)$
Knuth - Morris - Pratt	$\Theta(m)$	$\Theta(n)$

Cuadro 3.1: Los algoritmos de correspondenci1a de cadenas y sus tiempos de preprocesamiento y de correspondencia

El cuadro 3.1, muestra de manera comparativa los tiempos de preprocesamiento y de correspondencia de tres de los algoritmos relevantes en la correspondencia de cadenas. El tiempo total de ejecución de cada algoritmo es la suma del tiempo de preprocesamiento con el tiempo de correspondencia. El algoritmo de Rabin y Karp es un interesante algoritmo de correspondencia de cadenas, aunque el tiempo de ejecución $\Theta((n-m+1)m)$ en el peor de los casos de este algoritmo, no es mejor que el del método de fuerza bruta, pero funciona mucho mejor en promedio y en la práctica. También se generaliza muy bien a otros problemas de correspondencia de cadenas.

3.4.1. Notación y terminología

Dejaremos que Σ^* (leer "sigma-estrella") denote el conjunto de todas las cadenas de longitud finita formadas con caracteres del alfabeto Σ. En este trabajo, consideramos sólo cadenas de longitud finita. La cadena vacía de longitud cero, se denota como ε, también pertenece a Σ^*. La longitud de una cadena x se denota como $|x|$. La concatenación de dos cadenas x y y, se denota como xy, y tiene longitud $|x| + |y|$ y consiste de los caracteres de x seguidos por los caracteres de y.

Decimos que una cadena w es un *prefijo* de una cadena x, denotado $w \sqsubset x$, si $x = wy$ para alguna cadena $y \epsilon \Sigma^*$. Tenga en cuenta que si $w \sqsubset x$, entonces $|w| \leq |x|$. Del mismo modo, decimos que una cadena w es un *sufijo* de una cadena x, denotado $w \sqsupset x$, si $x = yw$ para alguna cadena $y \epsilon \Sigma^*$. Se deduce de $w \sqsupset x$ que $|w| \leq |x|$. La cadena vacía ε es tanto un sufijo y un prefijo de cada cadena. Por ejemplo, tenemos $ab \sqsubset abcca$ y $cca \sqsupset abcca$. Es útil señalar que para cualesquiera dos cadena x,y y para cualquier carácter a, tenemos que $x \sqsupset y$ si y sólo si $xa \sqsupset ya$. También tenga en cuenta que $\sqsubset$ y $\sqsupset$ son relaciones transitivas. El siguiente lema será útil más adelante.

Lema 1 (Sufijo de superposición)**.** *Supongamos que x, y y z son cadenas tales que $x \sqsupset z$ y $y \sqsupset z$. Si $|x| \leq |y|$, entonces $x \sqsupset y$. Si $|x| \geq |y|$, entonces $y \sqsupset x$. Si $|x| = |y|$, entonces x=y.*

Vea la figura 4 para una justificación gráfica.

Por brevedad de la notación, vamos a denotar el k-ésimo carácter del prefijo $P[1 \ldots k]$ del patrón $P[1 \ldots m]$ por P_k. En consecuencia, $P_0 = \varepsilon$ y $P_m = P = P[1 \ldots m]$. Del mismo modo, denotamos el k-ésimo carácter del prefijo del texto T como T_k. Usando esta notación, podemos enunciar el problema de correspondencia de cadenas como aquel que encuentra todos los desplazamientos s en el rango $0 \leq s \leq n - m$ de tal manera que $P \sqsupset T_{s+m}$.

Nuestro pseudocódigo, nos permite que dos cadenas de igual longitud sean comparadas por la igualdad como una operación primitiva. Si las cadenas se comparan de izquierda a derecha y se detiene la comparación cuando se descubre una incongruencia, asumimos que el tiempo tomado por dicha prueba es una función lineal del número de caracteres que coinciden. Para ser preciso, la prueba $x = y$, supone que debe tomar el tiempo $\Theta(t + 1)$, donde t es la longitud de la cadena más larga z de tal manera que $z \sqsubset x$ y $z \sqsubset y$.

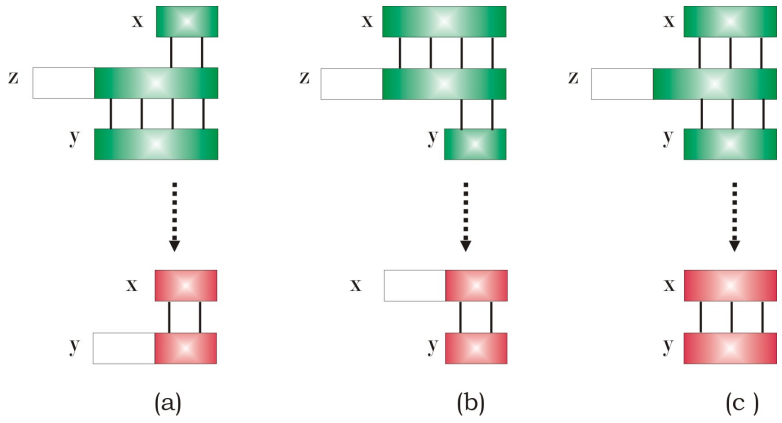


Figura 4. *Una prueba gráfica del Lemma 1. Suponemos que* $x \sqsupset z$ *y* $y \sqsupset z$*. Las tres partes de la figura ilustran los tres casos del lema. Las líneas verticales conectan regiones de correspondencia entre las cadenas(se muestran sombreadas).* (a) *Si* $|x| \leq |y|$*, entonces* $x \sqsupset y$*.* (b) *Si* $|x| \geq |y|$*, entonces* $y \sqsupset x$*.* (c) *Si* $|x| = |y|$*, entonces* $x = y$*.*

(Escribimos $\Theta(t+1)$ en lugar de $\Theta(t)$ para manejar el caso donde $t = 0$; los primeros caracteres comparados no coinciden, pero se necesita una cantidad de tiempo positiva para realizar esta comparación).

A continuación se describe un algoritmo de correspondencia de cadenas, mucho más eficiente que sus competidores mencionados en el cuadro 3.1; el algoritmo de Knuth-Morris-Pratt (o KMP). A pesar de que el algoritmo KMP tiene el mismo tiempo de preprocesamiento que el algoritmo Rabin-Karp, el algoritmo KMP reduce el tiempo de correspondencia a tan sólo $\Theta(n)$.

3.5. Algoritmo Knuth-Morris-Pratt

El algoritmo Knuth-Morris-Pratt es un algoritmo de correspondencia de cadenas de tiempo lineal gracias a sus creadores Knuth, Morris, y Pratt. Su algoritmo tiene un tiempo de correspondencia de $\Theta(n)$. Usando sólo una función auxiliar $\pi[1 \ldots m]$ previamente calculada a partir del patrón en un tiempo $\Theta(m)$.

3.5.1. La función prefijo para un patrón

La función prefijo π reduce el conocimiento acerca de como el patrón coincide con sus propios avances. Esta información puede ser usada para evitar avances de pruebas inútiles. Considerando una búsqueda por fuerza bruta, la figura 5 muestra el desplazamiento s de una plantilla que contiene el patrón $P = fgfgfhg$ contra de un texto T. En este ejemplo, se observa que se cumple la correspondencia para cinco de los caracteres, pero el sexto carácter del patrón no corresponde con el carácter correspondiente en el texto. La información de que q caracteres correspondieron satisfactoriamente determina los caracteres del texto correspondiente. Conociendo esos q caracteres del texto nos permite determinar inmediatamente que avances son inválidos. En el ejemplo de la figura 5, el avance $s+1$ es necesariamente inválido, puesto que el primer carácter del patrón (a) debería estar alineado con un carácter del texto que se sabe puede coincidir con el segundo carácter del patrón, como se muestra en el inciso (b). El avance mostrado $s' = s + 4$, en el inciso (b), alinea los primeros tres caracteres del patrón con los tres caracteres del texto que necesariamente deberían coincidir.

Dado que los caracteres del patrón $P[1 \dots q]$ coinciden con los caracteres del texto $T[s+1 \dots s+q]$

¿Cuál es el menor desplazamiento $s' > s$? tal que

$$P[1 \dots k] = T[s'+1 \dots s'+k] \tag{3.35}$$

donde $s'+k = s+q$.

Dicho avance s' es el primer avance mayor que s, que no es necesariamente inválido, debido a nuestro conocimiento de $T[s+1 \dots s+q]$. En el mejor de los casos, tenemos que $s' = s+q$, y los avances $s+1, s+2, \dots, s+q-1$ son todos inmediatamente descartados. En cualquier caso, para el nuevo avance s' no necesitamos comparar los primeros k caracteres de P con el carácter correspondiente de T, dado que hemos garantizado que ellos coinciden por la ecuación $P[1 \dots k] = T[s'+1 \dots s'+k]$. La información necesaria puede ser previamente calculada, comparando el patrón en contra de él mismo, como se ilustra en el inciso (c). Puesto que $T[s'+1 \dots s'+k]$ es parte de una porción conocida del texto, es un sufijo de la cadena P_q.

Por consiguiente, la ecuación 3.35 puede ser interpretada como una pregunta por el más grande $k < q$ tal que $P_k \sqsupset P_q$. Entonces, $s = s + (q-k)$ es

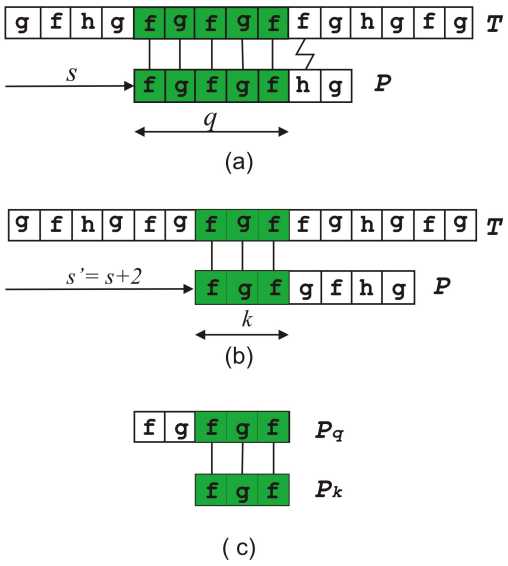


Figura 5. *La función prefijo π (a) El patrón $P = fgfgfhg$ está alineado con el texto T de manera que los primeros $q = 5$ caracteres corresponden. Los caracteres en correspondencia, se muestran de color verde, y están conectados con líneas verticales. (b) Usando solo nuestro conocimiento de los 5 caracteres en correspondencia, podemos deducir que un avance de s+1 es inválido, pero que un avance de s´=s+2 es consistente con todo lo que sabemos acerca del texto y por lo tanto es potencialmente válido. (c) La información utilizada para tales deducciones se pueden calcular previamente comparando el patrón con el mismo. Aquí vemos que la longitud del prefijo de P este es también que P_3 es un sufijo propio de P_5*

el siguiente desplazamiento potencialmente válido. Este se vacía de manera conveniente para almacenar el número k de caracteres que coinciden en el nuevo desplazamiento $s\acute{}$, en lugar de almacenar, decimos, $s\acute{} - s$.

Formalizaremos el cálculo previo requerido como sigue. Dado un patrón $P[1 \ldots m]$, la función prefijo para el patrón P, es la función π : {1,2,$\cdots$,m} $\Longrightarrow$ {0,1,$\cdots$,m-1} tal que

$$\pi[q] = max\{k : k < q \text{ y } P_k \sqsupset P_q\}$$

Esto es, $\pi[q]$ es la longitud del prefijo más largo de P, que es un sufijo propio de P_q. Como otro ejemplo, la figura 6 proporciona la función prefijo π para el patrón $fgfgfgfghf$.

El algoritmo de correspondencia Knuth-Morris-Pratt y el algoritmo para

i	1	2	3	4	5	6	7	8	9	10
P[i]	f	g	f	g	f	g	f	g	h	f
π[i]	0	0	1	2	3	4	5	6	0	1

(a)

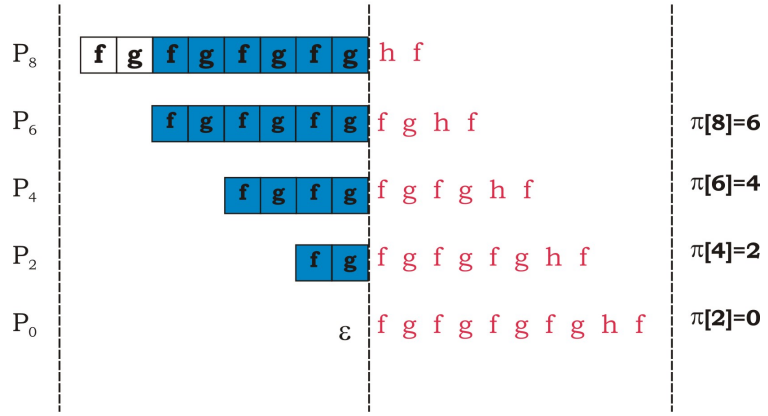


Figura 6.

el cálculo de la función prefijo se presentan abajo en lenguaje de programación Visual Basic y se les denomina como *buscador-KMP* y *función prefijo* respectivamente.

```
   FUNCIÓN PREFIJO
1  Private Function PrefijoKMP(Patron As String) As Integer()
2  Dim prefix(1 To 20) As Integer
3  Dim m As Integer
4  m = Len(Patron)
5  prefix(1) = 0
6  k = 0
7  For q = 2 To m
8    Do While ((k > 0) And (Mid(Patron, k + 1, 1) <> Mid(Patron, q, 1)))
9      k = prefix(k)
10   Loop
11   If (Mid(Patron, k + 1, 1) = Mid(Patron, q, 1)) Then
12     k = k + 1
13   End If
```

```
14   prefix(q) = k
15 Next q
16 PrefijoKMP = prefix
17 End Function


   BUSCADOR-KMP
1  Private Function KMP(Patron As String, Texto As Variant) As String
3  Dim n, m, i, q As Integer
4  Dim prefix() As Integer
2  Dim result As String
5  If (Patron = ) Then
6    MsgBox "Favor de capturar el patrón", vbCritical
7  Else
8    n = Len(Texto)
9    m = Len(Patron)
10   prefix = PrefijoKMP(Patron)
11   q = 0
12   For i = 1 To n
13     Do While ((q > 0) And (Mid(Patron, q + 1, 1) <> Mid(Texto, i, 1)))
14       q = prefix(q)
15     Loop
16     If (Mid(Patron, q + 1, 1) = Mid(Texto, i, 1)) Then
17       q = q + 1
18     End If
19     If q = m Then
20       result = result + Str(i - m) + ","
21       q = prefix(q)
22     End If
23   Next i
24 End If
25 KMP = result
26 End Function
```

3.5.2. Análisis del tiempo de ejecución

En esta sección analizaremos el tiempo de ejecución de la *función prefijo* la cual es $\Theta(m)$. Asociamos un potencial k con el k-estado actual del algoritmo. Este potencial tiene un valor inicial 0, en la línea 6. En la línea 9 decrece k cuantas veces se ejecuta, puesto que $prefix[k] < k$. Dado que $prefix[k] \geq 0$ para todo k, no obstante, k nunca puede llegar a ser negativo. La única otra línea que afecta a k es la línea 12, la cual incrementa k en al menos uno durante cada ejecución del cuerpo del ciclo for. Puesto que $k < q$ por encima de la entrada del ciclo for, y dado que q se incrementa en cada iteración del cuerpo del ciclo for, siempre se cumple que $k < q$. Adicionalmente, esto justifica la afirmación $prefix[q] < q$, en la línea 14. Podemos ser ventajosos por cada ejecución del cuerpo del ciclo *while* en la línea 9 con el correspondiente decremento en la función potencial, puesto que $prefix[k] < k$. La línea 12 incrementa la función potencial en al menos uno, así que el costo del cuerpo dentro del ciclo que comprende las líneas 13-23 es $O(1)$. Dado que el número de iteraciones fuera del ciclo es $\theta(m)$, y como la función potencial final es al menos tan grande como la función potencial inicial. Entonces, el tiempo total de ejecución actual en el peor de los casos de la *función prefijo* es $\theta(m)$.

Un análisis similar, usando el valor de q como la función potencial, muestra que el tiempo de correspondencia del Buscador-KMP es $\theta(n)$.

3.6. Interpolación de Hermite

Este método de interpolación, se debe a C. Hermite(1822-1901)[77].

Dados $n+1$ números distintos $x_0, x_1, \ldots, x_n$ en $[a, b]$ y los enteros no negativos $m_0, m_1, \ldots, m_n$, y $m = \text{máx}\{m_0, m_1, \ldots, m_n\}$. El polinomio osculante que aproxima una función $f \in C^m[a, b]$, en x_i para cada $i = 0, 1, \ldots, n$, es el polinomio de menor grado que concuerda con la función f y con todas sus derivadas de orden menor o igual que m_i en x_i para cada $i = 0, 1, \ldots, n$. El grado de este polinomio osculante es a lo más,

$$M = \sum_{i=0}^{n} m_i + n$$

ya que el número de condiciones por cumplir es $\sum_{i=0}^{n} m_i + (n+1)$, y un polinomio de grado M tiene $M+1$ coeficientes que podemos utilizar para satisfacerlas.

Definición 1. *Sean $x_0, x_1, \ldots, x_n$, n+1 números distintos en [a, b] y m_i un entero no negativo asociado a x_i para $i = 0, 1, \ldots, n$. Supóngase que $f \in C^m[a,b]$ y que $m = máx\,\{m_0, m_1, \ldots, m_n\}$. El polinomio osculante que aproxima f es el polinomio $P(x)$ de menor grado tal que*

$$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k} \text{ para cada } i = 0, 1, \ldots, n; k = 0, 1, \ldots, m_i$$

Cuando $m_i = 1$ para cada $i = 0, 1, \ldots, n$, se produce una clase de polinomios denominados ***polinomios de Hermite***.

Teorema 2. *Si $f \in C^1[a,b]$ y si $x_0, x_1, \ldots, x_n \in [a,b]$ son distintos, el polinomio único de menor grado que concuerda con f y f' en $x_0, x_1, \ldots, x_n$ es el polinomio de Hermite de grado a lo más $2n+1$ que está dado por*

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \widehat{H}_{n,j}(x)$$

donde

$$H_{n,j}(x) = [1 - 2(x - x_j) L'_{n,j}(x_j)] L^2_{n,j}(x)$$

y

$$\widehat{H}_{n,j}(x) = (x - x_j) L^2_{n,j}(x)$$

Dentro de este contexto $L_{n,j}(x)$ denota el j-ésimo polinomio de Lagrange de grado n.

Más aún, si $f \in C^{2n+2}[a,b]$ entonces para $x \in [a,b]$

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n+2)!} f^{2n+2}(\xi)$$

para algún ξ con $a < \xi < b$.

Demostración: Se tiene que

$$L_{n,j}(x_i) = \begin{cases} 0, & \text{si } i \neq j \\ 1, & \text{si } i = j \end{cases} \tag{3.36}$$

Por lo tanto, cuando $i \neq j$

$$H_{n,j}(x_i) = 0 \text{ y } \widehat{H}_{n,j}(x_i) = 0$$

mientras que

$$H_{n,i}(x_i) = [1 - 2(x_i - x_i)L'_{n,i}(x_i)]1 = 1$$

y

$$\widehat{H}_{n,i}(x_i) = (x_i - x_i) \cdot 1^2 = 0$$

En consecuencia,

$$H_{2n+1}(x_i) = \sum_{j=0}^{n} f(x_j) \cdot 0 + f(x_i) \cdot 1 + \sum_{j=0}^{n} f'(x_j) \cdot 0 = f(x_i)$$

así que H_{2n+1} concuerda con f en $x_0, x_1, \ldots, x_n$.

Si queremos demostrar la concordancia de H'_{2n+1} con f' en los nodos, primero observamos que $L_{n,j}(x)$ es un factor de $H'_{n,j}(x)$ de modo que $H'_{n,j}(x_i) = 0$ cuando $i \neq j$. Además, si $i = j$ y $L_{n,i}(x_i) = 1$, tenemos,

$$\begin{aligned} H'_{n,i}(x_i) &= -2L'_{n,i}(x_i) \cdot L^2_{n,i}(x_i) + [1 - 2(x_i - x_i)L'_{n,i}(x_i)]2L_{n,i}(x_i)L'_{n,i}(x_i) \\ &= -2L'_{n,i}(x_i) + 2L'_{n,i}(x_i) = 0 \end{aligned}$$

Por lo tanto $H'_{n,j}(x_i) = 0 \ \forall i, j$

Finalmente

$$\begin{aligned} \widehat{H}'_{n,j}(x_i) &= L^2_{n,j}(x_i) + (x_i - x_j)2L_{n,j}(x_i)L'_{n,j}(x_i) \\ &= L_{n,j}(x_i)[L_{n,j}(x_i) + 2(x_i - x_j)L'_{n,j}(x_i)] \end{aligned}$$

así que $\widehat{H}'_{n,j}(x_i) = 0$ si $i \neq j$ y $\widehat{H}'_{n,i}(x_i) = 1$, al combinar estos hechos tenemos

$$H'_{2n+1}(x_i) = \sum_{j=0}^{n} f(x_j) \cdot 0 + \sum_{j=0}^{n} f'(x_j) \cdot 0 + f'(x_i) \cdot 1 = f'(x_i)$$

Por lo tanto, H_{2n+1} concuerda con f y H'_{2n+1} con f' en $x_0, x_1, \ldots, x_n$ $\square$

Capítulo 4

Modelo propuesto

En este capítulo se describe el algoritmo propuesto, tema central en el presente trabajo de tesis. En la primera sección del capítulo, se explican las modificaciones realizadas al algoritmo de correspondencia de cadenas, mismo que se utilizará como parte fundamental en la construcción de un algoritmo para la predicción de contaminantes atmosféricos. Posteriormente, en la segunda sección del capítulo, se describe el algoritmo para la predicción de contaminantes atmosféricos.

4.1. Algoritmo Knuth-Morris-Pratt modificado

La principal tarea del algoritmo KMP, es la búsqueda de correspondencia entre cadenas de caracteres de manera eficiente. Utilizando esta ventaja de realizar búsquedas de correspondencia de cadenas en textos lo suficientemente grandes de manera eficiente, es lo que nos lleva a utilizar el algoritmo de KMP como parte fundamental para el desarrollo de un algoritmo que nos permita generar predicciones confiables que se puedan comparar con métodos tradicionales dentro del campo de la estadística.

En primer lugar, es necesaria una modificación a la entrada de datos, ya que en principio el algoritmo KMP realiza la correspondencia de cadenas carácter a carácter. Pero en nuestro caso se tienen valores reales con 3 y 5 caracteres de longitud; una parte entera, un punto y una parte decimal.

Dependiendo del contaminante del que se trate, la parte entera constará solamente de un dígito y la parte decimal puede estar compuesta por 1 dígito o bien por 3 dígitos. Este problema se resuelve de la siguiente manera, si cada valor de la serie tiene una longitud igual a 3, entonces se aplicarán desplazamientos de 4 caracteres por cada valor de entrada, pero si cada valor de la serie tiene una longitud igual a 5, entonces se aplicarán desplazamientos de 6 caracteres por cada valor de entrada. Más aún, de manera generalizada, si cada valor en la serie consta de k-caracteres entonces se consideran desplazamientos de k+1 caracteres por cada valor de entrada.

A continuación se presenta el algoritmo original KMP en seudocódigo.

```
   BUSCADOR-KMP
1  n ← length[T]
2  m ← length[P]
3  π ← Compute-Prefix Function(P)
4  q ← 0
5  for i ← 1 to n
6    do while q>0 and P[q+1]≠ T[i]
7       do q ← π[q]
8    if P[q+1]=T[i]
9     then q ← q+1
10   if q=m
11    then print: Pattern occurs with shift i-m
12      q ← π[q]
```

El siguiente cambio importante dentro del algoritmo KMP se da a partir de la línea 8, donde claramente se busca la correspondencia exacta entre cadenas de caracteres. Sin embargo, en nuestro caso no se busca una correspondencia exacta, puesto que habrá casos donde ni siquiera exista la correspondencia. Por tal motivo, se buscará obtener una aproximación entre una secuencia de números pequeña dentro de una secuencia de números más larga, que en términos de la correspondencia de cadenas quedaría como una aproximación entre el patrón P y el texto T, a la cual denominaremos como *correspondencia aproximada*. A partir de este momento se utilizarán las notaciones correspondientes al capítulo 3, con relación a la correspondencia de cadenas. Debemos mencionar que la correspondencia aproximada estará en

función del valor δ, pero es en la siguiente sección donde se describe el valor δ.

Una cuestión que se debe considerar antes de buscar la correspondencia aproximada es saber si los valores de las series se encuentran al mismo nivel o escala. En caso de no estar al mismo nivel o escala surge un problema importante por lo cual se debe realizar una normalización en los datos al momento de llevar a cabo la correspondencia. Para resolver este problema se proponen dos métodos:

1. Calcular las diferencias $\Delta P_t = P[t+1] - P[t]$ y $\Delta T_t = T[s+t+1] - T[s+t]$ para $t = 1 \ldots n-1$ y posteriormente calcular la diferencia $\Delta Error_t = \Delta P_t - \Delta T_t$ para $t = 1 \ldots n-1$

2. Sacar la medias, tanto del patrón como del texto y posteriormente sumar las medias con cada uno de los valores de la serie no correspondientes y por último calcular la diferencia $\Delta Error_t = T[s+t] - P[t]$

El valor s dentro del algoritmo KMP, indica el desplazamiento del patrón sobre un determinado texto.

Una vez que se ha elegido el método de normalización para los datos, se procede a calcular el error cuadrático medio (ECM) o bien la raíz del error cuadrático medio (RECM) dependiendo de la precisión deseada.

$$ECM = \frac{1}{n} \sum_{t=1}^{n} (Error_t)^2 \tag{4.1}$$

$$RECM = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (Error_t)^2} \tag{4.2}$$

Una vez que se ha calculado el RECM o bien el ECM se consideran dos posibles opciones:

- Autoajustar el valor de umbral δ; según sea el caso (RECM o ECM), cada vez que uno de estos valores sea menor que δ, se le asignará este nuevo valor a δ, de tal manera que δ se hará cada vez más pequeño, y esto dará como resultado un menor número en las correspondencias aproximadas. A esta modalidad se le denomina como *Auto*

- Dejar fijo el valor de umbral δ, lo cual dará como resultado un mayor número de correspondencias aproximadas. A esta modalidad se le denomina como *Manual*

Después de establecer la modalidad deseada, se verifica si se cumple una de las siguientes desigualdades $ECM < \delta$ o bien $RECM < \delta$ dependiendo de cuál valor se haya calculado. En caso de cumplirse la desigualdad, se incrementa un contador que nos indica el número total de correspondencias aproximadas y se procede a busca el valor $T[s + n + 1]$ para calcular la diferencia entre $T[s+n+1]$ y $T[s+n]$. El resultado de la diferencia se asigna a una variable de suma.

$$\Delta Suma = T[s + n + 1] - T[s + n] \tag{4.3}$$

ya que en caso de existir más de una correspondencia aproximada se divide la variable $\Delta Suma$ entre el número total de correspondencias encontradas(contador) y el resultado se suma con P[n], generando así el valor de predicción para P.

4.2. Método de predicción por aproximación de cadenas (MPAC)

Es importante mencionar que en cada proceso de predicción son necesarias dos series de tiempo: una con valores en el pasado a la cual llamaremos *serie 1* y otra con valores actuales, a la cual llamaremos *serie 2* donde tales valores se consideran actuales con relación a los valores de la serie 1. Las predicciones correspondientes se realizarán con base en la serie 2. Cabe mencionar que la serie 1 y 2 pueden ser de diferente tamaño.

Una vez que se han seleccionado las series, se procede de acuerdo con el siguiente método (algoritmo):

1. Se asigna el tamaño de la ventana, el cual definiremos como n.
2. Se introduce la serie 1 la cual simulará el *texto* dentro del algoritmo KMP.
3. Se introduce el *patrón* con n-valores extraídos de la serie 2.

4. Se asigna un valor de umbral al cual denominaremos como valor δ.

5. Se realiza una búsqueda de correspondencia de cadenas entre el patrón y el texto. Esto se realiza con ayuda del algoritmo modificado de *Knuth-Morris-Pratt*, el cual nos proporciona como valor de salida la predicción.

6. Si se desean obtener predicciones adicionales, regresamos al paso 1 incorporando el dato obtenido en la posición n dentro del patrón de dimensión $n - 1$.

Se debe mencionar que el texto y el patrón no deben tener el mismo tamaño. En principio el texto debe tener mayor tamaño para considerar la búsqueda de una o varias correspondencias del patrón dentro del texto.

En el punto 4 nos referimos a este valor como un valor de umbral el cual nos permite considerar la mínima cantidad de correspondencias posibles para generar el valor de predicción. En el paso 6, cuando nos referimos al patrón de dimensión $n - 1$, en realidad nos referimos al patrón original pero sin incorporar el primer valor. Sin embargo, no necesariamente se requiere que el patrón sea la serie más actual, ya que en el caso de querer comparar los valores en el pasado de una serie con los valores de predicción generados por MPAC, basta con solo introducir dicha serie y realizar un desplazamiento hacia adelante en el patrón sin tener reconstruir el nuevo patrón.

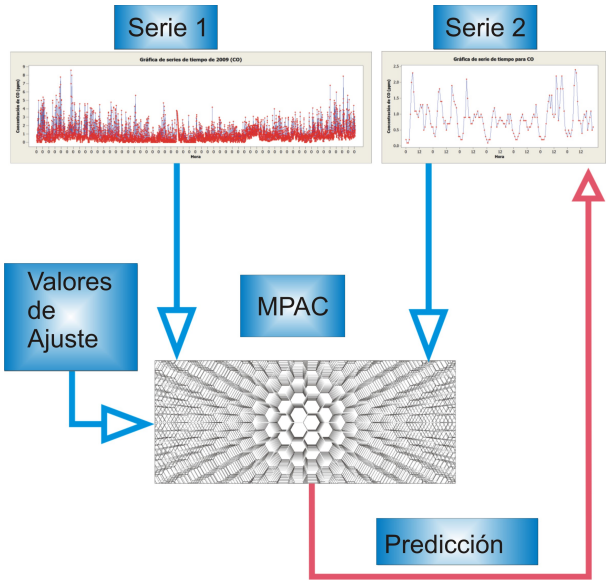


Figura 7. *Diagrama del método MPAC para la predicción de contaminantes atmosféricos.*

Capítulo 5

Resultados y Discusión

Este capítulo es de vital importancia en el presente trabajo de tesis, puesto que no sólo se ilustran los conceptos descritos en los capítulos anteriores, sino que también se expone la eficacia del método propuesto para la predicción de contaminantes atmosféricos; asimismo, se muestran los resultados de la implementación del método en una aplicación de cómputo.

Los experimentos se realizaron en una computadora portátil, con un procesador Intel(R) Core(TM) i3 a 2.40 GHz, 4.0 GB de memoria RAM y 281 GB de espacio en disco duro, con sistema operativo Windows 7 Home Premium de Microsoft. También se utilizaron las herramientas computacionales Minitab 15, Matlab 2010 y Visual Basic 6.0.

Los bancos de datos (bases de datos) se tomaron del Sistema de Monitoreo Atmosférico de la Ciudad de México (SIMAT)[78]; en específico de la Red Automática de Monitoreo Atmosférico (RAMA), la cual cuenta con 34 estaciones de monitoreo ubicadas en lugares estratégicos de la Ciudad de México; 21 están localizadas en el Distrito Federal y 13 en el Estado de México. Las estaciones que hoy integran a la RAMA, no se sabe con exactitud por cuánto tiempo estarán vigentes debido a que la RAMA es un sistema dinámico, que se adapta a las circunstancias de la calidad del aire.

Dentro los contaminantes atmosféricos que se registran en la mayoría de las estaciones de la RAMA, y que se utilizarán en este trabajo de tesis, están contemplados dos de ellos, mismos que se presentan en el cuadro 5.1.

Las bancos de datos de la (RAMA) contienen la información registrada

Contaminante(símbolo)
Monóxido de carbono(CO)
Ozono(O_3)

Cuadro 5.1: Contaminantes atmosféricos

de los niveles de concentración de ciertos contaminantes que se registran cada hora, generando así 24 registros por día y 8760 por año (no bisiesto) o 8784 (año bisiesto).

5.1. Resultados obtenidos por el método MPAC

Como se mencionó en el capítulo 4, existe un valor atípico de -99.9 que sirve para indicar si algún sensor dentro de la RAMA falló para una determinada estación de monitoreo. Este es un problema que se resuelve utilizando las facilidades que brinda MATLAB.

Otro aspecto que se debe remarcar es la forma en que se seleccionan las series de tiempo de cada una de las bases de datos. Debido a que no todas las estaciones de monitoreo presentan el mismo número de fallas, es importante seleccionar a las estaciones que registren el menor número de valores atípicos y posteriormente verificar que para un período de tiempo posterior sigan preservando dicha cualidad para fines de una comparación justa.

A continuación se analizan y se presentan los resultados de la predicción para cada uno de los contaminates atmosféricos presentados en el cuadro 5.1 utilizando el método MPAC, y posteriormente se analizarán los resultados obtenidos por medio de la metodología estadística ARIMA.

5.1.1. Resultados de predicción para el CO

En la figura 8 se muestra la representación gráfica de la serie de tiempo para la estación de monitoreo Xalostoc (XAL) para el año 2009. Dicha serie de tiempo servirá como serie 1 de acuerdo con el método MPAC. Posteriormente en la figura 9 se muestra la representación gráfica de la serie de tiempo para la estación de monitoreo Xalostoc (XAL) para el año 2010. Donde dicha serie

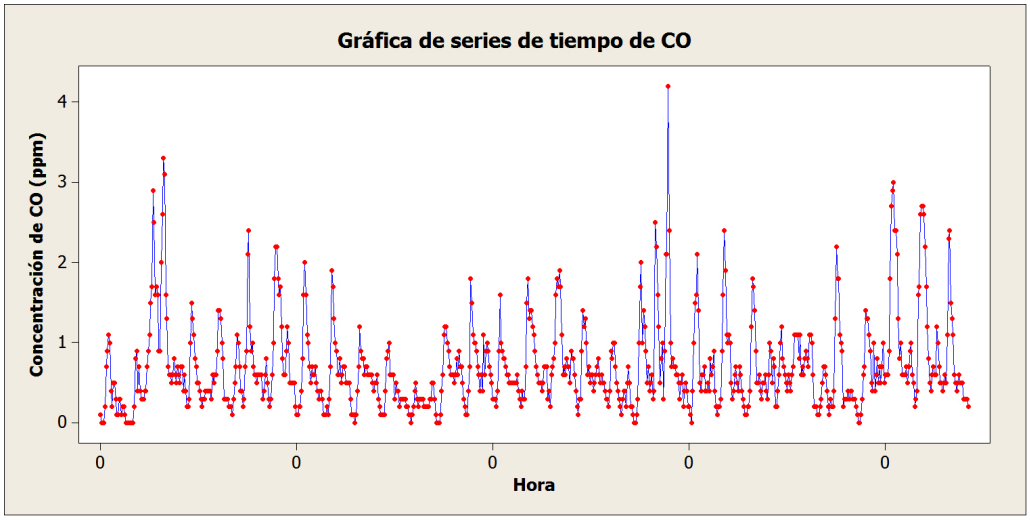


Figura 8. *Gráfica con los niveles de CO del mes de Julio del año 2009.*

servirá como serie 2 de acuerdo con el método de predicción MPAC. Una vez que hemos seleccionado las series 1 y 2 las cuales tomarán el papel de texto y de patrón respectivamente, se eligen varios tamaños de ventanas con la finalidad de obtener nuevos resultados y así poder compararlos. Dicha comparación nos permite conocer bajo qué cambios en el tamaño de la ventana los resultados pueden ser peores o mejores considerando el ECM.

Dentro del sistema de predicción se agrega un variable, denominada valor de ajuste. Este valor de ajuste nos permite limitar el número de correspondencias aproximadas que serán consideradas para obtener el valor de predicción de manera más controlada. Ya que al realizar ciertos experimentos se pudo apreciar que el número de correspondencias variaba drásticamente para cada de los diferentes patrones; por lo cual era necesario encontrar la forma de regular el número de opciones de manera equitativa.

Considerados los valores de cambio en la configuración del método MPAC se procedió a realizar una serie de pruebas bajo diferentes esquemas de configuración los cuales se enlistan en el cuadro 5.2. De las pruebas se obtienen diferentes resultados, los cuales se tienen que comparar y decidir bajo que criterio es mejor uno de otro. Como criterio de evaluación para medir la exactitud en las predicciones generadas por el método MPAC se propone el valor ECM.

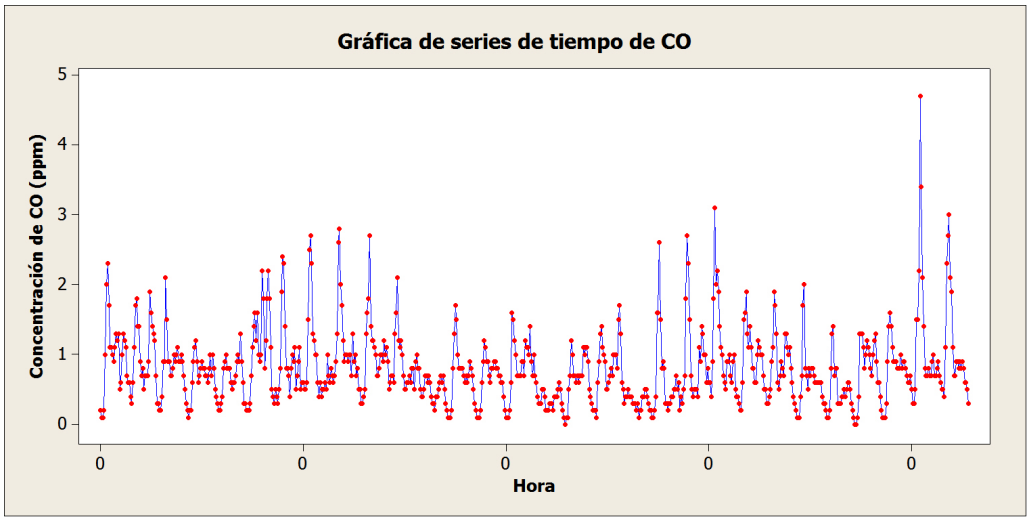


Figura 9. *Gráfica con los niveles de CO del mes de septiembre del año 2010.*

Valor δ	Tam. ventana	Val. ajuste	Modalidad	ECM
0.05	10	5	Auto	0.1281
0.1	10	5	Auto	0.1237
0.2	10	5	Auto	0.1338
0.3	10	5	Auto	0.1280
0.05	10	3	Auto	0.1297
0.1	10	3	Auto	0.1624
0.2	10	3	Auto	0.1448
0.3	10	3	Auto	0.1442
0.6	10	18	Auto	**0.1123**

Cuadro 5.2: Tabla de resultados con diferentes esquemas de configuración para el método MPAC

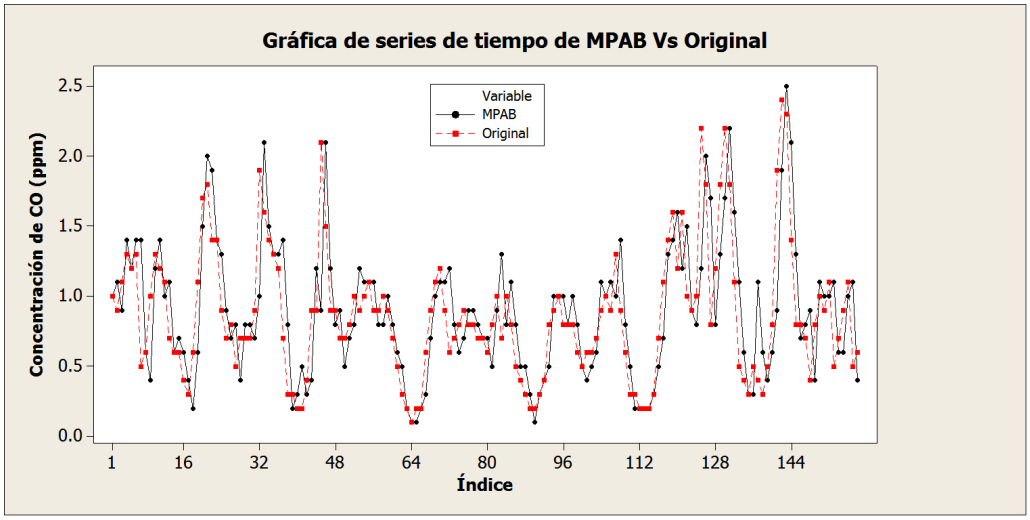


Figura 10. *Gráfica comparativa entre los datos de predicción de MPAC y los datos originales.*

Analizando el valor del ECM en el cuadro 5.2, podemos observar que la última configuración para el método MPAC, es la que nos permite obtener predicciones con un error cuadrático medio mucho menor.

En la figura 10 se muestran los valores de predicción generados por el método MPAC y los valores de la serie original. Los valores de predicción generados por el método MPAC son el resultado de la configuración con el menor ECM del cuadro 5.2. Es importante mencionar que para fines prácticos y de apreciación solo se presentan los datos correspondientes a la primera semana del mes de septiembre de 2010, donde se puede apreciar que los valores de predicción son muy aproximados a los reales y que no existen picos que sobrepasen por mucho a los valores reales.

5.2. Resultados con la metodología ARIMA

En esta sección del capítulo utilizaremos la metodología de Box-Jenkins para la identificación tentativa de un modelo apropiado que nos permita pronosticar valores futuros de las series de tiempo; además utilizaremos la diferenciación para transformar una serie de tiempo no estacionaria en una serie de tiempo estacionaria.

También veremos cómo caracterizar el comportamiento de la función de autocorrelación simple (FAC) y de la función de autocorrelación parcial (FACP) para decidir si una serie de tiempo es no estacionaria o estacionaria.

Posteriormente utilizaremos la función de autocorrelación simple y la función de autocorrelación parcial para identificar tentativamente un modelo apropiado y así pronosticar valores futuros de las serie de tiempo. Por último, se desea comparar los resultados obtenidos de la metodología ARIMA con los obtenidos por el método MPAC.

5.2.1. Resultados mediante ARIMA para CO

Es importante tener en cuenta que la aplicación de los modelos ARIMA se debe realizar sobre series estacionarias; por tal motivo es necesario analizar las cualidades de nuestra serie de tiempo. En la figura 11 se presenta la serie de tiempo original, la cual muestra un comportamiento estacional. Además se puede observar que dicha serie no presenta un tendencia clara y es difícil decidir si es estacionaria o no estacionaria.

Debido a que no podemos decidir de manera fácil si dicha serie es estacionaria o no estacionaria, revisaremos la función de autocorrelación y la función de autocorrelación parcial de la serie de tiempo en el nivel no estacional, es decir sin aplicar desfasamiento. En la figura 12(a) se observa que la FAC muestra un decrecimiento sinusoidal amortiguado lento, lo cual nos lleva a considerar que los valores de la serie de tiempo son no estacionarios, mientras que en la figura 12(b) la FACP se muestra estadísticamente grande para los desfasamientos 1 y 2, de acuerdo con los límites correspondientes al intervalo de confianza.

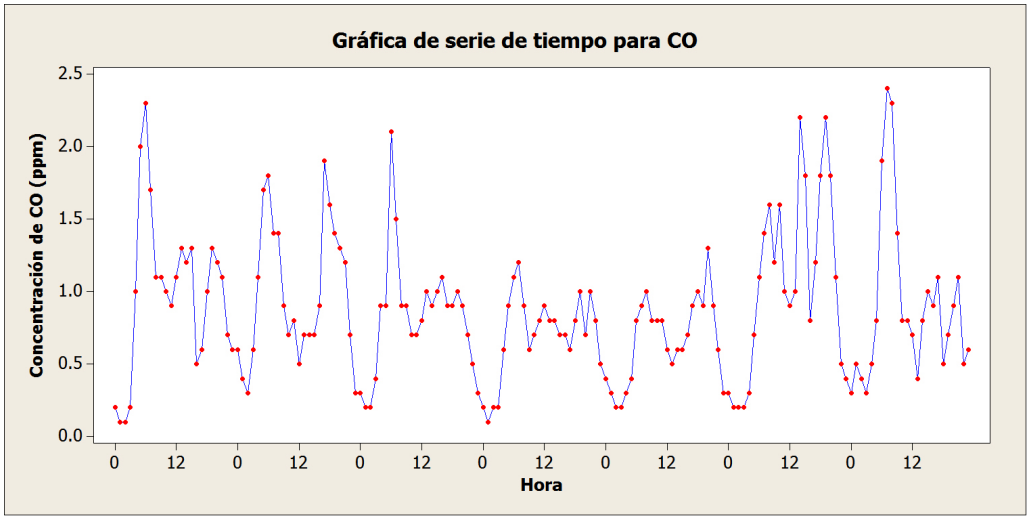
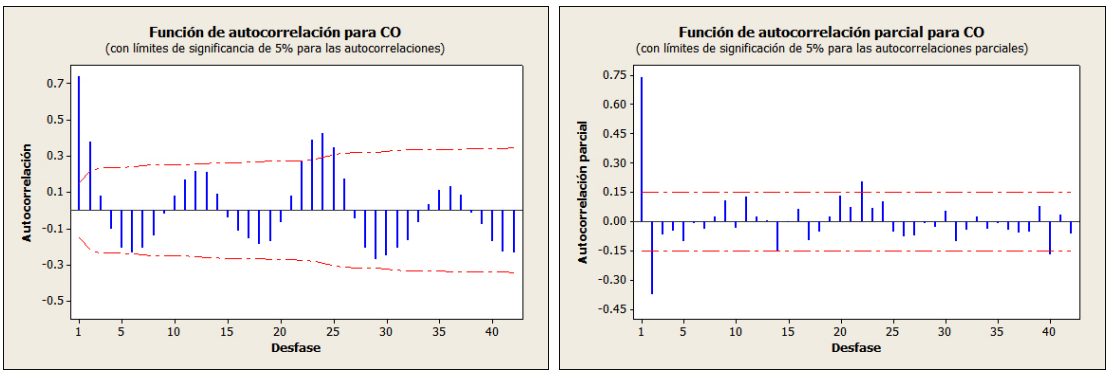


Figura 11. *Gráfica de la serie de tiempo con los niveles de CO en el año 2010.*



(a) Función de autocorrelación. (b) Función de autocorrelación parcial.

Figura 12. *Gráfica de las funciones FAC y FACP para la serie de tiempo de CO.*

Para convertir la serie de tiempo en estacionaria es necesario aplicar la primera diferenciación obteniendo como resultado una nueva serie que se muestra en la figura 13, donde se puede apreciar que las propiedades estadísticas de la serie de tiempo son prácticamente constantes a través del tiempo. Es decir, los valores de la serie fluctúan respecto de una media constante con variación constante, aunque es muy frecuente que la gráfica de las observaciones muestre una variabilidad que no es constante a lo largo del tiempo.

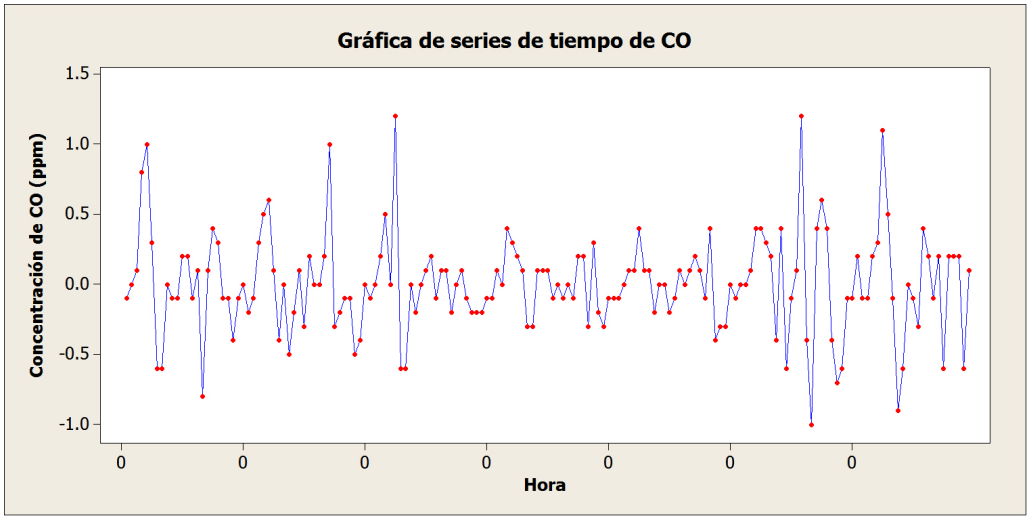
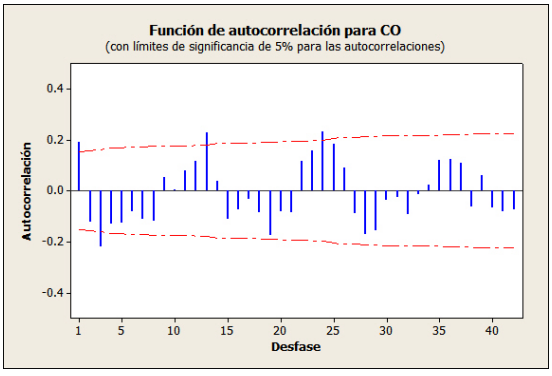
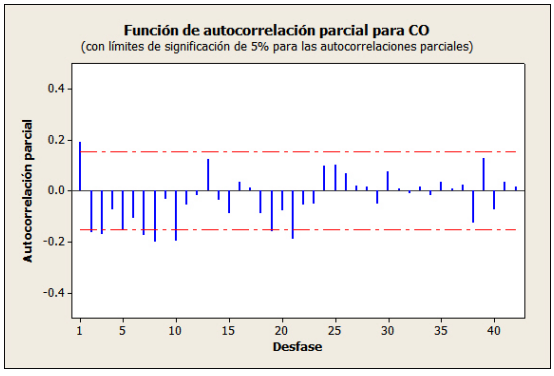


Figura 13. *Diferenciación de la serie de tiempo con los niveles de CO.*

Una vez que se ha conseguido una serie estacionaria, la identificación del orden del modelo se decidirá comparando las funciones FAC y FACP muestrales que se presentan en la figura 14, considerando para este caso un modelo AR (Autorregresivo) ya que se interpreta que la FAC presenta un decrecimiento sinusoidal amortiguado y la FACP presenta entre 2 o 3 valores significativos; es decir, 2 o 3 valores fuera del intervalo de confianza. Para este modelo concreto, el orden de la parte AR se propone de acuerdo con los retardos significativos de la FACP; es decir 2 o 3.



(a) Función de autocorrelación.



(b) Función de autocorrelación parcial.

Figura 14. *Gráfica de la serie de tiempo con los niveles de CO en el año 2010.*

Al querer expresar este modelo en la metodología ARIMA(p, d, q), donde d es el número de diferencias necesarias, que en este caso es 1 y como tentativamente se ha optado por un modelo AR, esto significa que $p = 2$ o $p = 3$. Para resolver qué valor es el adecuado, debemos realizar la estimación y considerar los estadísticos.

Para la estimación de los parámetros, utilizaremos las bondades de la aplicación MINITAB, donde los resultados de la estimación se presentan en la figura 15.

Figura 15.

Estimados finales de los parámetros

Tipo		Coef	Coef. de EE	T	P
AR	1	0.1960	0.0773	2.53	0.012
AR	2	-0.1277	0.0791	-1.62	0.108
AR	3	-0.1737	0.0782	-2.22	0.028
Constante		0.00344	0.02632	0.13	0.896

Diferenciación: 1 Diferencia regular
Número de observaciones: Serie original 168,
Residuos: SC = 18.8597 después de diferenci
MC = 0.1157 GL = 163

Estadística chi-cuadrada modificada de Box-Pierc (Ljung-Box)

Desfase	12	24	36	48
Chi-cuadrada	19.0	53.6	75.2	98.1
GL	8	20	32	44
Valor P	0.015	0.000	0.000	0.000

(a) Resultados del modelo ARIMA(3,1,0).

Estimados finales de los parámetros

Tipo		Coef	Coef. de EE	T	P
AR	1	0.2253	0.0771	2.92	0.004
AR	2	-0.1691	0.0778	-2.17	0.031
Constante		0.00287	0.02664	0.11	0.914

Diferenciación: 1 Diferencia regular
Número de observaciones: Serie original 168,
Residuos: SC = 19.4313 después de diferenciar 167
MC = 0.1185 GL = 164

Estadística chi-cuadrada modificada de Box-Pierce (Ljung-Box)

Desfase	12	24	36	48
Chi-cuadrada	19.6	53.7	73.1	95.1
GL	9	21	33	45
Valor P	0.021	0.000	0.000	0.000

(b) Resultados del modelo ARIMA(2,1,0).

Al comparar la significancia estadística de los coeficientes de la tabla de resultados mostrados en el inciso (a), podemos ver que el valor P para el segundo coeficiente no está por debajo del nivel de significancia, lo cual indica que no es estadísticamente significativo. En consecuencia, se puede concluir que no es necesario un tercer coeficiente para el modelo y por lo tanto se tiene que p=2.

Observando la tabla de resultados mostrados en la figura 15 inciso (b), se puede apreciar que ambos coeficientes son estadísticamente significativos, debido a que el valor de P, para ambos casos está por debajo del nivel de significancia. Además, el valor T cumple con la condición de ser mayor a 2 en

valor absoluto. Lo cual nos lleva a considerar que el modelo ARIMA(2,1,0) es un modelo adecuado.

5.2.2. Comparación de los resultados de predicción

Una vez que se establece tentativamente el modelo, procedemos a realizar las predicciones utilizando las bondades de Minitab. En esta etapa, es importante mencionar que Minitab nos permite elegir si se desea ajustar a un modelo estacional o no. Por lo cual, para fines experimentales se trabajan ambos modelos y después se analizan los resultados para elegir el más adecuado.

Los resultados de estos experimentos se presentan en la tabla 5.4. Podemos observar, que de los dos modelos ARIMA el que mejor se ajusta y el que tiene un menor valor ECM, es el modelo no estacional. Sin embargo, nuestro método (MPAC) es el que tiene el menor ECM de los tres.

Método	ECM
ARIMA(Estacional)	0.1637
ARIMA(No Estacional)	0.1183
MPAC	0.1129

Cuadro 5.3: Tabla con el ECM para valores ajustados

Dado que el modelo ARIMA no estacional presenta un menor valor ECM, consideramos relevante presentar en la figura 16, los valores ajustados de dicho modelo, así como los datos de la serie de tiempo original.

Posteriormente por medio de Minitab se obtienen 48 datos de predicción para cada uno de los modelos ARIMA. Estos resultados se presentan en la figura 17, donde podemos observar que los valores de predicción muestran un comportamiento muy diferente para cada uno de los diferentes modelos. Mientras que para el modelo ARIMA no estacional los valores de predicción se comportan con una tendencia casi constante, para el modelo ARIMA estacional los valores de predicción presentan un comportamiento aproximado al mostrado por los valores de la serie de tiempo original. En esta misma figura, también se incluyen los valores de predicción obtenidos con nuestro método MPAC.

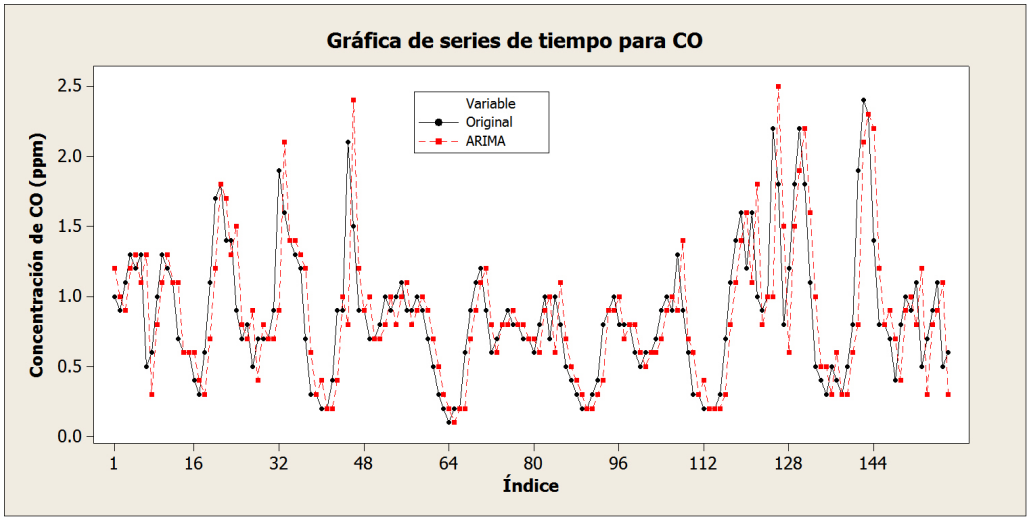


Figura 16. *Gráfica con los valores ajustados por el modelo ARIMA no estacional.*

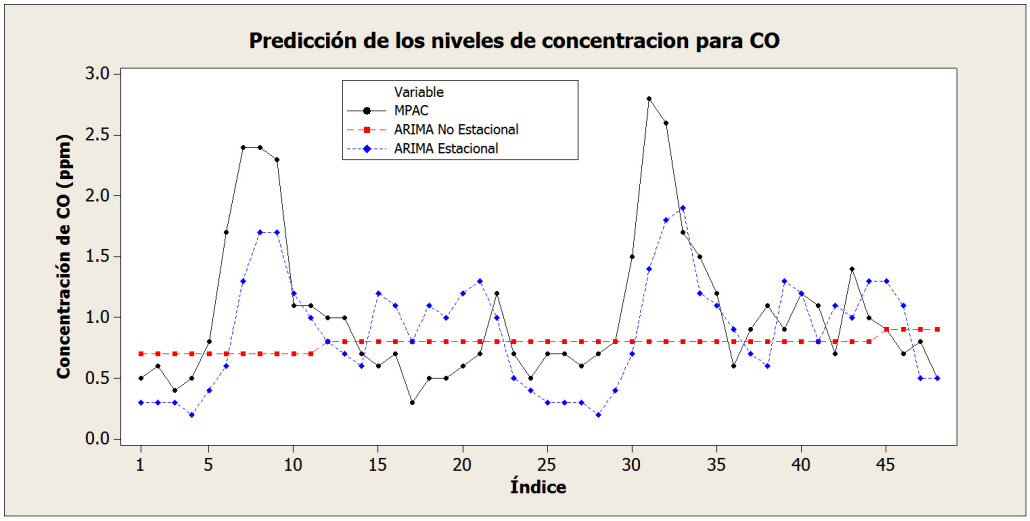


Figura 17. *Valores de predicción obtenidos por ARIMA (estacional-no estacional) y MPAC.*

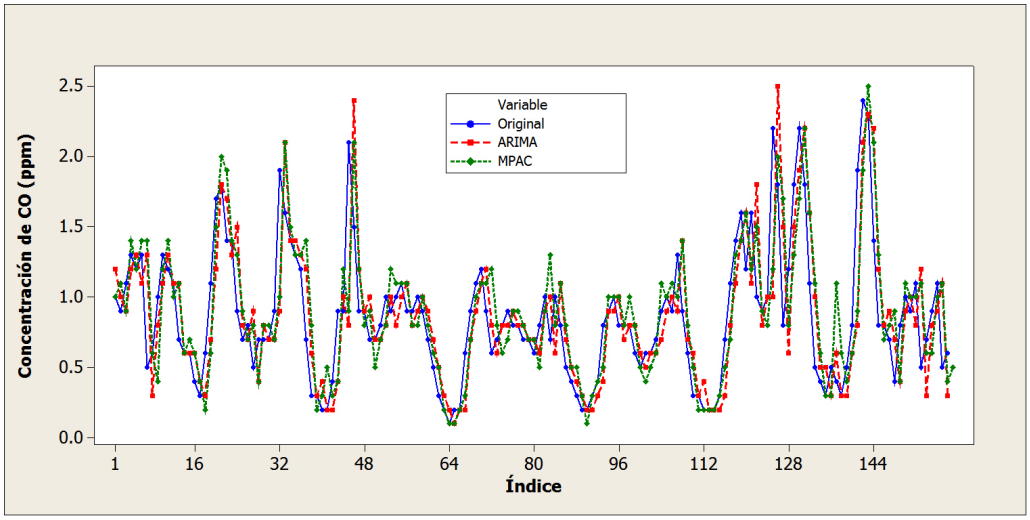


Figura 18. *Gráfica comparativa con los valores de predicción para CO.*

Por último, en la figura 18 se comparan gráficamente los datos de predicción obtenidos con nuestro método y los datos de ajuste correspondientes a un modelo ARIMA no estacional con los datos de la serie de tiempo original.

Una vez que analizamos los casos estacional y no estacional de un modelo ARIMA, no tenemos la menor duda de que los resultados son favorables para nuestro método de predicción. Pues no solo se obtiene un menor valor en el error cuadrático medio para valores en el pasados, sino que también genera buenos resultados con relación a los valores futuros. Para confirmar esta última parte, mostraremos la siguiente tabla la cual demuestra claramente que las predicciones generadas por parte de nuestro modelo MPAC obtienen un menor valor de error.

Método	ECM
ARIMA(No Estacional)	0.4773
ARIMA(Estacional)	0.3821
MPAC	0.1325

Cuadro 5.4: Tabla con el ECM para valores futuros

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones derivadas de los resultados obtenidos en el proceso de este trabajo de tesis; además, se proponen algunos de los trabajos que se podrían realizar con objeto de continuar con las ideas propuestas en este trabajo de tesis, dando la oportunidad a futuros investigadores de tratar los puntos no cubiertos, que en un futuro pudieran ser tratados en distintos trabajos de investigación.

6.1. Conclusiones

1. Se introduce la importancia de un método para la predicción de contaminantes atmosféricos con base en la correspondencia de cadenas.

2. Se define una nueva utilidad para los algoritmos de correspondencia de cadenas.

3. Por su naturaleza el algoritmo propuesto toma poco tiempo de ejecución y no requiere de significativos recursos computacionales.

4. El algoritmo propuesto no tiene problemas para generar predicciones a corto o largo plazo.

5. El desempeño mostrado por el modelo propuesto, con base en los experimentos donde se utilizan bancos de datos de SIMAT, es claramente superior al alcanzado por la metodología ARIMA.

6. No requiere de una cantidad relativamente grande de datos para poder llegar a predicciones válidas.

7. A diferencia de otros métodos convencionales no requiere de una fase de aprendizaje.

8. El algoritmo propuesto cuenta con un sistema sencillo para incorporar nuevos datos, sin tener que reajustarse, como en el caso de los modelos Box-Jenkins.

6.2. Trabajo futuro

1. Probar de manera complementaria con otras bases de datos, y comparar los resultados con los ya obtenidos.

2. Probar el método propuesto con algún otro algoritmo de correspondencia de cadenas.

3. Comparar los resultados de predicción del método propuesto con una red neuronal.

4. Realizar cambios a los criterios de correspondencia aproximada.

5. Tratar de minimizar el ECM de manera iterativa.

Apéndice A

Simbología

- μ Media
- x_t Valor de la serie en el tiempo t
- ϕ Operador autoregresivo estacionario
- φ Operador autoregresivo generalizado
- $\widetilde{Z}$ Desviación
- maxA Máximo de un conjunto A
- $\exists$ Cuantificador existencial
- $\in$ Pertenencia a un conjunto
- $\mathbb{R}^+$ Números reales positivos
- $\mathbb{N}$ Números naturales
- Σ Alfabeto
- $\forall$ Cuantificador universal
- $\subseteq$ Está contenido en
- $a \sqsubset b$ a es prefijo de b
- $b \sqsupset c$ b es sufijo de c

Apéndice B

Métodos de Interpolación en MATLAB

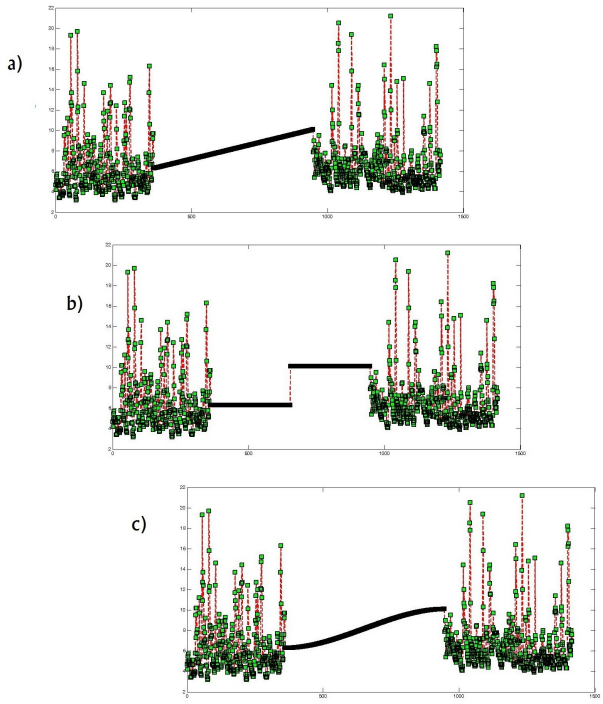


Figura B.1: Métodos de interpolación: a) Lineal b) Vecino más cercano c) Pchip

Gráficas de los datos de entrada.

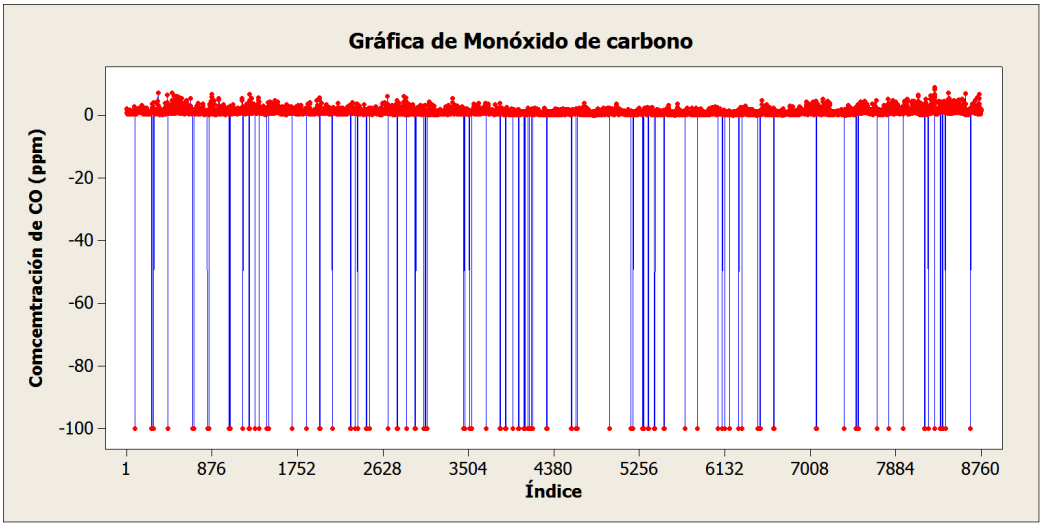


Figura B.2: Datos originales de (CO) en la estación Xalostoc (2010)

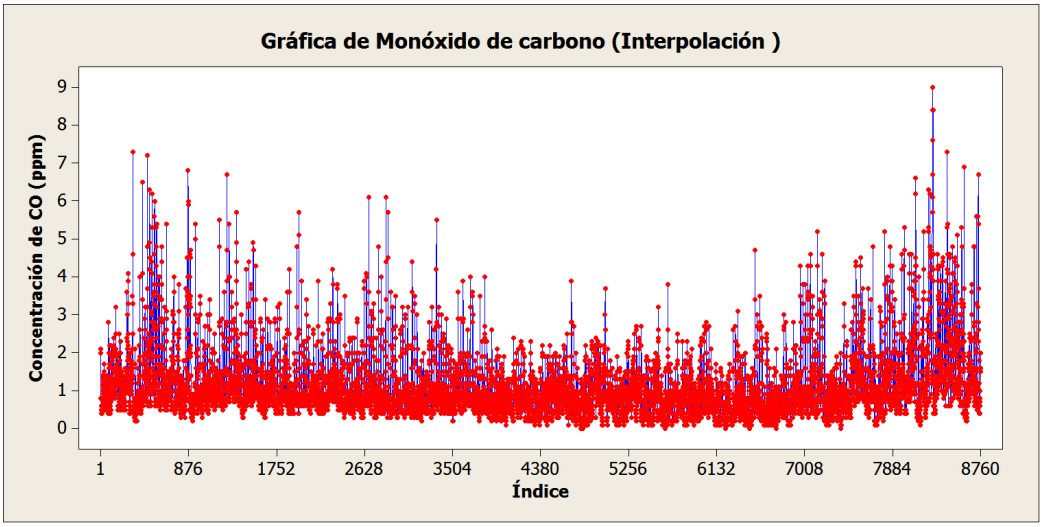


Figura B.3: Datos de (CO) con interpolación (pchip) (2010)

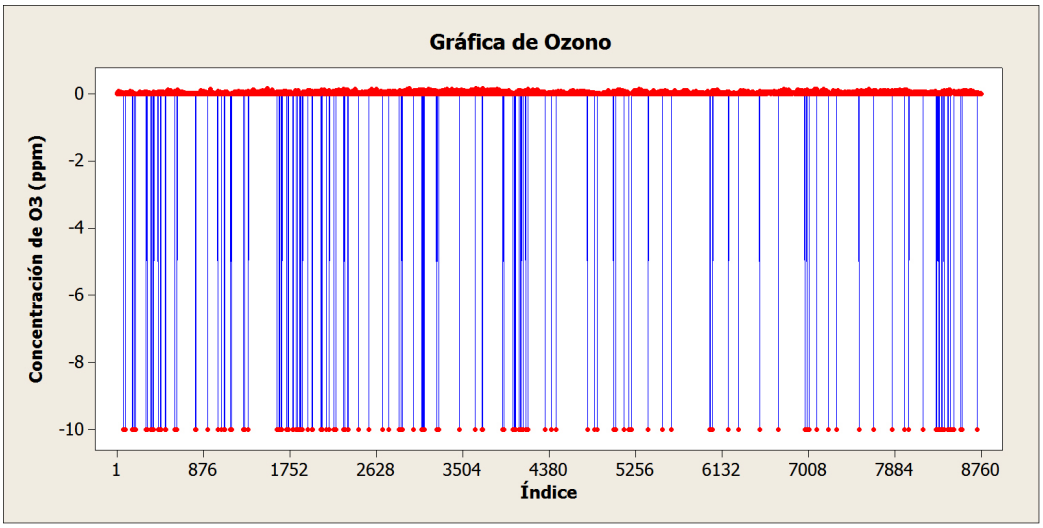


Figura B.4: Datos originales de (O3) en la estación Coyoacán (2010)

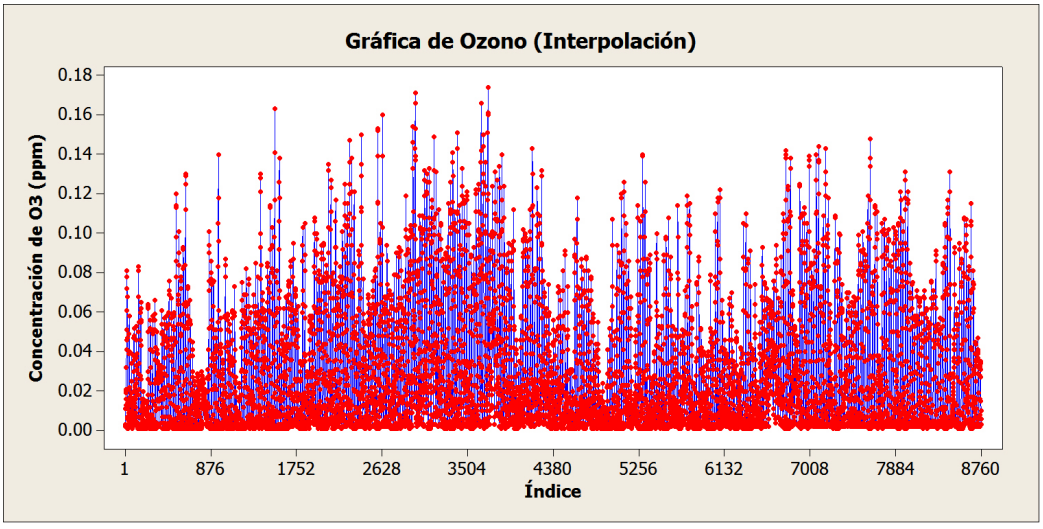


Figura B.5: Datos de (O3) con interpolación (pchip) (2010)

Apéndice C

Software desarrollado

En este Apéndice se presenta el software desarrollado para probar el algoritmo propuesto en este trabajo de tesis, también, se muestran las dos modalidades descritas en el capítulo 4. Este software fue diseñado y desarrollado en Visual Basic 6.0.

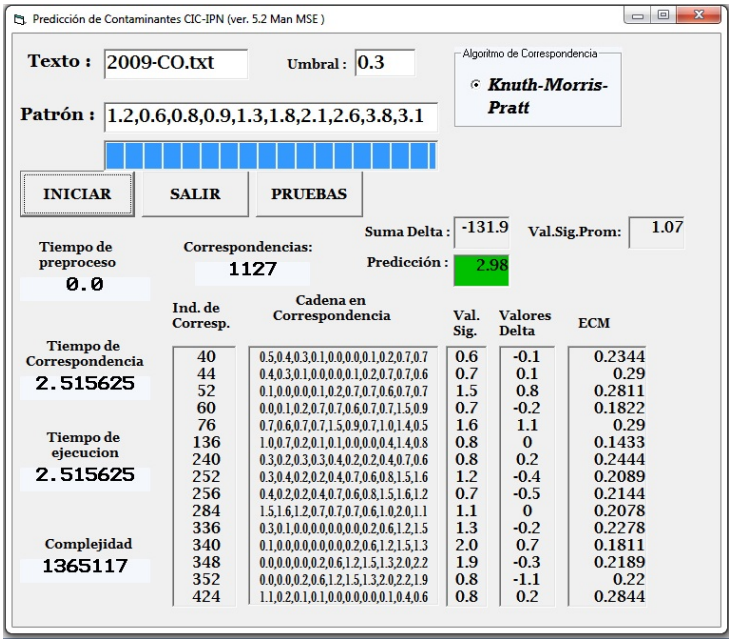


Figura C.1: Versión del software modalidad *MANUAL*

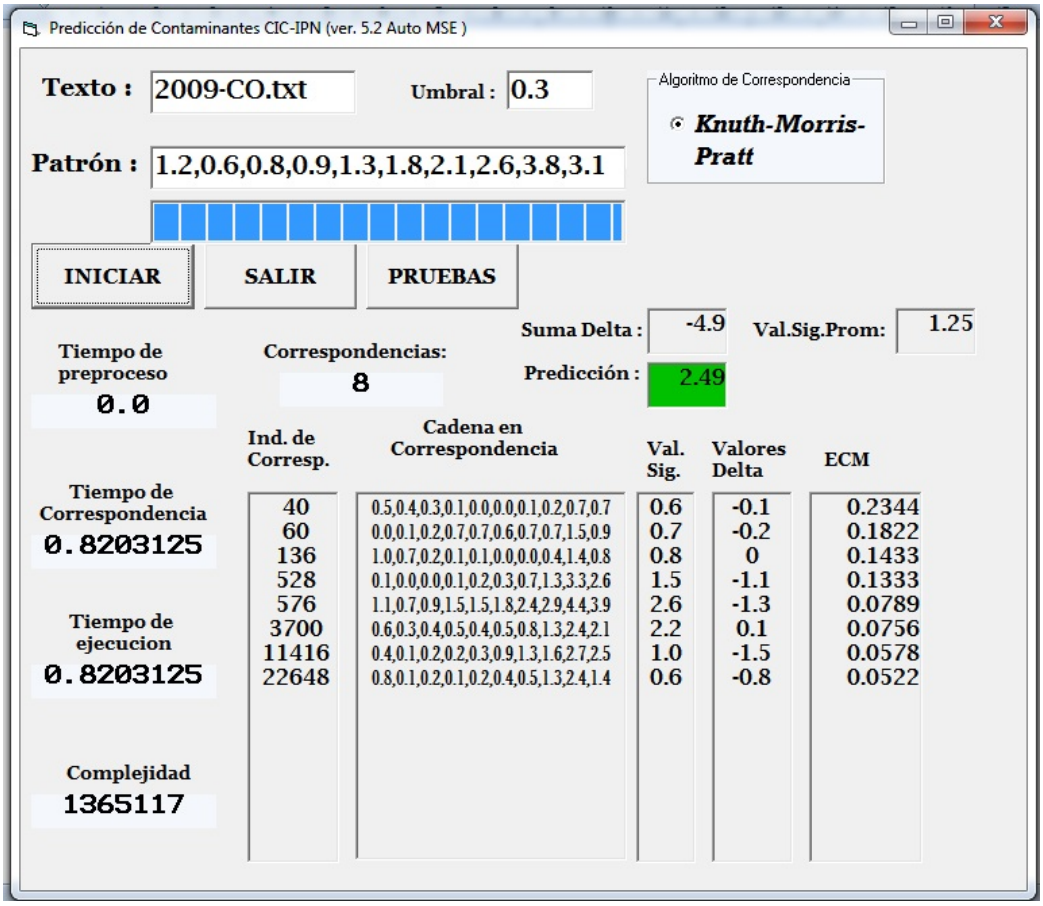


Figura C.2: Versión del software modalidad *AUTO*

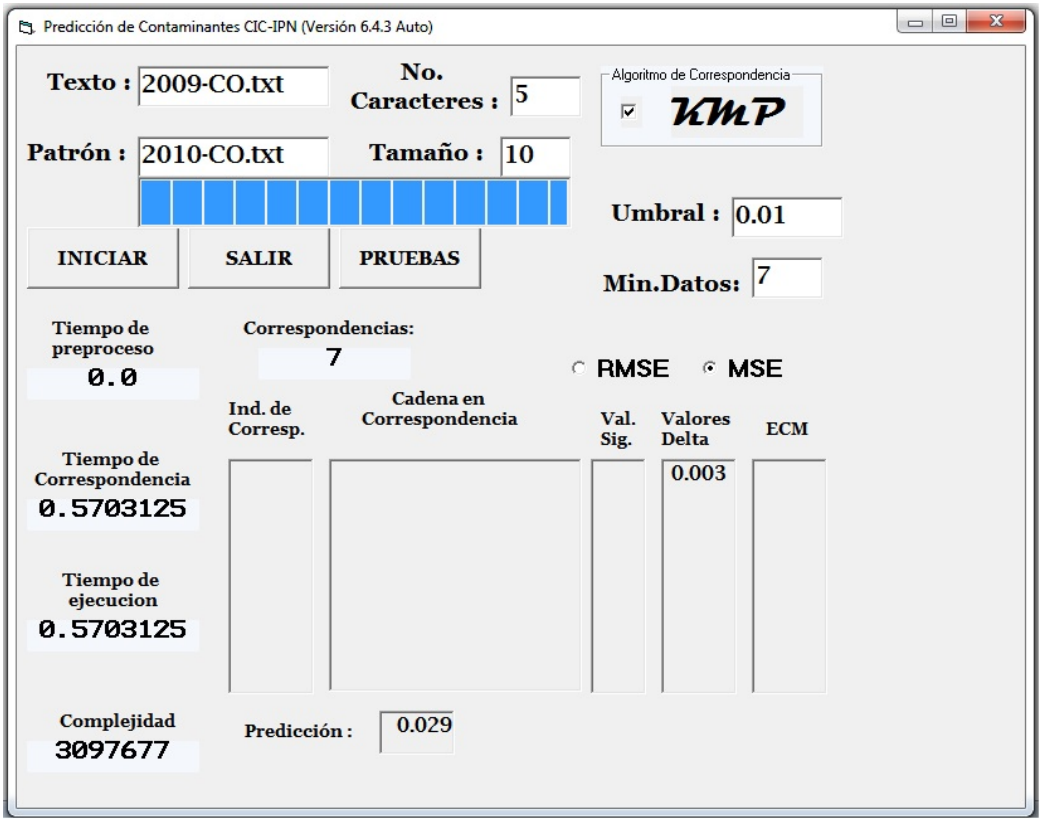


Figura C.3: Versión final del software.

Bibliografía

[1] Michelle L. Bell, Devra L. Davis, Nelson Gouveia, Víctor H. Borja-Aburto, and Cifuentes Luis A. The avoidable health effects of air pollution in three latin american cities: Santiago, São Paulo, and México City. *Environmental Research*, 100(3):431 – 440, 2006.

[2] G. H. Revlett. Ozone forecasting using empirical modeling. *Journal of the Air Pollution Control Association*, 28(4):338–343, 1978.

[3] P. A. Plato, D. F. Menker, and M. Dauer. Computer model for the prediction of the dispersion of airborne radioactive pollutants. *Health physics*, 13(10):1105–1115, 1967.

[4] H. Moses, J. B. Anderson, and D. F. Gatz. The tabulation technique for forecasting concentrations of urban air pollutants. ANL-7615. *ANL [reports].U.S.Atomic Energy Commission*, pages 167–179, 1968.

[5] Saxena U and Tsao KC. New method to forecast air pollutant concentrations. *ASME Pap 71-WA/APC1*, 1971.

[6] G. I. Sidorenko and M. A. Pinigin. Prediction of long-term maximum permissible concentration values of atmospheric pollutants based on rapid tests. *Zeitschrift fur die Gesamte Hygiene und ihre Grenzgebiete*, 18(12):880–882, 1972.

[7] C. M. Sheih and W. J. Moroz. A Lagrangian puff diffusion model for the prediction of pollutant concentrations over urban areas. *Proc. Third Inter. Clean Air Congress, VDI Verlag-GMBH*, pages B43–52, 1973.

[8] G. M. McCollister and K. R. Wilson. Linear stochastic models for forecasting daily maxima and hourly concentrations of air pollutants. *Atmospheric Environment*, 9(4):417–423, 1975.

[9] F. Hesek, D. Zavodsky, and S. Skulec. Mathematical modelling and forecasting the air pollution. *Metereol.Zpr.*, (30):124–128, 1977.

[10] H. Tokumaru and O. Habata. Prediction of pollution levels by mixed order multi-variable AR scheme. *Automatica*, 14(6):597–601, 1978.

[11] R. L. Carlson and M. M. Umble. Forecasting levels of air pollution: A conditional markovian analysis. *Journal of environmental management*, 8(3):215–221, 1979.

[12] M. T. Dmitriev, G. P. Zarubin, and V. A. Mishchikhin. Prediction of air pollution levels in quarters that make use of synthetic polymeric materials. *Gigiena i sanitariia*, (12):55–58, 1982.

[13] P. Holnicki and A. Zcochowski. A computer model for short-term forecasting and controlling air quality in a city. *Annual Review in Automatic Programming*, 12(PART 2):171–174, 1985.

[14] Takeshi Kawamura. Forecasting of air pollution potential in the south Kanto District in Japan. *Atmospheric Environment - Part A General Topics*, 20(10):2068, 1985.

[15] Piotr Holnicki, Andrzej Kaluszko, Marek Kurowski, Roman Ostrowski, and Antoni Zochowski. Urban-scale computer model for short-term prediction of air pollution. *Archiwum Automatyki i Telemechaniki*, 31(1-2):51–71, 1986.

[16] F. Haghighat, P. Fazio, and T. E. Unny. A predictive stochastic model for indoor air quality. *Building and Environment*, 23(3):195–201, 1988.

[17] V. V. Sukhanov and O. N. Putilina. Technique for forecasting air pollution in mines while applying new synthetic materials. *Meditsina Truda I Promyshlennaya Ekologiya*, (8):16–19, 1988.

[18] D. Pascual, M. L. Sanchez, M. C. Ramos, and I. Perez. A stochastic model to forecast lead pollutant. *Nuovo Cimento C Geophys.Space Phys.*, 12(4):415–425, 1989.

[19] S. M. Robeson and D. G. Steyn. A conditional probability density function for forecasting ozone air quality data. *Atmospheric Environment*, 23(3):689–692, 1989.

[20] M. Williams and T. Yamada. A microcomputer-based forecasting model: Potential applications for emergency response plans and air quality studies. *Journal of the Air and Waste Management Association*, 40(9):1266–1274, 1990.

[21] A. Kumar and P. Goyal. Forecasting of daily air quality index in Delhi. *Science of the Total Environment*, 2011. Article in Press.

[22] Margarita Vázquez-Garfias, Javier Audry-Sánchez, and Francisco Javier Garfias. Tropospheric ozone prediction in México city. *Journal of the Mexican Chemical Society (en línea)*, 49(1):2–9, 2005.

[23] A. C. Comrie. Comparing neural networks and regression models for ozone forecasting. *Journal of the Air and Waste Management Association*, 47(6):653–663, 1997.

[24] Anastasia Paschalidou, Spyridon Karakitsios, Savvas Kleanthous, and Pavlos Kassomenos. Forecasting hourly PM10 concentration in Cyprus through artificial neural networks and multiple regression models: implications to local environmental management. *Environmental Science and Pollution Research*, 18(2):316–327, 2011.

[25] S. I. V. Sousa, F. G. Martins, M. C. M. Alvim-Ferraz, and M. C. Pereira. Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. *Environmental Modelling and Software*, 22(1):97–103, 2007.

[26] S. M. Al-Alawi, S. A. Abdul-Wahab, and C. S. Bakheit. Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone. *Environmental Modelling and Software*, 23(4):396–403, 2008.

[27] Chin Cheng, Sue Huang, and Hia Teoh. Predicting daily ozone concentration maxima using fuzzy time series based on a two-stage linguistic partition method. *Computers and Mathematics with Applications*, 62(4):2016–2028, 2011.

[28] Vishy Karri and Tien Ho. Predictive models for emission of hydrogen powered car using various artificial intelligent tools. *Neural Computing And Applications*, 18(5):469–476, 2009.

[29] M. Hossain, Md. Hassan, and Michael Kirley. Forecasting urban air pollution using HMM-Fuzzy model. In *Advances in Knowledge Discovery and Data Mining*, volume 5012 of *Lecture Notes in Computer Science*, pages 572–581. Springer Berlin / Heidelberg, 2008.

[30] J. C. M. Pires, M. C. M. Alvim-Ferraz, M. C. Pereira, and F. G. Martins. Prediction of tropospheric ozone concentrations: Application of a methodology based on the Darwin's theory of evolution. *Expert Systems with Applications*, 38(3):1903–1908, 2011.

[31] Luis Enrique Sucar, Joaquín Pérez-Brito, J. Carlos Ruiz-Suárez, and Eduardo Morales. Learning structure from data and its application to ozone prediction. *Applied Intelligence*, 7(4):327–338, 1997.

[32] J. Kukkonen, L. Partanen, A. Karppinen, J. Ruuskanen, H. Junninen, M. Kolehmainen, H. Niska, S. Dorling, T. Chatterton, R. Foxall, and G. Cawley. Extensive evaluation of neural network models for the prediction of NO2 and PM10 concentrations, compared with a deterministic modelling system and measurements in central Helsinki. *Atmospheric Environment*, 37(32):4539–4550, 2003.

[33] W. Wang, W. Lu, X. Wang, and A. Y. T. Leung. Prediction of maximum daily ozone level using combined neural network and statistical characteristics. *Environment international*, 29(5):555–562, 2003.

[34] A. Elkamel, S. Abdul-Wahab, W. Bouhamra, and E. Alper. Measurement and prediction of ozone levels around a heavily industrialized area: A neural network approach. *Advances in Environmental Research*, 5(1):47–59, 2001.

[35] C. Borrego, A. Monteiro, M. T. Pay, I. Ribeiro, A. I. Miranda, S. Basart, and J. M. Baldasano. How bias-correction can improve air quality forecasts over Portugal. *Atmospheric Environment*, 45(37):6629–6641, 2011.

[36] P. Zito, Haibo Chen, and M.C. Bell. Predicting real-time roadside CO and concentrations using neural networks. *Intelligent Transportation Systems, IEEE Transactions on*, 9(3):514 –522, 2008.

[37] G. P. Zhang and M. Qi. Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, 160(2):501–514, 2005.

[38] A. Dutot, J. Rynkiewicz, F. E. Steiner, and J. Rude. A 24-h forecast of ozone peaks and exceedance levels using neural classifiers and weather predictions. *Environmental Modelling and Software*, 22(9):1261–1269, 2007.

[39] A. Kurt and A. B. Oktay. Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks. *Expert Systems with Applications*, 37(12):7986–7992, 2010.

[40] H. Sug. Ozone day prediction with radial basis function networks. In *International Conference on Systems - Proceedings*, volume 1, pages 608–611, 2010.

[41] G. Chattopadhyay and S. Chattopadhyay. Predicting daily total ozone over Kolkata, India: Skill assessment of different neural network models. *Meteorological Applications*, 16(2):179–190, 2009.

[42] C. Paoli, G. Notton, M. . Nivet, M. Padovani, and J. . Savelli. A neural network model forecasting for prediction of hourly ozone concentration in Corsica. In *2011 10th International Conference on Environment and Electrical Engineering, EEEIC.EU 2011 - Conference Proceedings*, 2011.

[43] B. Özbay, G. A. Keskin, S. C. Dogruparmak, and S. Ayberk. Predicting tropospheric ozone concentrations in different temporal scales by using multilayer perceptron models. *Ecological Informatics*, 6(3-4):242–247, 2011.

[44] L. Hrust, Z. B. Klaic, J. Krizan, O. Antonic, and P. Hercog. Neural network forecasting of air pollutants hourly concentrations using optimised temporal averages of meteorological variables and pollutant concentrations. *Atmospheric Environment*, 43(35):5588–5596, 2009.

[45] L. Cao. Support vector machines experts for time series forecasting. *Neurocomputing*, 51:321–339, 2003.

[46] Wenjian Wang, Changqian Men, and Weizhen Lu. Online prediction model based on support vector machine. *Neurocomput.*, 71:550–558, January 2008.

[47] Otávio Carpinteiro, João Leite, Carlos Pinheiro, and Isaías Lima. Forecasting models for prediction in time series. *Artificial Intelligence Review*, pages 1–9, 2011.

[48] I. López-Yáñez, A. J. Argüelles-Cruz, O. Camacho-Nieto, and C. Yáñez-Márquez. Pollutants time-series prediction using the gamma classifier. *International Journal of Computational Intelligence Systems*, 4(4):680–711, 2011.

[49] Cornelio Yáñez-Márquez, Itzamá López-Yáñez, and Guadalupe de la Luz Sáenz Morales. Analysis and prediction of air quality data with the Gamma classifier. In *Progress in Pattern Recognition, Image Analysis and Applications*, volume 5197 of *Lecture Notes in Computer Science*, pages 651–658. Springer Berlin / Heidelberg, 2008.

[50] Donald E. Knuth, James H. Morris, and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.

[51] Erick. N. Cabrera-Alvarez. *Algunos algoritmos de correspondencia de cadenas*. Tesis de licenciatura, Instituto Politécnico Nacional-Escuela Superior de Física y Matemáticas, 2011.

[52] Tommaso Toffoli. Non conventional computers. *Encyclopedia of Electrical and Electronics Engineering*, 14:455–471, 1998.

[53] James C. Bezdek. On the relationship between neural networks, pattern recognition and intelligence. *Int. J. Approximated Reasoning*, 6:85–102, 1992.

[54] David Poole, Alan Mackworth, and Randy Goebel. *Computational intelligence: A logical approach*. Oxford University Approach, New York, 1998.

[55] L.A. Zadeh. Soft computing and fuzzy logic. *IEEE Software*, pages 48–58, 1994.

[56] Raul Rojas. *Neural networks: A systematic introduction*. Springer, 1 edition, 1996.

[57] H.J Zimmermann. *Fuzzy Set Theory And Its Applications*. Kluwer Academic Publishers, Dordrecht, 2nd Edition, 1991.

[58] A. Kaufmann. *Introduction to the Theory of Fuzzy Sets*. Academic Press, New York, 1975.

[59] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5:115–37, 1943.

[60] V. Cherkassky and F Mulier. *Learning from Data*. John Wiley & Sons, New York, 1998.

[61] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 5(4):455–455, 1992.

[62] G. Zhang, B. Patuwo, and M. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998.

[63] F. Rosenblatt. *Principles of aerodynamics: perceptrons and the theory of brain mechanics*. Spartan Books, Washington D.C., 1962.

[64] M. Minsky and S. Papert. *Perceptrons*. MIT Press Cambridge, MA,USA, 1969.

[65] P. J Werbos. *Beyond regression new tools for prediction and analysis in the behavioral sciences*. Ph.d thesis, Harvard University, 1974.

[66] C. Cortes and Vapnik V. Support vector network. *Machine Learning*, 20:273–297, 1995.

[67] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[68] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, (9):155–161, 1997.

[69] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

[70] Christopher Chatfield. *Time-Series Forecasting*. Chapman & Hall/CRC, London U.K., 2000.

[71] George E. Box, Gwilym Jenkins, and Gregory Reinsel. *Time Series Analysis*. John Wiley & Sons, New York. Fourth Edition, 2008.

[72] Bruce L. Bowerman, Richard T. O'Connell, and Anne B. Koehler. *Forecasting, time series, and regression: An applied approach*. South-Western College, New York. Fourth Edition, 2004.

[73] S.L. Ho and M. Xie. The use of ARIMA models for reliability forecasting and analysis. *Computers Industrial Engineering*, 35(1-2):213–216, 1998.

[74] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press Cambridge, MA,USA. Second Edition, 2001.

[75] A. Aho, J. Hopcroft, and J. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, MA, 1974.

[76] S. Suhartono. Time series forecasting by using seasonal autoregressive integrated moving average: Subset, multiplicative or additive model. *Journal of Mathematics and Statistics*, 7(1):20–27, 2011.

[77] Richard L. Burden and J. Douglas Faires. *Análisis Numérico*. Cengage Learning Editores, Mexico D.F., 2001.

[78] Sistema de Monitoreo Atmosférico de la ciudad de México, 2011. http://www.calidadaire.df.gob.mx/calidadaire/index.php.