COLEGIO DE POSTGRADUADOS

INSTITUCIÓN DE ENSEÑANZA E INVESTIGACIÓN
EN CIENCIAS AGRÍCOLAS

CAMPUS MONTECILLO

POSTGRADO DE SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA
ESTADÍSTICA

Máquinas de Soporte Vectorial en el Análisis de Series de Tiempo

Enrique Rivera Castillo

T E S I S

PRESENTADA COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

MONTECILLO,TEXCOCO, EDO. DE MÉXICO
2012

La presente tesis titulada: **Máquinas de Soporte Vectorial en el Análisis de Series de Tiempo**, realizada por el alumno: **Enrique Rivera Castillo**, bajo la dirección del Consejo Particular indicado ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN CIENCIAS

SOCIOECONOMÍA, ESTADÍSTICA E INFORMÁTICA ESTADÍSTICA

CONSEJO PARTICULAR

CONSEJERO

Dr. Enrique Arjona Suárez

ASESOR

Dra. Graciela M. Bueno Aguilar

ASESOR

Dr. José A. Villaseñor Alva

ASESOR

Dr. Miguel Sánchez Hernández

Montecillo, Texcoco, Estado de México, julio de 2012

Máquinas de Soporte Vectorial en el Análisis de Series de Tiempo

La evapotranspiración de referencia (ETo) es un proceso no lineal empleado para determinar la cantidad de agua utilizada en los programas de irrigación. El nivel de precisión de esta variable a partir de datos históricos, ha sido siempre fundamental. En este trabajo, se presenta una aplicación de las Máquinas de Soporte Vectorial (SVMs) para la predicción de ETo y se compara su capacidad predictiva con otras dos metodologías de predicción: Redes Neuronales Artificiales de Multicapa (MLP) y modelos Autoregresivos Integrados de Promedio Móvil (ARIMA). Se propone un algoritmo heurístico de refinamiento para la implementación de las SVM resultando en una predicción mucho mejor que la obtenida con los otros dos métodos. La capacidad de predicción fue evaluada utilizando el Error Porcentual Medio Absoluto (MAPE).

Palabras clave: evapotranspiración, red neuronal, predicción, máquina de soporte vectorial.

Support Vector Machines in the Time Series Analysis

Reference crop evapotranspiration (ETo) is a non linear process used to determine the quantity of water used in irrigation programs and the level of accuracy of the prediction of this variable from historical data has always been fundamental. In this work, we present an application of Support Vector Machines (SVMs) for ETo forecasting and compare its prediction capacity with two other prediction methodologies: Multi Layer Perceptron (MLP) neural networks and Auto-Regressive Integrated Moving Average (ARIMA) models. A proposed heuristic refinement algorithm for the implementation of the SVM gave a very good forecasting, much better than those obtained with the other two methods. Forecasting capacity was evaluated using the Mean Absolute Percentage Error (MAPE).

Key words: evapotranspiration, neural network, forecasting, support vector machine.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado para la realización de mis estudios de postgrado.

Al Colegio de Postgraduados, por haberme recibido como estudiante y por la direccionalidad que he recibido de su cuerpo de profesores.

Dedicatoria

A mi madre, hermanos e hijo.

Índice

Índice de figuras

Capítulo 1

Introducción

La evapotranspiración de referencia (ETo) es una variable climática que se define como la combinación de dos procesos separados a través de los cuales se pierde el agua en una plantación o cultivo. En primer lugar, el agua se pierde de la superficie o suelo mediante el proceso de evaporación, y por otro lado, la pérdida se origina en el propio cultivo por medio del proceso de transpiración (Allen et al., 1998).

Por una parte, la evaporación está definida como el proceso por medio del cual el agua se transforma del estado líquido al estado gaseoso o vapor de agua (vaporización) y de esta forma es removida de la superficie principalmente a causa del aumento de la temperatura y acción del viento; por otro lado, el proceso de transpiración consiste en la vaporización del agua en estado líquido que se encuentra contenida en los tejidos de las plantas, a través de sus estomas. Ambos procesos ocurren de manera simultánea, por tanto resulta muy compleja la distinción entre la pérdida de agua debido a cada uno de los procesos de manera aislada, consecuentemente la ecuación que los modela debe estar en función de las variables involucradas en ambos fenómenos.

La ETo es una variable climática esencial en investigaciones en las que se involucra el balance energético, cálculo del balance hídrico, estimación del consumo de agua, programas de irrigación, entre otras características de relevancia en los cultivos. El cálculo de la ETo se basa en la fórmula de *Penman-Monteith FAO98*, esta fórmula tiene validez mundial en los diferentes tipos de clima, debido a que provee los resultados más consistentes para el uso real del agua en los diferentes tipos de cultivos, y ha sido verificada por organismos especializados como la Organización Metereológica Mundial (OMM). En base a lo anterior, esta fórmula de Penman-Monteith toma importancia en el cálculo de la programación de los volúmenes de agua a distribuirse en los distritos de riego.

En México, la programación de la distribución de los volúmenes de agua en los distritos de riego se realiza en periodos semanales, para lo cual se requiere de una estimación

confiable y robusta de la ETo. La estimación de los volúmenes de agua se basa, la mayoría de las veces, en un análisis de los datos registrados de los volúmenes de agua distribuidos en años anteriores; otra forma alternativa de estimar la distribución de agua de las semanas siguientes está basada en los valores promedio de la ETo de la semana anterior, por lo tanto, es altamente probable la mala estimación de la demanda de agua en los cultivos y puede dar lugar a grandes pérdidas de este líquido en los sistemas de distribución o redimiento en los cultivos (González et al., 2008).

Derivado de lo anterior, resulta de importancia la implementación de métodos novedosos como las Máquinas de Soporte Vectorial para la predicción de los valores de evapotranspiración de referencia y que, de acuerdo a la literatura, han demostrado tener una buena capacidad predictiva en problemas de clasificación o reconocimiento de patrones (Chi-Wei et al., 2010) y en problemas en los que se estima una regresión basada en las máquinas de aprendizaje con fines predictivos.

Capítulo 2

Antecedentes

En términos generales, una serie de tiempo se define como un proceso estocástico o sucesión ordenada a lo largo del tiempo de un conjunto de variables aleatorias, de forma tal que con una determinada realización del proceso se tiene solamente un valor u observación de cada una de las variables aleatorias que integran el sistema, y estos a su vez, evolucionan en el tiempo de acuerdo a las leyes probabilísticas (Guerrero, 2003).

En este sentido, el análisis de series de tiempo se define como el estudio de las observaciones realizadas secuencialmente en el tiempo, considerando las relaciones internas que tienen estas observaciones tales como la estructura de correlación, tendencia, estacionariedad, entre otras. La complejidad de una serie de tiempo está basada en los efectos directos e indirectos que tiene el tiempo en las variables del modelo. La metodología que tradicionalmente se utiliza para analizar el comportamiento de una serie de datos dependientes en el tiempo es la propuesta por Box y Jenkins (1970). De acuerdo a estos autores, la estrategia de construcción de un modelo consta de las etapas siguientes:

- Identificación del modelo.
- Estimación de parámetros implícitos en el modelo.
- Verificación de supuestos.
- Uso del modelo.

Considerando su campo de aplicación, las series de tiempo pueden ser estudiadas en el dominio del tiempo y en el dominio de la frecuencia en el tiempo. En el estudio con base en la frecuencia en el tiempo, el interés del investigador recae en las propiedades de frecuencia y hace uso de una función de densidad espectral para el análisis, por lo

que se trata de descomponer la serie en componentes cíclicos diferentes. Cuando se trabaja con series con dominio en el tiempo, es de interés la evolución del proceso en el tiempo y para su análisis se utiliza la función de autocorrelación entre las observaciones actuales y las pasadas.

Respecto al estudio en el dominio del tiempo, es posible modelar las series de tiempo a partir de dos enfoques:

- El enfoque tradicional siguiendo la metodología propuesta por Box y Jenkins. Esta metodología incluye los Modelos Autorregresivos Integrados de Medias Móviles (ARIMA), los cuales se caracterizan por ser modelos multiplicativos, lo que significa que los datos observados son el resultado de productos de factores que incluyen operadores de ecuaciones diferenciales que corresponden a ecuaciones de ruido blanco.
- El enfoque que utiliza modelos aditivos o estructurales. En este enfoque se asume que las observaciones incluyen la suma de componentes, cada uno de los cuales se ocupa de una estructura especificada de serie de tiempo.

La predicción de variables de tipo ambiental dependientes en el tiempo, se ha convertido en uno de los problemas más desafiantes debido a su valor práctico en el monitoreo climático, detección de sequía, predicción de climas severos, producción y agricultura, planeación en la industria de producción de energía, planeación aérea, comunicación y dispersión de contaminantes ambientales, entre otros (Radhika y Sashi, 2009).

Los modelos ARIMA, se han utilizado ampliamente en las ciencias ambientales, obteniendo muy buenos resultados en términos de inferencia y predicción, sobre todo para el análisis de series de tiempo univariadas, manteniendo modelos relativamente simples (Robenson y Steyin, 1989); sin embargo, han mostrado ser conservadores en tanto a la precisión en el aspecto predictivo cuando los datos tienen un comportamiento complejo en las escalas temporales y espaciales.

Por el tipo de información que generalmente se maneja y la importancia de la precisión en el aspecto predictivo, se ha comparado el desempeño de los modelos ARIMA con métodos alternativos fundamentados en la teoría del aprendizaje estadístico, inicialmente con el modelo clásico de máquinas de aprendizaje conocidas como Redes Neuronales Artificiales (ANN) por sus siglas en inglés (Prybutok, et al. 2000).

Las ANN han mostrado ser eficientes en investigaciones relacionadas con la predicción de variables de tipo atmosférico, predicciones climatológicas, contaminantes ambientales y en problemas en los que interviene de manera directa la temperatura ambiental. Se han reportado varias investigaciones en las que se muestran mejores resultados cuando se emplean las ANN al compararlos con los métodos tradicionales, los cuales

incluyen los modelos ARIMA (Radhika y Sashi, 2009; Tektas, 2010). Es necesario mencionar que la comparación de los métodos ARIMA y las ANN, debido a que son herramientas de distinta naturaleza, está en función solamente de su capacidad predictiva o precisión al momento de predecir.

Las redes neuronales tienen la capacidad de manejar problemas no lineales complejos, no obstante, de acuerdo al fundamento matemático que las sustenta, es posible que se tengan errores de *overfitting* o sobreentrenamiento o que, dependiendo del tipo de datos con que se esté trabajando, se lleguen a generar modelos poco precisos como consecuencia de que en la solución del problema de programación no lineal que plantean, se caiga erróneamente en un punto de mínimo local. Otra desventaja que se ha encontrado en las ANN, es que con datos que presentan variabilidad, ruido o son de una dimensión compleja, en las redes neuronales de propagación hacia atrás, que son los modelos más comunes de ANN, la estimación de los parámetros intrínsecos de las ANN se convierte en un proceso muy complejo, aunado a que la estimación de los parámetros es en base a una heurística (Kim, 2003).

Otra herramienta, que tiene también su fundamento en el aprendizaje estadístico son las Máquinas de Soporte Vectorial (SVM). Los fundamentos teóricos de esta herramienta son contemporáneos y un tanto similares a los de las ANN, sin embargo en un inicio se le dio mayor difusión y aplicación a las ANN en la solución de problemas con datos reales, de ahí que se considere a las SVM como un método novedoso.

La construcción de las Máquinas de Soporte Vectorial se basa en la idea de transformar o proyectar un conjunto de datos perteneciente a una dimensión dada, a un espacio de dimensiones superiores mediante el uso de una función *kernel*, y a partir del espacio característico de dimensión superior, operarlos como si se tratase de un problema de tipo lineal, lo cual significa que el problema se resuelve sin considerar la dimensionalidad de los datos (Vapnik, 1999).

Las SVM originalmente se emplearon para resolver problemas de clasificación o de reconocimiento de patrones y de acuerdo a los resultados obtenidos se extendió su aplicación a problemas de regresión (Chih-Wei et al., 2010); recientemente se han utilizado para la predicción de series de tiempo, aunque las experiencias reportadas son un tanto limitadas y se han encontrado algunos problemas relacionados con su especificación o el establecimiento de sus parámetros intrínsecos (Velásquez, et al., 2010).

Las ANN y las SVM encuentran un modelo a partir de las características o de la información extraída de los datos, por lo que para la formulación de un modelo se utilizan datos de entrenamiento, validación y prueba. Los datos de entrenamiento se emplean para definir el modelo; mientras que el conjunto de datos de validación es utilizado para encontrar los mejores parámetros y finalmente, los datos de prueba se utilizan para corroborar que el modelo encontrado es el óptimo. En este sentido, se han realizado comparaciones en términos de sus fundamentos teóricos y de la formu-

lación matemática e implementación de ambos métodos. Una de las diferencias que sobresale es que en la fase de entrenamiento, las SVM siempre encuentran un mínimo global, no así las ANN, que son susceptibles a caer en un mínimo local (Burges, 1998; Taylor y Cristianini, 2004). En contraste a las ANN, las SVM seleccionan automáticamente el tamaño de sus modelos a partir del número de vectores soporte. El desarrollo de las ANN sigue una trayectoria heurística, con aplicaciones y un experimentación extensa que precede a la teoría, mientras que el desarrollo de las SVM se fundamenta en la teoría y posteriormente en la implementación y experimentación.

Debido a su capacidad para extraer información a partir de un conjunto de muestras, las Máquinas de Soporte Vectorial se han convertido en una técnica para predecir series de tiempo; inicialmente se realizaron predicciones sobre datos sintéticos obteniendo buenos resultados incluso para series de datos caóticos (Müller, et al., 1998). Una de las debilidades de esta herramienta se encuentra en la determinación heurística de sus parámetros libres, en la mayoría de las implementaciones se utiliza la técnica de *cross validation* para hallar los valores óptimos, sin embargo esto se traduce a un cómputo más exhaustivo de los datos.

Las SVM se han aplicado al análisis de varios tipos de datos. La literatura muestra algunas aportaciones a datos de origen financiero dependientes en el tiempo (Cao y Tay, 2003), aplicaciones en la predicción de contaminantes ambientales (Lu y Wang, 2005), así como en la predicción de temperatura atmosférica (Radhika y Sashi, 2009). De acuerdo a estos reportes, la naturaleza de los datos es determinante en la estimación de los parámetros libres, dejando esta tarea a quien implementa estas herramientas.

2.1. Planteamiento del problema

De acuerdo a la literatura, los resultados de las SVM en problemas de clasificación y regresión, han mostrado ser susceptibles a los valores de los parámetros libres que resultan de resolver el problema de programación cuadrática que las modela, por lo que la predicción utilizando series de tiempo no es la excepción. En términos de predicción, tienen ventaja sobre los métodos estadísticos utilizados tradicionalmente e incluso sobre las ANN; en este sentido, se propone utilizar las SVM para formular un modelo capaz predecir con un alto grado de precisión, los valores diarios de evapotranspiración de referencia partiendo de datos históricos registrados desde febrero de 1997 a junio de 2001.

Uno de los problemas que se enfrentan al momento de implementar las SVM es la estimación de los parámetros libres C y ϵ. Para resolver este problema se plantea utilizar la metodología propuesta por Cherkassky y Yunquian (2004) y la propuesta por Velásquez et al. (2010); asimismo se utilizará un método heurístico para la estimación del parámetro σ que se encuentra involucrado en el kernel de base radial o

kernel gausiano que es la función kernel que, de acuerdo a la revisión bibliográfica, ha mostrado mejores resultados en problemas de clasificación y regresión.

2.2. Hipótesis

Las Máquinas de Soporte Vectorial, por su formulación matemática, tienen mayor capacidad predictiva que los métodos ARIMA y las ANN en el análisis de series de tiempo, para un conjunto de datos disponibles de evapotranspiración de referencia.

2.3. Objetivos

2.3.1. Objetivo general

Diseñar e implementar una metodología para el análisis de series de tiempo de evapotranspiración de referencia, utilizando Máquinas de Soporte Vectorial.

2.3.2. Objetivos específicos

- Pronosticar los valores de una serie de tiempo de datos de evapotranspiración de referencia utilizando Máquinas de Soporte Vectorial, modelos ARIMA y ANN.
- Comparar la capacidad predictiva de las Máquinas de Soporte Vectorial con un modelo ARIMA y con redes neuronales artificiales.

Capítulo 3

Marco Teórico

La teoría del aprendizaje estadístico desarrollada por Vladimir N. Vapnik, fue introducida a finales de 1960 y en un principio su desarrollo fue principalmente en el análisis teórico del problema de la estimación de funciones dado un conjunto conocido de datos (Scholkopf et al., 1999). A mediados de 1990 se propuso un nuevo tipo de algoritmos de aprendizaje basados en esta teoría, que son conocidos como Máquinas de Soporte Vectorial (SVM). Estos algoritmos han permitido hacer de la teoría del aprendizaje estadístico no solamente una herramienta para el análisis teórico sino también una herramienta para crear algoritmos prácticos para estimar funciones multidimensionales.

En este capítulo se presentan algunos de los conceptos importantes en que se fundamenta la teoría del aprendizaje estadístico y que están relacionados con las máquinas de soporte vectorial.

3.1. El problema del aprendizaje a partir de ejemplos

El objetivo principal de la teoría del aprendizaje estadístico es proveer de los elementos necesarios para el estudio del problema de inferencia, esto quiere decir, obtener conocimiento, hacer predicciones y tomar decisiones o construir modelos partiendo de un conjunto de datos conocidos (Hastie et al., 2009). Respecto a la inferencia, la forma en que se realiza es mediante la elaboración de un modelo. El modelo general del aprendizaje a partir de ejemplos o aprendizaje supervisado, se describe a partir de tres componentes importantes (Vapnik, 1999).

- Un generador (G) de las condiciones que determinan el entorno en el cual actúan

el supervisor y la máquina de aprendizaje. En su forma más simple genera los vectores aleatorios $x \in \Re^n$ seleccionados de manera independiente a partir de una función de distribución $F(x)$ desconocida.

- Un supervisor (S) que asigna una respuesta y a cada vector x de acuerdo a la distribución condicional $F(y|x)$.
- Una máquina de aprendizaje (LM) capaz de implementar un conjunto de funciones $f(x, \alpha)$ con α que pertenece al conjunto de parámetros Λ; donde una de esas funciones tiene la característica de ser la más aproximada a la respuesta proporcionada por el supervisor.

De esta forma, el problema del aprendizaje basado en ejemplos o aprendizaje supervisado considerando a estos tres elementos, se define de la manera siguiente:

Definición 1 (Problema de aprendizaje) *Dado un conjunto de funciones $f(x, \alpha)$ con $\alpha \in \Lambda$, el problema del aprendizaje se enfoca a seleccionar aquella función que mejor se aproxima a la respuesta del supervisor (S), a partir de un conjunto de observaciones independientes e idénticamente distribuidas $(x_1, y_1), ..., (x_n, y_n)$ las cuales han sido seleccionadas de acuerdo a $F(x, y) = F(x)F(y|x)$.*

Para seleccionar la función que más se aproxima a la respuesta del supervisor, se busca minimizar la pérdida o discrepancia $L(y, f(x, \alpha))$ entre la respuesta y del supervisor y la respuesta $f(x, \alpha)$ proporcionada por la máquina de aprendizaje, en donde la esperanza o el valor esperado de la pérdida o discrepancia está dado por la función del riesgo funcional:

$$R(\alpha) = \int L(y, f(x, \alpha)) dF(x, y) \tag{3.1}$$

A partir de la ecuación del riesgo funcional, es posible hallar la ecuación $f(x, \alpha)$, que minimiza el valor de $R(\alpha)$, con $\alpha \in \Lambda$, en donde la función de distribución conjunta $F(x, y)$ es desconocida y solamente se cuenta con la información contenida en el conjunto de datos de entrenamiento $(x_1, y_1), ..., (x_n, y_n)$.

Cada problema práctico que se esté abordando, tiene su correspondiente ecuación del riesgo funcional que lo determina. Dentro de los problemas más importantes en que se aplican las máquinas de aprendizaje se encuentra el reconocimiento de patrones o proceso de clasificación supervisada, la estimación de densidades y la estimación de regresión, de esta última es en la que, por su aplicación, se abunda en este documento.

- *Estimación de regresión.* En este problema, la respuesta del supervisor (S) es un valor real, y $\{f(x, \alpha)\}$ es un conjunto de funciones reales, las cuales contienen

la función de regresión

$$f(x,\alpha) = \int y dF(y|x) \tag{3.2}$$

En este caso particular, la función de pérdida está dada en términos del cuadrado del error:

$$L(y, f(x,\alpha)) = (y - f(x,\alpha))^2 \tag{3.3}$$

.

En términos generales, el problema de aprendizaje trata de minimizar el riesgo funcional partiendo de un conjunto de datos empíricos o datos conocidos provenientes de una muestra aleatoria. Una forma de minimizar el riesgo funcional sabiendo que los datos provienen de una función de distribución, aunque desconocida, es aplicando el principio del riesgo funcional empírico construido a partir de los datos de entrenamiento.

$$R_{emp}(\alpha) = \frac{1}{l}\sum_{i=1}^{l} Q(z_i,\alpha) \tag{3.4}$$

Entonces, se minimiza el riesgo funcional a partir de la minimización del riesgo empírico, de tal forma que el proceso de aprendizaje está definido en términos de un principio inductivo y por lo tanto, para un conjunto dado de observaciones, la máquina de aprendizaje elige la mejor aproximación utilizando este principio. Para el caso de la regresión, la minimización del riesgo empírico está dado en términos de la expresión:

$$R_{emp} = \frac{1}{l}\sum_{i=1}^{l}(y_i - f(x_i,\alpha))^2 \tag{3.5}$$

que corresponde a la forma del método de mínimos cuadrados. De esta forma, en el problema de aprendizaje se tienen dos aspectos importantes a considerar:

- Estimar una función óptima a partir de un conjunto amplio de funciones.
- Estimar la función deseada a partir de un número dado de muestras conocidas.

En el caso de la estimación de regresión, el paradigma clásico está basado en el modelo de medición de una función con ruido aditivo.

Supongamos que una función desconocida tiene la forma paramétrica: $f_0(x) = f(x,\alpha_0)$, donde $\alpha_0 \in \Lambda$ es un vector de parámetros desconocidos. Supongamos además que para cualquier punto x_i es posible medir esta función con ruido aditivo:

$$y_i = f(x_i,\alpha_0) + \xi_i \tag{3.6}$$

Donde ξ_i no depende de x_i y se distribuye de acuerdo a una función de densidad conocida $p(\xi)$. Entonces, el problema es estimar la función $f(x, \alpha_0)$ de un conjunto $\{f(x, \alpha)\}$, $\alpha \in \Lambda$ usando los datos obtenidos de la función a estimar y que contienen ruido aditivo; es decir, utilizando el conjunto de observaciones $(x_1, y_1), ..., (x_l, y_l)$ es posible estimar los parámetros α_0 de la función desconocida $f(x, \alpha_0)$ utilizando el método de máxima verosimilitud. Concretamente, maximizando la función:

$$L(\alpha) = \sum_{i=1}^{l} ln[p(y_i - f(x_i, \alpha))] \tag{3.7}$$

Como $p(\xi)$ es una función conocida y $\xi = y - f(x, \alpha_0)$, considerando que el ruido tiene una distribución normal con media cero y varianza fija.

$$p(\xi) = \frac{1}{\sigma\sqrt{2\pi}} exp\left\{-\frac{\xi^2}{2\sigma^2}\right\} \tag{3.8}$$

Se obtiene el método de mínimos cuadrados:

$$L^*(\alpha) = -\frac{1}{2\sigma^2}\sum_{i=1}^{l}(y_i - f(x_i, \alpha))^2 - l * ln(\sqrt{2\pi\sigma^2}) \tag{3.9}$$

Maximizar $L^*(\alpha)$ sobre los parámetros α, es equivalente a minimizar la función que corresponde $M(\alpha) = \sum_{i=1}^{l}(y_i - f(x_i, \alpha))^2$

Es importante conocer, bajo qué condiciones una máquina que minimiza el riesgo empírico tiene la capacidad de reducir el error al momento de generalizar; es decir, una vez que se ha obtenido la función que minimiza el riesgo empírico, verificar si es que se minimiza el error al ser utilizada con datos que no pertenecen al conjunto de datos de entrenamiento. Entonces, es necesario conocer las condiciones de consistencia de las máquinas. Estas condiciones están dadas por la entropía de Vapnik-Chervonenkins (VC) (Cortes y Vapnik, 1995).

En primer lugar, es necesario definir la consistencia del principio de minimización del riesgo empírico.

Se dice que el principio de minimización del riesgo empírico es consistente, para el conjunto de funciones $Q(z, \alpha)$, con $\alpha \in \Lambda$ y para la función de distribución $F(z)$ si se cumplen las siguientes convergencias en probabilidad:

- $R(\alpha_l)_{l\to\infty} \to inf_{\alpha\in\Lambda} R(\alpha)$ y
- $R_{emp}(\alpha_l)_{l\to\infty} \to inf_{\alpha\in\Lambda} R(\alpha)$

Es decir, que el riesgo esperado y el riesgo empírico convergen al valor mínimo posible, bajo este esquema está formulado el siguiente teorema de consistencia de un solo lado:

Teorema 3.1
Sea $Q(z, \alpha)$, $\alpha \in \Lambda$, un conjunto de funciones que satisfacen la condición:

$A \leq \int Q(z, \alpha) dF(z) \leq B$ de otro modo $A \leq R(\alpha) \leq B$).

Entonces, para que el principio de minimización del riesgo empírico, sea consistente, una condición necesaria y suficiente es que el riesgo empírico $R_{emp(\lambda)}$ converja uniformemente al riesgo esperado $R(\lambda)$ sobre el conjunto $Q(z, \alpha)$, con $\alpha \in \Lambda$, en el sentido siguiente:

$lim_{l \to \infty} P \{sup_{\alpha \in \Lambda}(R(\alpha) - R_{emp}(alpha)) > \epsilon\} = 0, \forall \epsilon > 0$

En el caso de que se requiera consistencia uniforme (por ambos lados), se puede escribir de la siguiente manera:

$lim_{l \to \infty} P \{[sup_{\alpha}(R(\alpha) - R_{emp}(\alpha)) > \epsilon] \vee [sup_{\alpha}(R_{emp}(\alpha) - R(\alpha)) > \epsilon]\} = 0$

3.2. El hiperplano de separación óptimo

Otra herramienta utilizada en el problema de aprendizaje es el hiperplano de separación óptimo. Esta herramienta utiliza el principio de mantener el valor del riesgo empírico fijo y minimizar el intervalo de confianza. El hiperplano de separación óptimo, está basado en la teoría de Vapnik-Chervonenkis (Cortes y Vapnik, 1995).

Supongamos que se tienen los datos de entrenamiento $(x_1, y_1), ..., (x_l, y_l)$, $x \in R$, $y \in \{+1, -1\}$, que pueden ser separados por el hiperplano $(\langle w.x \rangle - b = 0)$.

Se dice que el conjunto de vectores de entrenamiento es separable por un hiperplano óptimo o por el hiperplano de margen máximo si los datos son separados sin error y la distancia entre los vectores más cercanos al hiperplano es máxima.

El hiperplano está dado por la ecuación:

$$y_i [\langle w.x_i \rangle - b] \geq 1 \tag{3.10}$$

con $i = 1, ..., l$

Resulta fácil de observar que el hiperplano óptimo satisface la expresión anterior

y minimiza:

$$\phi(w) = ||w||^2 \tag{3.11}$$

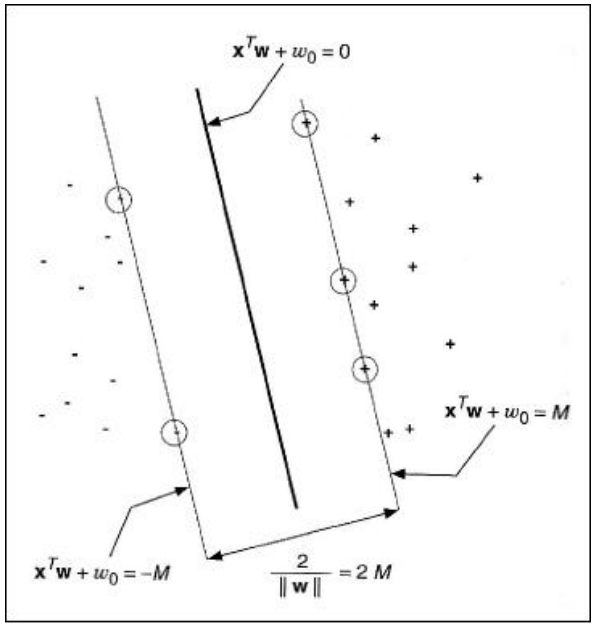

Figura 3.1: Hiperplano de separación óptimo

3.2.1. Hiperplano de separación con $\Delta-$margen

Se llama al hiperplano:

$$(w^*.x) - b = 0, ||w^*|| = 1 \tag{3.12}$$

el hiperplano de separación con $\Delta-$margen, si clasifica vectores x de la forma:

$$y = \begin{cases} 1 \text{ si } \langle w^*.x \rangle - b \geq \Delta \\ -1 \text{ si } \langle w^*.x \rangle - b \leq -\Delta \end{cases}$$

De ahí el siguiente teorema:

Teorema 3.2 *Si los vectores $x \in X$ pertenecen a una esfera de radio R, entonces el conjunto de hiperplanos de separación con $\Delta-$margen tiene la dimensión h, limitada por la desigualdad:*

$$h \leq min\left(\left[\frac{R^2}{\Delta^2}\right], n\right) + 1 \tag{3.13}$$

La dimensión de la dimensión Vapnik-Cherbonenkins VC puede ser menor a $n + 1$,

de esta forma se tiene:

Con probabilidad $1-\eta$ se puede establecer que la probabilidad de que las muestras de prueba no son separadas correctamente por el hiperplano con $\Delta-$margen, se definen por la desigualdad:

$$P_{error} \leq \frac{m}{l} + \frac{\epsilon}{2}\left(1+\sqrt{1+\frac{4m}{l\epsilon}}\right) \tag{3.14}$$

donde el error ϵ está dado por $\epsilon = 4\frac{h\left(ln\left(\frac{2l}{h}+1\right)\right)-ln(\eta/4)}{l}$, m es el número de muestras de entrenamiento que no son separadas correctamente con el hiperplano de separación con $\Delta-$margen y h es el limite de la dimensión VC.

3.2.2. Teorema de Mercer

Una condición necesaria y suficiente para garantizar que la función simétrica $K(u,v)$ que define un producto interno en un espacio característico L_2 tiene una expansión:

$$K(u,v) = \sum_{k=1}^{\infty} a_k \psi_k(u)\psi_k(v) \tag{3.15}$$

con coeficientes positivos $a_k > 0$ es la siguiente:

$$\int\int K(u,v)g(u)g(v)dudv > 0 \tag{3.16}$$

la cual es válida para toda $g \neq 0$ para la cual

$$\int g^2(u)du < \infty \tag{3.17}$$

La convolución del producto interno, permite la construcción de funciones de decisión que son no lineales en el espacio de entrada:

$$f(x) = sign\left(\sum_{vectoressoporte} y_i\alpha_i K(x_i,x) - b\right) \tag{3.18}$$

y que son equivalentes a las funciones de decisión en el espacio característico de altas dimensiones $\psi_1(x), ..., \psi_N(x)$ donde $K(x_i,x)$ es una convolución del producto interno en este espacio característico. Para encontrar los coeficientes α_i en el caso separable y

análogamente en el caso no separable, basta con encontrar el máximo de la función:

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{3.19}$$

sujeto a las restricciones:

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0, i, j = 1, 2, ..., l \tag{3.20}$$

Las máquinas de aprendizaje que implementan la función:

$$f(x) = sign \left(\sum_{vectoressoporte} y_i \alpha_i K(x_i, x) - b \right) \tag{3.21}$$

son llamadas, máquinas de vector soporte o de soporte vectorial, ya que su solución está basada en los vectores soporte, por tanto, la complejidad de su construcción depende del número de vectores soporte más que en la dimensión del espacio característico. El algoritmo general de una SVM es el siguiente:

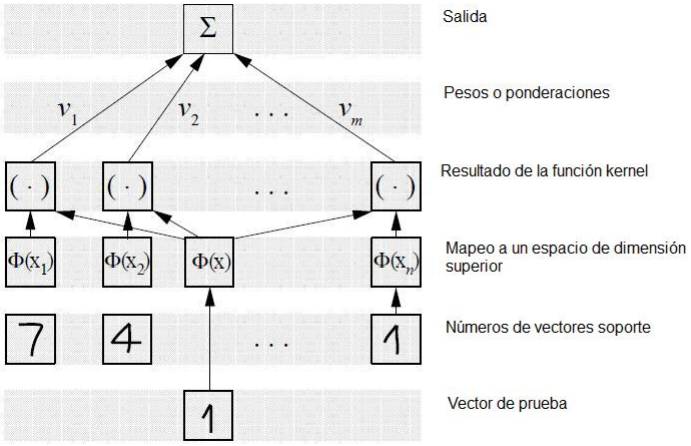

Figura 3.2: Algoritmo de una SVM

3.3. Máquinas de Soporte Vectorial en clasificación

Las máquinas de soporte vectorial, por su construcción, en un inicio fueron utilizadas en la clasficación o reconocimiento de patrones; es decir, para generar una función que

sirviera para clasificar datos o muestras provenientes de dos poblaciones distintas. En ese sentido, fueron comparadas con algunos de los métodos de clasificación ubicados en la estadística clásica que pertenecen al área de aprendizaje no supervisado.

3.3.1. Aprendizaje no supervisado

En el caso del aprendizaje no supervisado, no se tiene información respecto al número de poblaciones o clases en que se pueden clasificar los datos. Un primer recurso es partir de las diferencias entre cada una de las observaciones, de ahí que, a este tipo de aprendizaje se le conozca también como aprendizaje basado en observaciones. Una forma común de establecer la diferencia entre las observaciones, cuando los datos son de tipo numérico, es utilizando la distancia entre los vectores; de tal modo que, mientras el valor de la distancia sea menor, mayor será la similitud entre los vectores. Las distancias que se emplean con más frecuencia son la euclidiana, la de manhattan y la de mahalanovich.

De entre los algoritmos de clustering, destacan los siguientes:

k-Medidas

Partiendo de un conjunto de puntos dados en el espacio *d-dimensional* y un entero k, el algoritmo genera k o menos grupos calculando un centroide para cada grupo y reasignando cada punto al centroide más cercano, procedimiento que repite hasta que el proceso se estabiliza.
Dentro de las desventajas de este algoritmo se encuentran las siguientes:

- El número de grupos (parámetro k) debe ser elegido de antemano, o tratar con varios valores de este.
- Los datos deben ser de tipo numérico y se deben comparar utilizando la distancia euclidiana.
- El algoritno funciona adecuadamente con datos que contienen grupos esféricos, con otras geometrías, se puede dar el caso de que el algoritmo no funcione.
- El algoritmo es sensible a los *outliers* o datos que no pertenecen a ningún grupo, provocando que los grupos se desagreguen y el algoritmo se vuelva inestable.

Agrupamiento Jerárquico

Este algoritmo toma como datos de entrada un conjunto de puntos y crea un árbol en el cual los puntos son las hojas y los nodos internos revelan la estructura de similitud de los puntos. El algoritmo coloca todos los puntos en sus grupos correspondientes y mientras existan más de un grupo, fusiona pares de grupos más cercanos. De esta forma, el funcionamiento del algoritmo depende de qué tan cercanos se definen los pares de grupos. Las desventajas que presenta este algoritmo, son las siguientes:

- Si dos puntos de grupos disjuntos se encuentran cercanos, la distinción entre los dos grupos se puede perder.
- Realmente no produce un grupo, el usuario debe decidir en que parte del árbol crear los grupos.
- Es sensible a outliers.

3.3.2. Aprendizaje supervisado

El aprendizaje supervisado se basa en conocimiento previo de los datos; es decir, se conoce el número de poblaciones en que se pueden clasificar los datos y se tienen bien localizados los datos que pertenecen a cada una de ellas, es por esto que se conoce también como aprendizaje basado en ejemplos.

Dentro de los algoritmos que se utilizan para realizar este procedimiento, se encuentran los algoritmos basados en el Kernel, es decir que utilizan un proyección a un espacio de dimensión superior, como es el caso de las ANN y las SVM.

3.4. Solución del problema de programación cuadrática

El problema de programación cuadrática que se plantea en la ecuación 3.19 sujeto a las restricciones 3.20, es el que determina los parámetros de las máquinas de vector soporte, depende del número de estos vectores que se encuentran en el problema y de acuerdo a la definición del problema, no existe pérdida si el número de vectores soporte disminuye. En este sentido, Anguita et al. (2011) proponen la disminución del número de vectores soporte para su mejor tratamiento, sin embargo, a pesar de esta reducción, se debe resolver un problema de programación cuadrática.

En las primeras implementaciones de las SVM, se utilizó un *software* especializa-

do en el tratamiento de este tipo de problemas. En Smola y Scholkopf (2003), se muestra un comparativo de los métodos existentes para resolver este problema.

3.4.1. Método de chunking

En el planteamiento de las SVM, Vapnik (1995) describe el método de Chunking; este método implementa la característica de que el valor de la forma cuadrática permanece invariante si se eliminan las columnas o filas que corresponden a un valor nulo de los multiplicadores de lagrange que intervienen en el problema. De esta forma, se reduce la dimensión de la matriz a operar. Además el problema general se puede dividir en una serie de subproblemas, de forma tal que sea posible identificar los multiplicadores de Lagrange que no son cero y descartar todos los que tienen un valor nulo.

El algoritmo de Chunking, entonces, se emplea para identificar aquellas filas o columnas que pueden ser eliminadas del problema general; en cada paso, el algoritmo selecciona las M peores muestras que violan las condiciones de Karush Kuhn Tucker (KKT). Si existen menos de M muestras que violan la condiciones de KKT en un paso, entonces todas las muestras son añadidas al problema. Cada problema es inicializado con los resultados de los subproblemas previos. Al último paso, se han identificado todos los multiplicadores de Lagrange que son cero y con ese resultado se resuelve el problema de programación cuadrática 3.19.

El método de Chunking reduce la dimensión del problema 3.19 y se garantiza que el problema reducido converge; sin embargo, para el caso de problemas de gran dimensión, estos siguen siendo intratables computacionalmente, aún con la reducción propuesta por este método.

3.4.2. Método de Osuna

Osuna et al. (1997) implementa un algoritmo de descomposición, mediante el cual, es posible reducir el problema original en un número n de subproblemas; de modo que, mientras al menos una muestra que viola las condiciones KKT es añadido a las muestras del subproblema previo, se reduce la función objetivo original y se mantiene el punto factible que satisface todas las condiciones. Más aún, una secuencia de subproblemas que siempre agrega al menos una muestra que no satisface las condiciones KKT, se garantiza que converje.

En la implementación del algoritmo, Osuna et al. sugieren una matriz de tamaño constante para cada uno de los sub-problemas, lo cual impllica que se agregue y elimine el mismo número de muestras en cada paso. Otra característica de utilizar el tamaño de matriz fijo, es que se permite la dimensión arbitraria en el almacenamiento

de los datos de entrenamiento. El algoritmo no determina el número de sub-problemas que se pueden formular, y los investigadores lo fijan de manera heurística. Una de las desventajas de este método es que los subproblemas se resuelven computacionalmente, utilizando algún software adecuado para realizar análisis numérico.

3.4.3. Optimización Mínima Secuencial

El algoritmo de Optimización Mínima Secuencial (SMO) emplea el teorema demostrado por Osuna, reduciendo a dos el número de sub-problemas que se pueden resolver en cada paso. La ventaja de este método radica principalmente en que, al ser dos los sub-problemas que se analizan en cada paso, se puede resolver analíticamente, evitando completamente la optimización numérica. Adicionalmente a esto, no es necesario el almacenamiento matricial extra, además del empleado para almacenar cada uno de los subproblemas.

Debido a que son solamente dos sub-problemas y a las restricciones de desigualdad, los multiplicadores de Lagrange se deben localizar en una caja. La restricción de igualdad provoca que estos multiplicadores se localicen en un línea diagonal; entonces en cada paso, el algoritmo debe encontrar un óptimo de la función objetivo en una línea diagonal. (ver la figura 3.3)

Para resolver los dos multiplicadores de Lagrange, el algoritmo calcula en primer

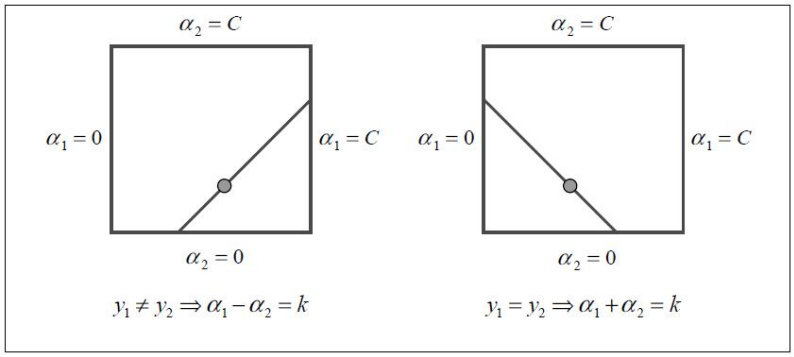

Figura 3.3: Ubicación de los multiplicadores de Lagrange.

lugar las restricciones de esos multiplicadores y después calcula las del mínimo restringido y obtiene α_1.

El algoritmo calcula el segundo multiplicador de Lagrange α_2 y entonces calcula el final del segmento de línea en términos de α_2. Al ser datos conocidos, se conoce el valor de la variable dependiente o de respuesta y_i para toda x_i, en este caso, si el valor de la respuesta Y_1 no es igual al valor de la respuesta de Y_2, entonces se aplica

el siguiente acotamiento para α_2:

$$L = max(0, \alpha_2 - \alpha_1), H = min(C, C + \alpha_2 - \alpha_1) \tag{3.22}$$

Si se da el caso de que la respuesta y_1 es igual a la respuesta y_2, entonces los siguientes límites son aplicados a α_2:

$$L = max(0, \alpha_2 + \alpha_1 - C), H = min(C, \alpha_2 + \alpha_1) \tag{3.23}$$

donde C se representa al multiplicador de Lagrange que se busca minimizar analíticamente.

La segunda derivada de la función objetivo sobre la linea diagonal, puede ser expresada en términos de:

$$\eta = K(\hat{x}_1, \hat{x}_1) + K(\hat{x}_2, \hat{x}_2) - 2K(\hat{x}_1, \hat{x}_2) \tag{3.24}$$

Bajo circunstancias normales, la función objetivo será positiva definida, este será un mínimo sobre la dirección de la restricción lineal de igualdad, y η será mayor que cero. En este caso, el algoritmo SMO calcula el mínimo sobre la dirección de la restricción.

$$\alpha_2^{new} = \alpha_2 + \frac{y_2(E_1 - E_2}{\eta} \tag{3.25}$$

donde, $E_i = u_i - y_i$ es el error de la i-ésima muestra de entrenamiento. En el siguiente paso, la restricción mínima se encuentra recortando el mínimo no restringido al final del segmento de línea.

$$\alpha_2^{new,recortado} = \begin{cases} H \text{ si } \alpha_2^{new} \geq H \\ \alpha_2^{new} \text{ si } L < \alpha_2^{new} < H \\ L \text{ si } \alpha_2^{new} \\ leqL \end{cases}$$

Sea $s = y_1 y_2$. El valor de α_1 es calculado del α_2 nuevo y recortado

$$\alpha_1^{new} = \alpha_1 + s(\alpha_2 - \alpha_2^{new,recortado} \tag{3.26}$$

Bajo circunstnacias inusuales, η será negativo. Esto ocurre cuando el Kernel no satisface las condiciones de Mercer, lo cual puede ocasionar que la función objetivo llegue a ser indefinida.

Se da el caso de que $\eta = 0$ incluso con un Kernel correcto, cuando más de una muestra o vector de entrada x de entrenamiento tiene el mismo valor para su variable de respuesta. El algoritmo SMO funciona adecuadamente, incluso para $\eta < 0$, en cuyo

caso la función objetivo ψ debe ser evaluada en cada segmento de línea.

$f_1 = y_1(E_1 + b) - \alpha_1 K(\vec{x_1}, \vec{x_1}) s \alpha_2 K(\vec{x_1}, \vec{x_2})$
$f_2 = y_2(E_2 + b) - \alpha_1 K(\vec{x_1}, \vec{x_2}) s \alpha_2 K(\vec{x_2}, \vec{x_2})$
$L_1 = \alpha_1 + s(\alpha_2 - L),$
$H_1 = \alpha_1 + s(\alpha_2 - H),$
$\psi_L = L_1 f_1 + L f_2 + \frac{1}{2} L_1^2 K(\vec{x_1}, \vec{x_1}) + \frac{1}{2} L^2 K(\vec{x_2}, \vec{x_2}) + s L L_1 K(\vec{x_1}, \vec{x_2})$
$\psi_H = H_1 f_1 + H f_2 + \frac{1}{2} H_1^2 K(\vec{x_1}, \vec{x_1}) + \frac{1}{2} H^2 K(\vec{x_2}, \vec{x_2}) + s H H_1 K(\vec{x_1}, \vec{x_2})$

El algoritmo SMO mueve los multiplicadores de Lagrange al final del punto que tiene el valor más pequeño de la función objetivo. Si la función objetivo es la misma al final de ambas líneas, y el Kernel satisface las condiciones de Mercer, entonces el algoritmo no tiene progreso alguno. En esas situaciones, se debe seleccionar otro multiplicador para ser optimizado.

La selección de los multiplicadores que continuarán el proceso de optimización se basa en una heurística, en la que se consideran las muestras que no satisfacen las condiciones de KKT con un ϵ determinado, de forma tal que primero se calculan aquellos multiplicadores que se encuentran más alejados de ese valor ϵ.

El umbral o la constante b se calcula en cada paso, de tal forma que las condiciones de KKT son satisfechas por ambas muestras optimizadas. El valor siguiente para b_1 es válido cuando la nueva α_1 no está en los límites, debido a que se fuerza la salida de la SVM a estar en y_1 cuando la entrada es x_1

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1) K(\vec{x_1}, \vec{x_1}) + y_1(\alpha_2^{new,recordado} - \alpha_2) K(\vec{x_2}, \vec{x_1}) + b \qquad (3.27)$$

El umbral b_2 es válido cuando el nuevo α_2 no está en los límites, debido a que se fuerza la salida de la SVM a que sea y_2 cuando la entrada es x_2

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1) K(\vec{x_1}, \vec{x_2}) + y_2(\alpha_2^{new,recordado} - \alpha_2) K(\vec{x_2}, \vec{x_2}) + b \qquad (3.28)$$

Cuando se da el caso de que b_1 y b_2 son válidos, son iguales. Cuando ambos multiplicadores están en un límite y L no es igual al H, entonces en el intervalor entre b_1 y b_2 está todo el umbral que satisface las condiciones de KKT y el algoritmo SMO elige la media de esos dos valores.

Para calcular una SVM lineal, se necesita registrar solamente un vector para $\vec{w}$ al menos que todas las muestras de entrenamiento correspondan a un multiplicador de Lagrange con valor diferente de nulo. Si la optimización conjunta tiene éxito, el vector de peso registrado necesita ser actualizado para reflejar el valor del nuevo multiplicador de Lagrange. Debido a la linealidad de la SVM, la actualización del vector de peso es como se muestra:

$$\overrightarrow{w}^{new} = \overrightarrow{w} + y_1(\alpha_1^{new} - \alpha_1)\,\overrightarrow{x_1} + y_2(\alpha_2^{new,recortado} - \alpha_2)\,\overrightarrow{x_2} \tag{3.29}$$

Es importante mencionar que el algoritmo SMO, por su efectividad, ha sido implementado en varias de las librerias o programas que implementan a las SVM.

La finalidad de los algoritmos mostrados, es reducir la dimensionalidad de la matriz con la que se opera en el problema de programación cuadrática, en la figura 3.4 se muestra el comparativo de los tres algoritmos, el tamaño de los bloques representa la dimensión de la matriz con la que se calculan los resultados.

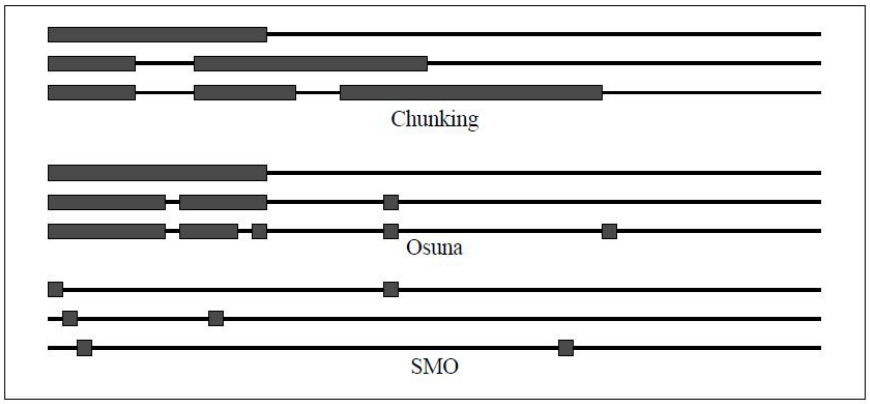

Figura 3.4: Desempeño de los métodos de Chunking, Osuna y SMO.

3.5. La función Kernel

La función que se utiliza para hacer la proyección de datos que se encuentran en un espacio característico a un espacio representado en una dimensión superior, se conoce como *Kernel*. La función Kernel debe satisfacer las condiciones de Mercer, y debe ser una función simétrica.

De las funciones que se han utilizado como kernel, resaltan por su importancia, las siguientes:

- El kernel lineal implementado como la función más simple:
  $k(x, x') = \langle x, x' \rangle$

- El Kernel Gausiano de Función de Base Radial (RBF):
  $k(x, x') = exp(-\sigma||x - x'^2||)$
- El kernel polinomial:
  $k(x, x') = (scale. \langle x, x' \rangle + offset)^{grado}$
- El kernel tangente hiperbólico:
  $k(x, x') = tanh(scale. \langle x, x' \rangle + offset)$
- El kernel de función de Bessel del primer tipo:
  $k(x, x') = \frac{Bessel^n_{v+1}(\sigma||x-x'||)}{(||x-x'||)^{-n(v+1)}}$
- El kernel de función de base radial de Laplace:
  $k(x, x') = exp(-\sigma||x - x'||)$
- El kernel radial de base ANOVA:
  $k(x, x') = \left(\sum_{k=1}^{n} exp(-\sigma(x^k - x'^k)^2) \right)^d$
- el kernel de esplines lineales en una dimensión:
  $k(x, x') = 1 + xx'min(x, x') - \frac{x+x'}{2} \left(min(x, x')^2 + \frac{min(x,x'^3)}{3} \right)$
  y para el caso multidimensional $k(x, x') = \prod_{k=1}^{n} (x^k, x'^k)$

El kernel de función de base radial gausiano y de Laplace, son kernels de propósito general utilizados cuando no hay conocimiento *a priori* acerca de los datos. El kernel lineal es útil cuando se trabaja con vectores de datos muy escasos, como en la categorización de texto. El kernel polinomial es porpular en el área de procesamiento de imágenes y el caso del kernel sigmoide es utilizado como *proxi* para redes neuronales. Los esplines y kernel de función de base radial ANOVA tienen buen desempeño en problemas de regresión. (Karatzoglou et al., 2006).

De acuerdo a las implementaciones que se han realizado de las SVM, se ha reportado que, para el caso de clasificación y regresión, el kernel de función de base radial RBF ha sido el que mejor ajusta.

3.6. Implementación de SVM en R

Las máquinas de soporte vectorial se han implementado en varios lenguajes de programación, algunos emplean librerias o funciones provenientes de algún *software* especializado para resolver el problema de programación cuadrática que se desprende

de su planteamiento. (Meyer, 2011)

En R se han implementado 4 paquetes que incluyen librerias para calibrar las SVM para clasificación y regresión y estimación de densidades:

- e1071.
- kernlab
- klaR
- svmpath

Karazoglou et al. (2006), presentan un estudio comparativo entre los paquetes que implementan las SVM en R utilizando bases de datos para los procesos de clasificación y regresión.

Dentro de las características que resaltan del paquete e1071, que es el que más se emplea en regresión, son las siguientes:

- Es aplicable en procedimientos de clasificación, regresión y estimación de densidades.
- Implementa 4 kernels, Gausiano, polinomial, lineal y sigmoide.
- Utiliza el algoritmo SMO para resolver el problema de programación cuadrática.
- Selecciona el modelo (calibración de la máquina) utilizando una función de búsqueda cruzada para los parámetros.
- Almacena los datos en fórmulas, matrices y matrices poco densas.
- Utiliza interfaces en archivos implementados en lenguaje C y utiliza un sistema de clases $S3$.
- Proporciona herramientas de graficado y precisión mediante el comando *tune*.

Capítulo 4

Materiales y métodos

4.1. Datos de evapotranspiración de referencia

De acuerdo a González y coautores (2008), los datos climáticos se obtuvieron de la base de datos de la red agroclimática "Valle del Fuerte", ubicada en el distrito de riego 075 en la ciudad de los Mochis Sinaloa, localizada a una altitud de $25°48.89'$ N, longitud de $109°1.53'$, y a una altitud de 20 metros.

La base de datos contiene registros observados en el periodo comprendido de febrero de 1997 a junio 2001. Como no es posible disponer de todas las variables climáticas, por practicidad y sin pérdida de una precisión significativa, se consideran solamente las siguientes:

- La temperatura media del aire (T) en $°C$, calculada a partir de las temperaturas máxima y mínima $T = ((T_{max} + T_{min})/2)$.
- La radiación solar (Rg) medida en $\frac{MG}{m^2 d}$.
- La humedad relativa del aire calculada, en su valor porcentual, a partir de sus valores máximo y mínimo diario $Hr = ((Hr_{max} + Hr_{min})/2)$.
- La velocidad del viento (U_2) en $\frac{m}{s}$, observada a una altura de 2 metros.

A partir de las variables anteriores, se calcula el valor de la evapotranspiración de referencia utilizando una de las ecuaciones más sencillas de *Penman-Monteith*(Allen et al., 1998; Guevara, 2006).

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T+273} U_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34U_2)} \tag{4.1}$$

de la ecuación anterior:

- Δ es la pendiente de la curva de vapor saturado $\frac{kPa}{^oC}$
- R_n es la radiación neta sobre la superficie del cultivo $\frac{MJ}{m^2 dia}$
- G es el flujo calórico utilizado en el calentamiento del suelo $\frac{MJ}{m^2 dia}$
- $(R_n - G)$ es la energía disponible el la superficie del suelo e igual a la superficie desde el aire $(H + 1E)$ por el calor sensible H (convección) y el calor latente $1E$ (evaporación).
- γ es la constante psicrométrica $\frac{kPa}{^oC}$
- $(e_s - e_a)$ es el déficit de la tensión de vapor (kPa), e_s y e_a son la tensión de vapor saturado y la tensión actual respectivamente.
- es el calor latente de vaporización en (MJ/kg), el cual se considera fijo a 20^oC, con un valor constante de $2.45\frac{MJ}{Kg}$

Empleando la ecuación 4.1, es como se tiene el registro diario de evapotranspiración de referencia para el periodo en el cual se observaron los datos.

4.2. Máquinas de Soporte Vectorial en regresión

El concepto fundamental de las SVM es mapear el conjunto de datos originales X en un espacio característico F con una alta dimensionalidad a través de una función de mapeo no lineal o función kernel y construir un hiperplano de separación óptimo en ese espacio.

4.2.1. Fundamentos del método

Dado un conjunto de datos de entrenamiento o de datos que son conocidos de manera *a priori* $\{(x_1, y_1), ..., (x_n, y_n)\} \subset X \times \Re$ donde X denota el espacio de los patrones de entrada ubicados en una cierta dimensión d; es decir $X = \Re^d$. El objetivo de la $\epsilon-$regresión es encontrar una función $f(x)$ que se desvía a lo más un valor ϵ de los valores de las observaciones y_i para todos los datos contenidos en el conjunto de entrenamiento, y que a la vez esa función sea tan plana como sea posible. Esto quiere decir que el error no es significativo mientras sea menor que un valor de ϵ y que en este mismo sentido, no se permitirá ninguna desviación mayor a este valor ϵ.

Para el caso más simple, que corresponde a una función lineal, la función $f(x)$ buscada en el proceso de $\epsilon-$regresión, debe tener la forma:

$$f(x) = \langle w, x \rangle + b \tag{4.2}$$

con $w \in X$, $b \in \Re$, y en donde $\langle ., . \rangle$ denota el producto interno en X.

De acuerdo a la definición, se está interesado en una función que sea tan plana como sea posible, esto significa que se está buscando el valor más pequeño para w. Una forma de asegurar que el valor de w sea mínimo, es minimizar su norma $||w||^2 = \langle w, w \rangle$; lo cual conduce a un problema de optimización convexa:

$$\text{minimizar } \tfrac{1}{2}||w||^2$$

$$\text{sujeto a } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases}$$

Del problema de optimización convexa planteado, se trabaja bajo el supuesto de que tiene una solución factible; en situaciones en las cuales no sea posible esa factibilidad o en los casos en que se necesite ampliar el margen de error ϵ, de acuerdo a Bennett y Mangasarian, 1992, es posible introducir variables de holgura ξ_i, ξ_i^* para dar tratamiento a las restricciones de no factibilidad del problema de optimización. El problema de minimización original propuesto por Vapnik (1995) se formula de la forma siguiente:

$$\text{minimizar } \tfrac{1}{2}||w||^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)$$

$$\text{sujeto a } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i + \xi_i^* \geq 0 \end{cases}$$

con $i = 1, ..., l$

En donde el valor de la constante $C > 0$, determina el equilibrio entre lo plano de la función f y el margen para el cual las desviaciones mayores que el margen de error ϵ serán toleradas. Esto equivale a operar la función de pérdida ϵ-sensible, definida como:

$$|\xi|_\epsilon := \begin{cases} 0 \text{ si } |\xi| \leq \epsilon \\ |\xi| - \epsilon \text{ d.o.m} \end{cases}$$

El problema de programación cuadrático planteado, por su complejidad, es resuelto

en su formulación dual equivalente. Una de las ventajas de la formulación dual del problema, es que proporciona las herramientas necesarias para extender las Máquinas de Soporte Vectorial a funciones no lineales.

De acuerdo a Fletcher (1989), para resolver el problema dual de programación cuadrática, se debe iniciar con la construcción de una función lagragiana del problema primal y su respectivo conjunto de restricciones, agregando un conjunto de variables duales; este planteamiento, de acuerdo a Vandervei (1997) conlleva a la obtención de una solución única del problema dual, en un punto de silla.

El lagrangiano del problema dual, se formula de la siguiente manera:

$$\begin{aligned} L := \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i\xi_i^*) \\ - \sum_{i=1}^{l}\alpha_i(\epsilon + \xi_i - y_i + \langle w, x_i\rangle + b) \\ - \sum_{i=1}^{l}\alpha_i^*(\epsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) \end{aligned} \tag{4.3}$$

En donde L es el lagrangiano y $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ son los respectivos multiplicadores de Lagrange.

Debido a que las variables duales deben de satisfacer la restricción de positividad, entonces:

$$\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \geq 0 \tag{4.4}$$

De acuerdo a la condición del punto de silla, las derivadas parciales de L respecto de las variables primales (w, b, ξ_i, ξ_i^*) deben desaparecer por optimalidad, entonces:

$$\partial_b L = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{4.5}$$

$$\partial_w L = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i = 0 \tag{4.6}$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \tag{4.7}$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0 \tag{4.8}$$

Sustituyendo las derivadas parciales en el lagrangiano del problema dual de optimización L, se tiene:

$$\text{maximizar} \begin{cases} -\frac{1}{2} \sum_{j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\epsilon \sum_{j=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{j=1}^{l} y_i (\alpha_i - \alpha_i^*) \end{cases}$$

$$\text{sujeto a: } \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0 \text{ y } \alpha_i, \alpha_i^* \in [0, C]$$

Es importante señalar que en el problema anterior, se eliminan las variables duales η_i y η_i^* ya que sus derivadas parciales se igualan a cero. De esta forma, las condiciones 4.7 y 4.8, se reformulan de la forma: $\eta_i = C - \alpha_i$ y $\eta_i^* = C - \alpha_i^*$

Con esta modificación, la ecuación 4.6, se reescribe como:

$$w = \sum_{i=1}^{l} (\alpha_i, \alpha_i^*) x_i \tag{4.9}$$

entonces

$$f(x) = \sum_{i=1}^{l} (\alpha_i, \alpha_i^*) \langle x_i, x \rangle + b \tag{4.10}$$

Lo cual se conoce como la expansión de los vectores soporte, y puede visualizarse como una combinación lineal de los datos de entrenamiento; en este sentido, la complejidad de la función no depende de la dimensión del espacio de entrada X, y depende únicamente de los vectores de soporte, que son los vectores que se ubican fuera del espacio determinado por margen de error ϵ.

El algoritno, en su totalidad, puede ser operado a partir del producto punto entre los datos, para el caso del valor de w no es necesario que se calcule explícitamente, incluso cuando se está evaluando $f(x)$, es por esta razón que no importa la dimensión de los datos.

Otro valor de importancia, es el valor del umbral o la constante b, para calcularlo se utilizan las condiciones de Karush Kuhn Tucker (KKT), las cuales establecen

que en el punto solución del problema, el producto entre las variables duales y las constantes son iguales a cero. De esta forma:

$$\begin{aligned} \alpha_i(\epsilon + \xi_i - y_i + \langle w, x_i\rangle + b) = 0 \\ \alpha_i^*(\epsilon + \xi_i^* + y_i - \langle w, x_i\rangle - b) = 0 \end{aligned} \tag{4.11}$$

$$\begin{aligned} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0 \end{aligned} \tag{4.12}$$

Lo anterior permite hacer varias conclusiones de utilidad.

- Únicamente las muestras (x_i, y_i) con su correspondiente $\alpha_i, \alpha_i^* = C$ permanecen fuera de la región acotada por el margen de error ϵ.
- Si el producto $\alpha_i\alpha_i^* = 0$, no es posible que se tenga un conjunto de variables duales α_i, α_i^* que sean iguales a cero de manera simultánea, con lo cual se puede concluir que:

  $\epsilon - y_i + \langle w, x_i\rangle + b \geq 0 \quad$ y $\quad \xi_i = 0 \quad$ si $\quad \alpha_i < C$
  $\epsilon - y_i + \langle w, x_i\rangle + b \leq 0 \quad$ si $\quad \alpha_i > 0$

  Un análisis similar se hace para α_i^*, conjugando los resultados se tiene:

  $$max\left\{-\epsilon + y_i - \langle w, x_i\rangle \,|\alpha_i < C \vee \alpha_i^* > 0\right\} \leq b \leq min\left\{-\epsilon + y_i - \langle w, x_i\rangle \,|\alpha_i > 0 \vee \alpha_i^* < C\right\}$$

Una observación importante es que si $|f(x_i) - y_i| \geq \epsilon$ los multiplicadores de Lagrange deben ser diferentes de cero, lo cual significa que para todas las observaciones que se encuentran dentro de la región delimitada por el margen de error ϵ, los valores α_i y α_i^* deben ser iguales a cero. Cuando $|f(x_i) - y_i| < \epsilon$ los valores de α_i y α_i^* deben ser iguales a cero para que se cumplan las condiciones de KKT, por lo tanto, el valor de w se calcula a partir de aquellas observaciones de la forma $|f(x_i) - y_i| \geq \epsilon$, conocidos como *vectores de soporte*.

La formulación dual del problema de programación cuadrática, para el caso lineal, se representa de acuerdo a la figura siguiente:

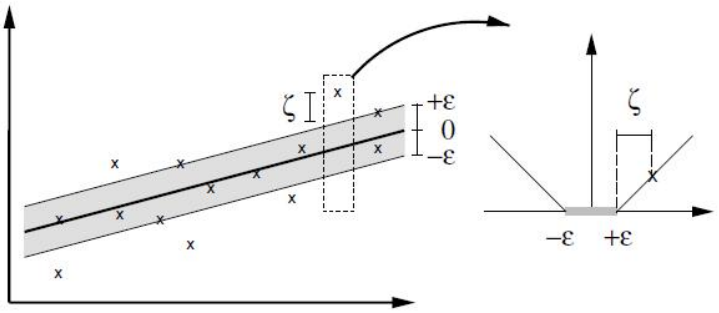

Figura 4.1: Margen suave de la función de pérdida para una SVM lineal.

4.2.2. La ventana móvil

Las SVM implementan una ventana de tamaño n para poder utilizar regresión en un conjunto pequeño de observaciones, de esta forma es que es posible generar, por muestreo, un modelo para poder predecirlo (Müller at al., 1998). El tamaño de ventana depende, en la mayoría de los casos, de la naturaleza de los datos. Para fijar el tamaño de la ventana móvil se realizan varias pruebas con tamaños de 3 a 10 y de ahí se selecciona el valor más adecuado (Lu y Wang, 2005) Por su parte, Bo-Juen et al. (2001) demuestran que para el problema particular del consumo de energía eleéctrica, una ventana de tamaño 7 permite obtener un modelo más preciso. Por otra parte Radhika y Sashi (2009) hacen referencia a que una ventana de tamaño 5, para la predicción de temperatura atmosférica, proporciona una muy buena precisión. Partiendo de sta información, es conveniente implementar inicialmente una ventana de tamaño 5 para la estimación de los parámetros libres de la SVM.

4.2.3. Estimación de los parámetros libres

Es bien sabido que el proceso de generalización o la precisión al momento de predecir de las SVM depende del establecimiento de los parámetros libres C y ϵ, así como los parámetros de la función kernel que se utilice para proyectar el problema a un espacio característico en una dimensión superior, en este caso se utiliza una función de base radial (RBF). Por una parte, el parámetro C determina la compensación entre la complejidad del modelo y el grado en el cual las desviaciones son mayores que el valor ϵ tolerado en el problema de optimización que modela las SVM, de esta forma, si C es demasiado grande, entonces se estará minimizando solamente el riesgo empírico sin tomar en cuenta la parte de la complejidad en la formulación del problema de optimización. El parámetro ϵ controla el ancho de la zona $\epsilon-$sensible utilizada para ajustar los datos de entrenamiento, entonces el valor de ϵ afecta directamente el

número de vectores soporte utilizados en la construcción de la función de regresión, de esta forma un valor pequeño para ϵ provoca que el número de vectores soporte se reduzca y un valor grande de ϵ provoca una estimación más plana.

Para el establecimiento del parámetro C se utilizó la metodología planteada por Cherkassky y Ma (2004); de acuerdo a su reporte, la elección óptima del parámetro de regularización C se deriva de la parametrización estándar de la solución de la SVM dada de la forma:

$$\begin{aligned} |f(x)| &\leq |\sum_{i=1}^{l} (\alpha_i - \alpha_i^* K(x_i, x)| \\ &\leq \sum_{i=1}^{l} |(\alpha_i - \alpha_i^*)| \cdot |K(x_i, x)| \\ &\leq \sum_{i=1}^{l} C \cdot |K(x_i, x)| \end{aligned} \tag{4.13}$$

Utilizando la función de base radial como la función kernel, se tiene:

$$K(x_i, x) = exp\left(-\frac{||x - x_i||^2}{2\sigma^2}\right) \tag{4.14}$$

tal que $K(x_i, x) \leq 1$, de ahí se obtiene el siguiente límite superior de la SVM para regresión:

$$|f(x)| \leq C \cdot l \tag{4.15}$$

donde l es el número de vectores soporte.

La ecuación 4.15 establece la relación entre el parámetro de regularización C y el número de vectores soporte, para un valor dado ϵ. Sin embargo, se debe observar que el número de vectores soporte está en función del valor del parámetro ϵ. Para determinar el valor de C cuando se desconoce el número de vectores soporte, se puede tener la desigualdad $C \geq |f(x)|$ para todas las muestras de entrenamiento, con lo cual se puede establecer el valor de C como el rango de valores de respuesta de los datos de entrenamiento, sin embargo este resultado es un tanto sensible a la presencia de *outliers*, entonces se propone utilizar el siguiente valor:

$$C = max\left(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|\right) \tag{4.16}$$

donde $\bar{y}$ y σ_y son la media y desviación estándar respectivas del conjunto de datos de entrenamiento.

El valor de ϵ debe ser proporcional al nivel de ruido de los datos de entrada $\epsilon \propto \sigma$, asumiendo que la desviación estándar σ del ruido es conocido o puede ser calculado a partir de los datos. Sin embargo, la elección de ϵ debe depender del número de los

datos de entrenamiento, de la teoría estadística de la regresión lineal:

$$\sigma_{y/x}^2 \propto \frac{\sigma^2}{n} \tag{4.17}$$

Se sugiere entonces la siguiente fórmula para el cálculo de ϵ:

$$\epsilon \propto \frac{\sigma}{\sqrt{n}} \tag{4.18}$$

El cálculo anterior resulta razonable para el caso de muestras pequeñas, sin embargo para muestras grandes el valor de ϵ es demasiado pequeño provocando que se tenga un número reducido de vectores de soporte, por lo que se propone la siguiente expresión empírica para la estimación de ϵ

$$\epsilon = \tau\sigma\sqrt{\frac{ln(n)}{n}} \tag{4.19}$$

De acuerdo a una aproximación empírica, un valor de $\tau = 3$ proporciona un buen desempeño para datos de diferentes tamaños (Cherkassky y Ma, 2004).

4.3. Redes Neuronales Artificiales

Las redes neuronales son una herramienta de la inteligencia artificial que busca emular el funcionamiento del cerebro humano y bajo este esquema, se utilizan en la implementación de procesos de aprendizaje, en base a unidades conocidas como neuronas. La forma en que se estructuran las neuronas en una red, tiene un comportamiento similar con la forma en que se genera el conocimiento en el cerebro humano; en redes neuronales humanas, cuando una neurona reacciona neurotransmite un estímulo al siguiente grupo de neuronas en una reacción en cadena, de esta forma se transmite un mensaje entre diferentes grupos de neuronas, este mensaje puede tener tres formas diferentes:

- *Excitación.* Neurotransmisiones de estímulos incrementan el estado de excitación en reacciones en cadena.
- *Inhibición.* Neurotransmisiones de inhibición, decrementan el estado de excitación en reacciones en cadena.
- *Potenciación.* Este concepto se refiere al ajuste de la sensibilidad del siguiente grupo de neuronas, ya sea en el estado de excitación o inhibición, esto es equivalente al proceso de aprendizaje.

De lo anterior, se pueden visualizar las neuronas como interruptores simples, entonces, se pueden modelar las conecciones entre neuronas como matrices de números conocidos como pesos, de tal forma que los pesos positivos indican excitación y los pesos negativos indican inhibición, y la forma en que se modela el aprendizaje depende del paradigma utilizado.

Las redes neuronales artificiales, no pueden modelar sistemas tan complejos como el cerebro humano, sin embargo pueden consistir de redes de a lo más unos cientos o miles de neuronas con un número limitado de conexiones entre ellas. Por lo general, estas redes neuronales son dispositivos básicos de entrada y salida con las neuronas organizadas en capas; de esta forma un perceptron simple o una red neuronal simple, consiste de una capa de neuronas de entrada que corresponen a una capa de neuronas de salida, con una capa simple de pesos entre ellas.

En la forma más simple, el proceso de aprendizaje consiste en encontrar los valores correctos para los pesos entre la capa de entrada de salida, el esquema más elemental de una red neuronal es la que se muestra en la figura ??.

Matemáticamente, para el caso más simple de red neuronal, las entradas (I) y salidas

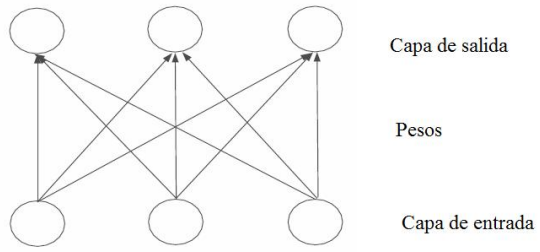

Figura 4.2: Estructura de una red neuronal simple

(O) corresponden a vectores y por su parte, los pesos a una estructura matricial. La salida de la red se calcula utilizando la siguiente expresión.

$$O = f(IW_{io})$$

De acuerdo a la ecuación, los datos son presentados en la capa de entrada, la red procesa la entrada multiplicándola por la matriz de pesos, el resultado de esta multiplicación es procesada por la capa de salida utilizando una función que determina los nodos que son activados.

El proceso de encontrar los valores correctos para los pesos, como ya se ha mencinado es el proceso de aprendizaje, una forma de establecer este proceso es partiendo de una matriz de ceros y unos e ir cambiando estos valores hasta encontrar los pesos correctos que solucionan un problema particular. La forma en que se encuentran estos valores,

es partiendo de datos conocidos, por lo tanto se dice que implementan la técnica de aprendizaje supervisado o a partir de ejemplos.

En la determinación de los pesos correctos, se busca minimizar la diferencia entre las respuestas generadas por la red neuronal y los valores de las observaciones que se utilizan como entrenamiento, por lo que se plantea un problema de minimización que se resuelve por el método del gradiente ascendiente. En base a lo anterior, podemos concluir que las redes neuronales implementan la estrategia de mantener un intervalo de confianza fijo y minimizar el riesgo empírico.

El método que se implementa más comunmente en el proceso de aprendizaje, es el método de propagación hacia atrás, este método contiene tres elementos:

- paso hacia adelante.

  $x_i(k) = S\{w(k)x_i(k-1)\}$, $i = 1, ..., l$ y $k = 1, ..., m$

  con las condiciones de límite

  $x_i(0) = x_i$, $i = 1, ..., l$

- paso hacia atrás.

  $b_i(k) = w^T(k+1)\nabla S\{w(k+1)x_i(k)\}\, b_i(k+1)$ para $i = 1, ..., l$, $k = 1, ..., m-1$

  Con las condiciones de límite

  $b_i(m) = 2(y_i - x_i(m))$, $i = 1, ..., l$

- Modificación del peso por el peso de las matrices $w(k)$, $k = 1, 2, ..., m$

  $$w(k) \leftarrow w(k) - \gamma(.) \sum_{i=1}^{l} \nabla S\{w(k)x_i(k-1)\}\, w(k)x_i^T(k-1)$$

El tipo de red neuronal que se implementa en la mayor parte de la literatura, con fines de comparación con otros métodos, es el de *multilayer perceptron*, esta es una clase de redes neuronales la cual consiste de un conjunto de unidades sensoriales que constituye la capa de entrada, una o más capas ocultas de nodos y una capa de salida. Una de las funciones de activación no lineal es la función sigmoide, y el procedimiento que se realiza es el siguiente:

1. Inicializar los pesos a valores aleatorios pequeños.

2. En el paso de propagación hacia atrás, cada unidad de entrada recibe una señal de entrada y transmite esta señal a cada una de las unidades ocultas. Cada unidad oculta calcula la función de activación y envía esta señal a cada unidad de salida. La unidad de salida calcula la función de activación para formular la respuesta de la red para un patrón de entrada dado.

3. En el paso de propagación del error hacia atrás, después de computar la activación con su peso, se calcula el error asociado a ese patrón y es distribuido hacia atrás a todas las unidades de la capa anterior.

4. Se hace la actualización de los pesos y sesgos y se finaliza el entrenamiento con una condición de paro, ya sea la minimización del error o el número de repeticiones.

Es importante resaltar los problemas principales, que se reportan en la literatura sobre el uso de las redes neuronales:

- La función de riesgo empírico, en el peor de los casos, tiene muchos mínimos locales. Los procedimientos de optimización estándar garantizan la convergencia de uno de ellos. La certeza de obtener una solución depende de muchos factores, principalmente de la inicialización de la matriz de pesos $w(k)$, $k = 1, ..., m$. La elección de los parámetros de inicialización para alcanzar un mínimo local pequeño, está basado en una heurística.

- La convergencia del método del gradiente es un tanto lenta; en este sentido, existen varias heurísticas para agilizar el proceso de convergencia.

- La función sigmoide empleada en el algoritmo, tiene un factor de escalamiento que afecta directamente la calidad de la aproximación encontrada. La elección del factor de escalamiento es el equilibrio entre la calidad de la aproximación y la razón de convergencia. Existen recomendaciones empíricas para la elección del factor de escalamiento.

De lo anterior, se puede decir que las redes neuronales no son unas máquinas de aprendizaje muy controladas, no obstante, en muchas aplicaciones prácticas, las redes neuronales han demostrado buenos resultados.

4.4. Método para evaluar la capacidad predictiva

Con la finalidad de corroborar la capacidad predictiva de los métodos que se están empleando, se utiliza el Error Porcentual Absoluto Medio (MAPE), el cual se calcula

como el promedio de las diferencias absolutas entre los valores pronosticados y los valores reales , para calcularlo se utiliza la expresión:

$$MAPE = 100\frac{\sum_{i=1}^{n} |\frac{L_i - \hat{L}_i}{L_i}|}{n} \tag{4.20}$$

Capítulo 5

Implementación y experimentos

En este capítulo, para dar cumplimiento a los objetivos específicos planteados en esta investigación, se presenta el procesamiento de la base de datos y la implementación de 3 métodos orientados a la predicción de series de tiempo. Se detalla sobre la implementación de las SVM por tratarse de la herramienta de más interés y se realiza un comparativo de los resultados obtenidos con un modelo ARIMA y una red neural en su versión de *multilayer perceptron* (MLP).

5.1. Algoritmo de refinamiento

Se desarrolló un algoritmo heurístico de refinamiento que resulta de la aplicación de la primera estimación de los parámetros libres de una SVM. Los pasos del algoritmo son los siguientes:

- Primero. Se establece un valor para la ventana móvil. de acuerdo a la mayoría de las implementaciones de las SVMs, un valor inicial de 5, pero si la serie de tiempo describe un comportamiento periódico relativamente corto, es preferible establecer tal periodicidad. Con el tamaño de inicial de la ventana móvil y con los valores iniciales de C y ϵ obtenidos de las fórmulas (4.16) y (4.19), se establece el parámetro del kernel σ por medio de una búsqueda heurística en el intervalo [0,1] utilizando el comando tune el cual está incluido en el lenguaje R (Karatzoglu et al. 2006).
- Segundo. Después de la primera estimación, es necesario establecer un tamaño de ventana móvil más adecuado, para esto se implementó una búsqueda binaria con un tamaño de paso de una unidad para ventanas con valores entre 3 y 10, verificando en error MAPE para determinar cuál es el mejor valor para la ventana móvil.

- Tercero, como la segunda estimación de la ventana móvil es, en teoría, la mejor aproximación, el siguiente paso es recalibrar los parámetros libres implementando nuevamente una búsqueda binaria. Este procedimiento se realiza con un tamaño de paso menor.

5.2. Implementación de las SVM

Se tiene como primer objetivo "pronosticar una serie de tiempo de evapotranspiración de referencia utilizando Máquinas de Soporte Vectorial", la implementación se hace sobre los datos de evapotranspiración de referencia (ETo) calculada a partir de los promedios diarios de temperatura, humedad relativa del aire, radiación solar y velocidad del viento. Se cuenta con un total de 1591 observaciones diarias de ETo, registradas en el periodo del 18 de febrero de 1997 al 27 de junio de 2001.

La implementación de las SVM se realiza en las siguientes etapas:

- Descripción de los datos y selección del conjunto de datos de entrenamiento y prueba.
- Establecimiento del valor inicial de los parámetros libres y del tamaño de ventana móvil.
- Estimación del modelo y pruebas.

5.2.1. Descripción de los Datos

El conjunto de datos se dividió en dos grupos de datos:

- *Datos de entrenamiento.* Los datos comprendidos del 18 de febrero de 1997 al 31 de mayo de 2001.
- *Datos de prueba.* Este conjunto de datos corresponde a 27 días del mes de junio de 2001.

El valor máximo registrado corresponde a junio de 1999, con un valor de $8.36\frac{mm}{dia}$ y el valor mínimo $0.67\frac{mm}{dia}$ se observó en septiembre de 1997. El rango de los valores registrados es de 7.67, y los valores máximo y mínimo son ambos positivos, como se puede observar en la figura (5.1). Como en la implementación de las SVM se está utilizando la estimación de parámetros libres propuesta por Cherlassky y Ma

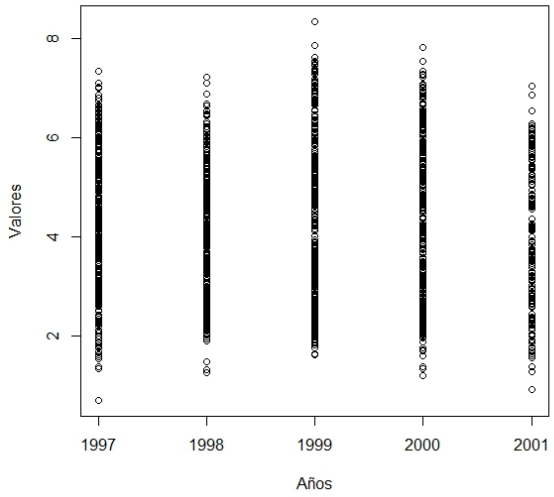

Figura 5.1: Valores mínimos y máximos de los registros de Evapotranspiración de referencia, de febrero de 1997 a junio de 2001.

(2004), es conveniente el escalamiento de los datos a valores equivalentes en un rango $[0, 1]$. Este procedimiento se realizó con la fórmula:

$$y' = \frac{y - y_{min}}{y_{max} - y_{min}} \tag{5.1}$$

Este escalamiento de los datos se aplicará también para la predicción de la serie de tiempo utilizando las ANN.

De acuerdo con Cherlassky y Ma (2004), es conveniente estandarizar los datos con media $\mu = 0$ y varianza $\sigma^2 = 1$, sin embargo esta estandarización ocasiona que haya valores con signo negativo, y esto no es conveniente para la predicción por medio de las MLP (Radhika y Sashi, 2009). En los datos de ETo intervienen 4 variables climáticas importantes y la ETo es resultado de aplicar la fórmula de Penman-Monteith FAO98, por lo que se considera como una serie de tiempo univariada. De acuerdo a la figura (5.2), los datos describen un comportamiento periódico anual, en donde los valores mínimos de evapotranspiración están registrados para los periodos de otoño e invierno y los valores máximos para los periodos de primavera y verano.

Con las SVM, es común realizar un proceso de clasificación en los datos históricos para realizar una mejor predicción (Bo-Juen, et al., 2001), de esta forma, al predecir el mes de junio, se consideraría en el modelo los registros históricos de los meses de abril a agosto; sin embargo, para hacer la comparación en las mismas condiciones con los otros métodos, se consideraron todos los datos para establecer el modelo, sin hacer esta separación.

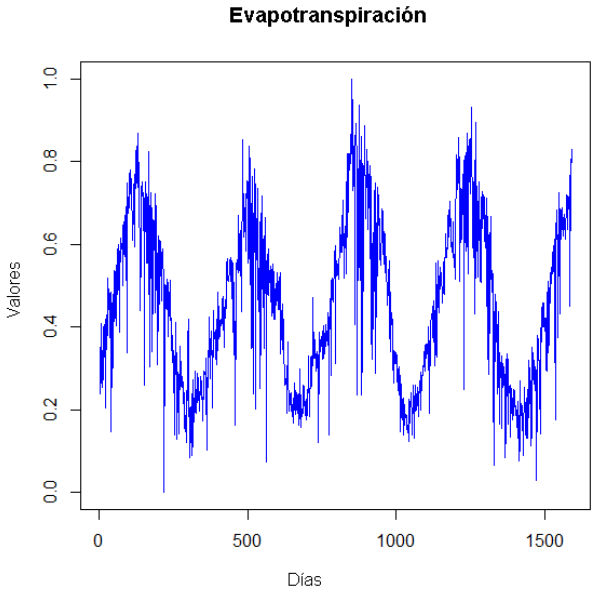

Figura 5.2: Comportamiento de los registros de ETo en el periodo de estudio

5.2.2. Establecimiento del valor inicial de los parámetros libres y del tamaño de ventana móvil

Para la estimación de los parámetros C y ϵ, se utilizó la propuesta de Cherkassky y Ma (2004).

De los datos de entrenamiento se obtuvieron los valores de: $\bar{y} = 0.4528454$ y $\sigma_y = 0.1938998$. Aplicando las ecuaciones (4.16) y (4.19) se obtuvieron los valores inciales $C = 1.034$ y $\epsilon = 0.045$.

Debido a que se está utilizando un kernel de base radial, se tiene que estimar el parámetro σ. La estimación de este parámetro se obtuvo fijando los valores para ϵ y C y mediante un proceso iterativo se asignaron diferentes valores para σ de modo que $0 \leq \sigma \leq 1$, con un tamaño de paso de 0.01. El tamaño de ventana móvil se estimó, de acuerdo a la literarura, con un valor inicial de 5. Como resultado del proceso anterior, se obtuvo el valor $\sigma = 0.125$.

Con el procedimiento descrito anteriormente, se obtuvo una primera estimación de los parámetros. El siguiente paso fue realizar la estimación del tamaño de la ventana móvil con esos valores fijos. De acuerdo a la literatura, la ventana no puede tener un tamaño mayor de 20 y no tiene sentido utilizar una ventana menor a 2 observaciones.

Para verificar el comportamiento del MAPE con diferentes tamaños de ventana, se

realizaron estimaciones de 3 a 10 con tamaño de paso igual a la unidad analizando su comportamiento (ver figura 5.3)

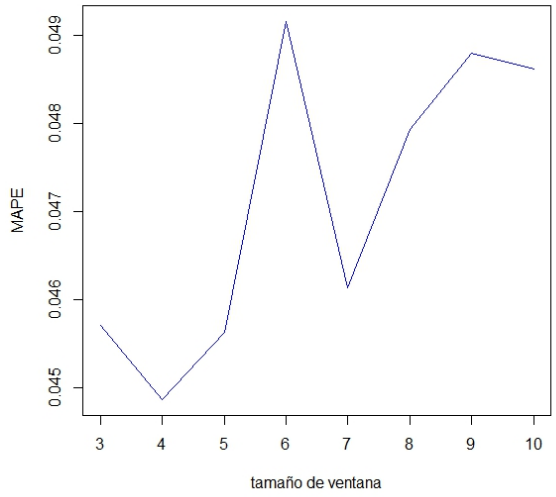

Figura 5.3: Estimación del tamaño de ventana con parámetros fijos

Se fijó el tamaño de la ventana móvil en 4, esto significa que en el proceso de regresión se consideran cuatro valores previos para estimar el quinto valor. Una vez que se ha fijado el tamaño de la ventana móvil, se hace el proceso de "refinamiento", este proceso consiste en asignar diferentes valores para C, ϵ y σ de tal modo que se encuentre el menor error de predicción posible.

5.2.3. Estimación del modelo y pruebas

En la figura (5.4), se presenta la estimación de los 27 valores utilizando una ventana móvil de tamaño 4, $C = 1.034$, $\epsilon = 0.045$ y $\sigma = 0.125$.

En la primera estimación, se tomaron en cuenta solamente los valores iniciales para los parámetros libres y el tamaño de ventana estimada mediante un proceso iterativo. El refinamiento consiste en disminuir el error MAPE. El error obtenido para estos valores de los parámetros es de MAPE=7.0738. Para disminuir este error, el procedimiento que se llevó a cabo es iterar sobre valores muy proximos a los fijados y analizar el comportamiento del error.

En el proceso iterativo para disminuir el error MAPE, se realizaron iteraciones con un tamaño de paso de 0.01, sin embargo el error disminuyó solamente en un orden de

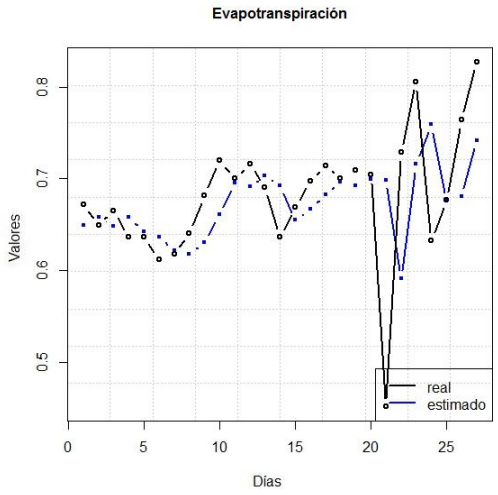

Figura 5.4: Ajuste con una ventana de tamaño 4

10^{-2}. Es importante señalar que el ajuste se realizó por medio de un procedimiento de remuestreo, por lo tanto el resultado sigue siendo muy aproximado. Los valores para los parámetros libres son los siguientes: $C = 1.05$, $\epsilon = 0.07$ y $\sigma = 0.125$. Lo cual representa un error MAPE=7.0309 (ver figura 5.5).

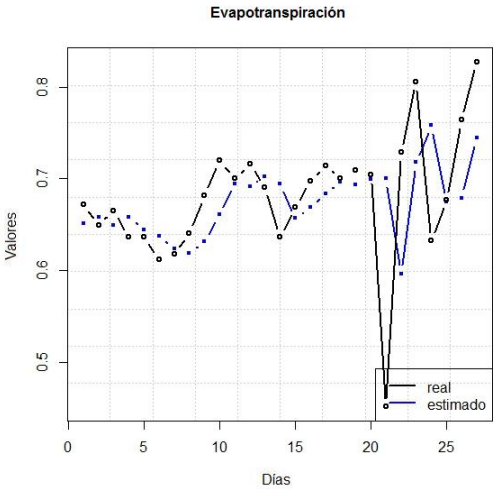

Figura 5.5: Refinamiento con una ventana de tamaño 4

5.3. Redes neuronales

Nuestro segundo objetivo fue utilizar redes neuronales para pronosticar la serie de datos de ETo utilizada en 5.1. De acuerdo a la revisión bibliográfica, las ANN que han tenido más aceptación en la predicción de series de tiempo son los *multilayer perceptron* o perceptrones multicapa. Al ser una herramienta computacional muy aplicable, es posible implementarlas en muchos lenguajes de programación, incluyendo el programa R.

Para estimar los parámetros iniciales, se utilizaron los probados por Radhika y Shashi (2009), para resolver su problema. Ellos utilizaron una red neuronal de tres capas, con una capa de entrada, una capa oculta y una capa de salida, utilizaron un función sigmoide como función de activación. El número de neuronas utilizadas en las capas ocultas es del orden de $(2i+1)$ donde i es el número de entradas. Y el entrenamiento lo realizaron hasta un número específico de iteraciones.

Para el ajuste de los datos de ETo, se inició el procedimiento con una red neuronal de 3 capas, con una capa de entrada, una oculta y una capa de salida, el número de neuronas empleadas en las capas ocultas es de 3 y el entrenamiento se realizó con un total de 100 iteraciones. Bajo este esquema se obtuvo un error MAPE=7.887732, se aumentó una capa oculta y se obtuvo un error MAPE=8.396579, por tanto se utilizó el primer modelo para el comparativo.

El comparativo con los datos reales, está dado en la figura 5.6. El comportamiento de la red neuronal con tamaño de ventana 4 y con dos capas ocultas, 3 neuronas en cada capa oculta y entrenamiento con 100 iteraciones, está dado en la figura 5.7.

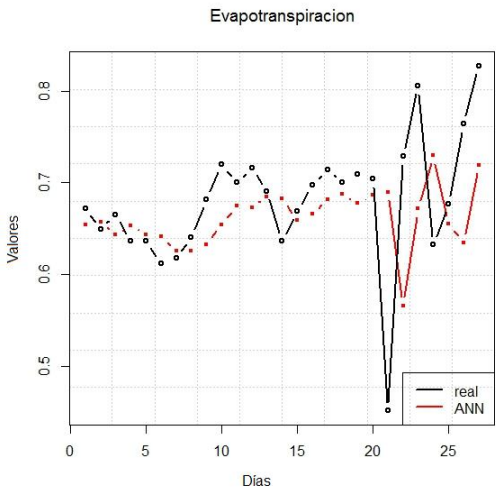

Figura 5.6: Estimación utilizando una red neuronal

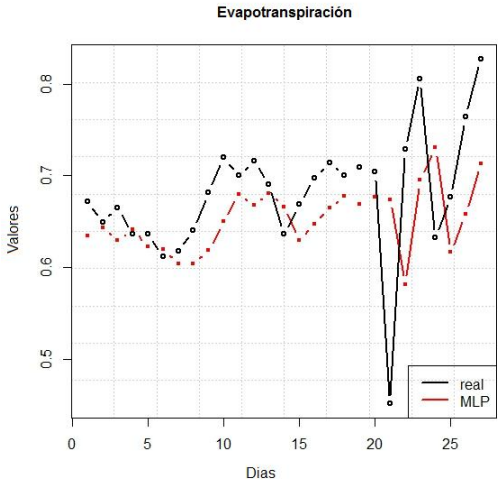

Figura 5.7: Estimación utilizando una red neuronal con dos capas ocultas

5.4. Modelo ARIMA

Posteriormente, se utilizó un modelo ARIMA para pronosticar la serie de datos de ETo utilizados en 5.1 y 5.2. Para el ajuste de esta serie de tiempo, se utilizó el programa SAS, en este caso se estimó la serie considerando los valores sin escalar y los datos escalados, observando que la precisión no tuvo una diferencia significativa. Para realizar el comparativo en iguales condiciones, se utilizaron los datos escalados y se empleó también el conjunto de datos definido como de entrenamiento para la estimación del modelo.

Se realizó una diferenciación a la serie y se obtuvo por medio de la prueba de *Dickey-Fuller* que se comportaba como una serie estacionaria, se empleó la componente ciclica de 360, resultando un proceso autorregresivo de orden $p = 8$. El error estándar estimado promedio es de 0.122.

Para el análisis de la serie de tiempo en SAS se utilizó una rutina sencilla que incluye el procedimiento *arima* y las funciones *identify*, *estimate* y *forecast*.

El error obtenido mediante este modelo es MAPE=19.2934788

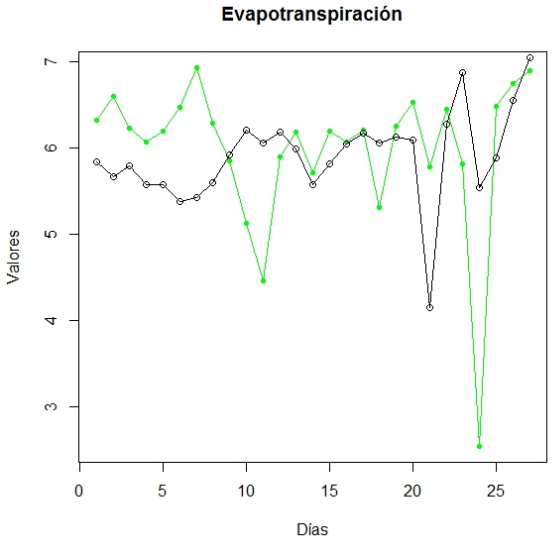

Figura 5.8: Estimación utilizando los modelos ARIMA

5.5. Comparativo de los modelos

La capacidad predictiva de las SVM, se comparó con la red neuronal y un modelo ARIMA, el comparativo es en términos del MAPE, de acuerdo a los ajustes realizados, los valores mínimos del MAPE son los siguientes:

Tabla 5.1: Comparativo del MAPE en los 3 modelos

Modelos	MAPE
ARIMA	19.2934788
ANN	7.887732
SVM	7.0309

De lo anterior, se puede concluir que para este tipo de datos, las máquinas basadas en la teoría del aprendizaje estadístico modelan más apropiadamente los datos. Sin embargo es importante mencionar que la capacidad predictiva conforme incrementa el número de datos a predecir tiende a ser inestable.

Para la predicción de un valor futuro, las SVM consideran 4 datos como ventana móvil, de esta forma para la predicción del primer día del mes de junio, el modelo requiere considerar los valores correspondientes del 28, 29, 30 y 31 de mayo. Para la siguiente predicción se considerará al último elemento estimado, es por esto que si una observación tiene mucha variabilidad el modelo creado con las SVM provoca una diferencia mayor entre los datos estimados y reales.

De acuerdo a los programas establecidos para satisfacer los requerimientos hídricos de los cultivos, la predicción de la ETo puede realizarse con una semana de antelación sin mayor dificultad y con alta precisión, ya que se considera la suma de ETo diaria para una predicción a 7 días.

En sí, la comparación se realiza con varias pruebas basadas o derivadas del error cuadrado medio, por lo que es de esperarse que la proporción de error obtenida con el MAPE se conserve, y pueda utilizarse otro tipo de error para algún problema particular.

En la figura (5.9) y en la tabla (5.2) se muestran los resultados para cada una de las observaciones y los métodos empleados.

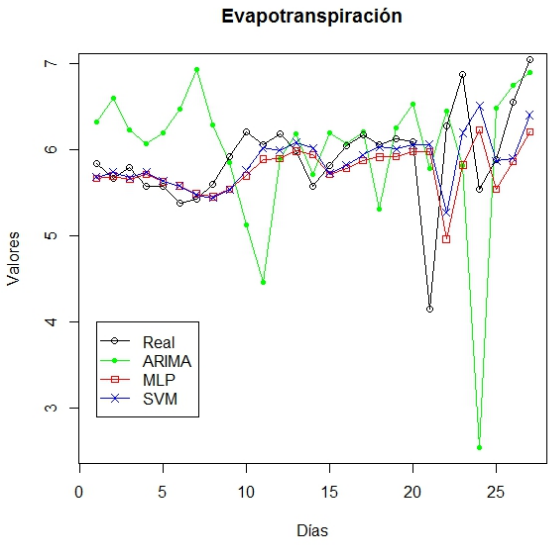

Figura 5.9: Datos reales y el ajuste de los 3 métodos para evapotranspiración

Tabla 5.2: Ajuste de los 3 modelos para datos de ETo

REAL	SVM	ANN	ARIMA
5.84	5.68273971	5.66715351	6.315945
5.67	5.74021026	5.69290783	6.591298
5.79	5.67206001	5.65405621	6.228507
5.57	5.73937039	5.71267802	6.06667
5.57	5.63388181	5.62570943	6.19706
5.38	5.57829195	5.57808793	6.467044
5.43	5.4759151	5.48691157	6.934147
5.6	5.43782818	5.45371275	6.280663
5.92	5.53484371	5.53821927	5.848842
6.21	5.75837358	5.69575724	5.120959
6.06	6.02040993	5.88777799	4.465174
6.18	5.99196267	5.8919474	5.892561
5.99	6.07945206	5.98249252	6.184788
5.57	6.01674597	5.94257017	5.706947
5.82	5.72724335	5.70899259	6.188623
6.04	5.81812672	5.78252948	6.071272
6.17	5.93600005	5.87024053	6.206264
6.06	6.02913455	5.91274	5.308107
6.13	6.00585304	5.92146463	6.246915
6.09	6.05729343	5.97387757	6.522268
4.15	6.05877604	5.97591779	5.777511
6.28	5.26523707	4.96182568	6.440966
6.87	6.19482573	5.81968219	5.815861
5.54	6.5044713	6.2253692	2.540004
5.88	5.86764654	5.5415703	6.484685
6.55	5.89959976	5.86288807	6.747766
7.04	6.40033264	6.20270665	6.891195

5.6. Ajuste de las series de tiempo de variables climáticas

Siguiendo el mismo procedimiento que se empleó para datos de ETo, se implementó el algoritmo de refinamiento para ajustar las series de datos de las variables climáticas que intervinieron en el cálculo de la ETo. Se realizó el ajuste utilizando los tres métodos que se han comparado en este trabajo de investigación. En lo general, con las SVM se obtuvo evidentemente un mejor ajuste que el modelo ARIMA y un ajuste ligeramente superior a las redes neuronales del tipo Multilayer Perceptrón MLP.

Respecto a los datos de temperatura media y humedad relativa, considerando el MAPE, se obtuvo un mejor ajuste por medio de la implementación de las SVM en un orden de 0.1 y 0.03 respectivamente. Las mayor diferencia se obtuvo con los datos de radiación con un orden de 2.1, y respecto a la velocidad del viento la diferencia obtenida fue del orden de 0.5. De acuerdo al algoritmo de refinamiento implementado, los mejores ajustes de las SVM se obtienen cuando los datos no presentan demasiada

variabilidad.

Como resultado del proceso de refinamiento, para cada una de las series de variables climáticas se obtuvieron los parámetros siguientes.

Tabla 5.3: Parámetros de las SVM obtenido con el algoritmo de refinamiento

VARIABLE CLIMATICA	variable C	variable ϵ	variable σ
Temperatura media	1.05	0.037	0.01
Radiación	3	0.0524	0.125
Humedad relativa	3	0.067	0.0312
Velocidad del viento	2.5	0.07534	0.01

Se realizó el comparativo en términos del MAPE, obteniendo los siguientes valores:

Tabla 5.4: Comparativo del error MAPE obtenido con los métodos implementados

VARIABLE CLIMATICA	SVM	ANN	ARIMA
Temperatura media	4.1896	4.29039	4.730355
Radiación	6.6437	8.5509	11.5790
Humedad relativa	8.43	8.46	11.3886
Velocidad del viento	15.60594	16.1954	19.6687

La descripción gráfica de los ajustes es la siguiente:

- En la figura (5.10) se muestran los resultados del ajuste de la temperatura media.
- En la figura (5.11) se muestran los resultados obtenidos de ajustar la radiación.
- En la figura (5.12) se muestran los valores predichos para la humedad relativa.
- En la figura (5.13) se muestran los ajustes de la velocidad del viento.

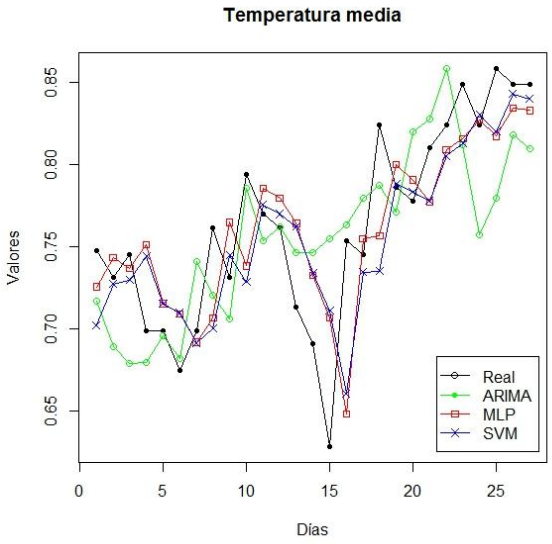

Figura 5.10: Datos reales y el ajuste de los 3 métodos para la temperatura.

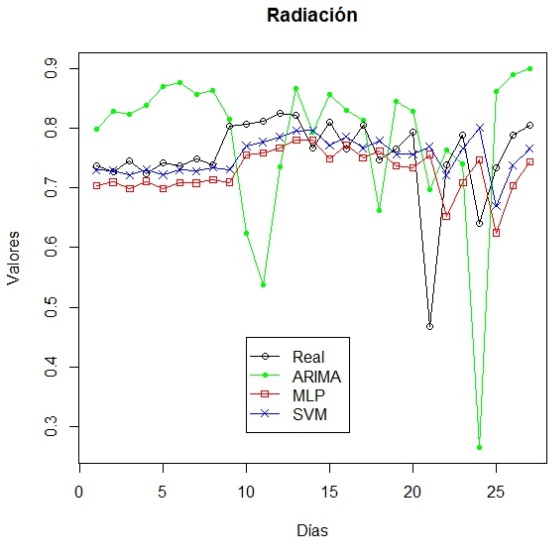

Figura 5.11: Datos reales y el ajuste de los 3 métodos para la radiación.

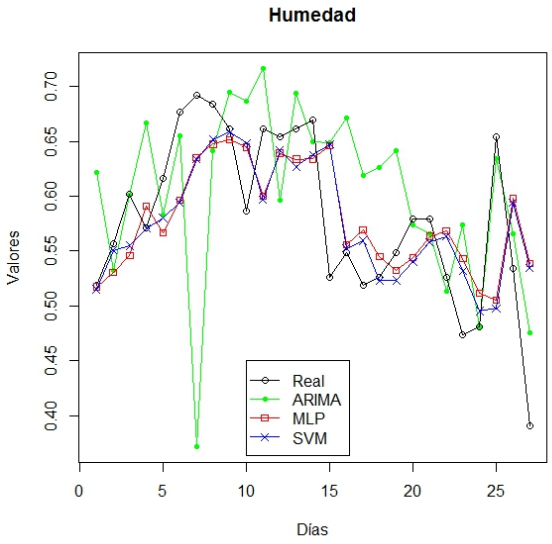

Figura 5.12: Datos reales y el ajuste de los 3 métodos para la humedad relativa.

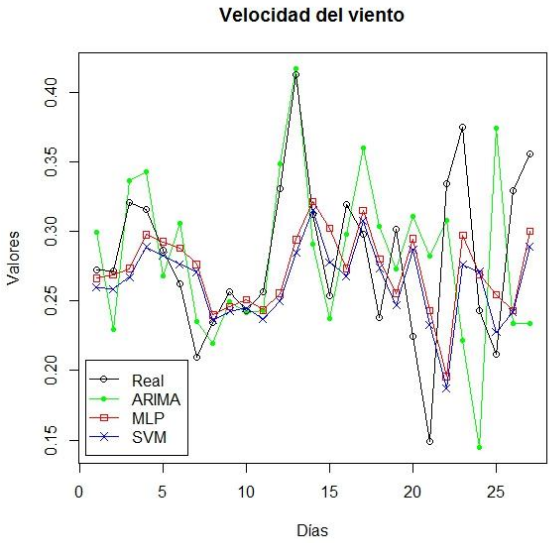

Figura 5.13: Datos reales y el ajuste de los 3 métodos para la velocidad del viento.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Se ha presentado la implementación de las SVM, SNN y un modelo ARIMA en la predicción a 27 días de la serie de tiempo de la variable atmosférica evapotranspiración de transferencia (ETo). La base de datos presenta una variabilidad importante incluso para observaciones provenientes de días consecutivos, además de un número considerable de observaciones atípicas atribuibles a diferentes factores. Esta variabilidad es compleja de modelar para los tres métodos implementados, como consecuencia los modelos obtenidos son un tanto conservadores respecto a su capacidad de predicción.

En las SVM se implementó la estimación inicial de los parámetros libres a partir de la información contenida en fuentes diversas, obteniendo un punto inicial muy próximo al óptimo, evitando en cierta forma la búsqueda mediante procesos iterativos y métodos completamente heurísticos.

Los valores pronosticados con las SVM se han comparado con su contraparte las ANN, en su versión de *Multilayer Preceptron*, de esta comparación se destaca que las SVM se pueden utilizar para conseguir una mejor predicción, para este tipo de datos. Respecto a la comparación con los modelos ARIMA, la capacidad predictiva de las SVM resultó ser todavía mayor.

Las SVM proporcionan un modelo de predicción cuya precisión depende también del número de datos que se desean predecir a futuro, para el caso particular de las observaciones de la ETo, proporciona elementos para hacer una buena estimación de los requerimientos hídricos de los cultivos, impactando directamente en el uso adecuado del agua.

6.2. Trabajo futuro

Una vez que se ha verificado la capacidad predictiva de las SVM, se han detectado las siguientes áreas de oportunidad.

- La predicción a largo plazo disminuye la capacidad predictiva de estas herramientas, en la literatura se han reportado pocas investigaciones sobre la implementación de parámetros adaptativos, lo cual abre un área de oportunidad para la investigación.
- El tiempo computacional que se necesita para el entrenamiento de estas máquinas, está relacionado con el número de variables que se desean modelar, esto se debe a que se resuelven a partir de un problema de programación cuadrática, en este sentido resulta de interés la implementación de otro tipo de algoritmos que permitan encontrar la región factible de este problema en un tiempo menor.
- En la mayoría de las aplicaciones de las SVM se utiliza una función de base radial como función kernel, de acuerdo a la literatura se ha mostrado que el kernel tiene importancia al tratar con datos caóticos, es ente sentido es conveniente ahondar en investigaciones para proponer algún kernel aplicable a problemas altamente complejos.

Referencias

[1] Allen R., Pereira L., Raes D., Smith M., (1998), *Crop evapotranspiration - Guidelines for computing crop requirements, FAO Food and Agriculture Organization of the United Nations*

[2] Anguita D., Carlino L., Ghio A., Ridella S., (2011) *Maximal Discrepancy for Support Vector Machines* Neurocomputing, Vol. 74, Pp. 1436-1443.

[3] Ben-Hur A., Horn D., Siegelman H., Vapnik V., (2001) *Support Vector Clustering* Journal of Machine Learning Research 2.

[4] Benett K. and Mangasarian O., (1992) *Robust linear programming discrimination of two linearly inseparable sets*, Optimization methods and software, Vol. 1, Pp. 23-34.

[5] Burges Christopher, (1998) *A tutorial on Support Vector Machine for pattern recognition*, Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Boston.

[6] Box G. and Jenkins G., (1972) *Time Series analysis forecasting and control*, holden-day, Oakland, California.

[7] Cao L. and Tay F., (2003) *Support Vector Machine with adaptative parameters in financial time series forecasting*, IEE transactions on neural networks, vol. 14 no. 6.

[8] Cherkassky V. and Ma Y., (2004) *Practical selection of SVM parameters and noise estimation for SVM regression*, Neural Networks, Vol. 17 Issue 1, Pp. 113-126.

[9] Chih-Wei H., Chih-Chung C., and Chih-Jen L., (2010), *A Practical Guide to Support Vector Classification*, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.

[10] Fletcher R., (1989) *Practical Methods of Optimization.* John Wiley and Sons, New York.

[11] González J., Cervantes R., Ojeda W., López I., (2008) *Predicción de la evapotranspiración de referencia mediante redes neuronales artificiales* Ingeniería Hidráulica en México, vol. XXIII, Pp. 127-138.

[12] Guerrero V., (2003), *Análisis estadístico de series de tiempo económicas*, Thomson, segunda edición.

[13] Guevara J., (2006), *La fórmula de Penman-Monteith FAO 1998, para determinar la evapotranspiración de referencia, ETo*, Terra Nueva Etapa, vol. XII, número 031, Universidad central de Venezuela, Caracas Venezuela.

[14] Hastie T., Tibshiranni R., Friedman J., (2008) *The Elements os Statistical Learning, Data Mining, Inference and Prediction*, Springer-Verlag, NY, USA. 2nd edition.

[15] Karatzoglu A., Meyer D., Hornik K., (2006)*Support Vector Machines in R*, Journal Statistical Software, Vol. 15, Issue 9.

[16] Kim k., (2003) *Financial time series forecasting using support vector machines*, Neurocomputing Vol. 55 Pp. 307-319.

[17] Lu W., Wang W., (2005) *Potential assessment of the "Support Vector Machine"method in forecasting ambient air pollutant trends*, Chemosphere 59, pp. 693-701.

[18] Luts J., Ojeda F., Van Plas R., De Moor B., Van Huffel S., Suykens J., (2010), *A Tutorial on support vector machine based methods for classification problems in chemometrics*, Analytica Chimica Acta, Vol. 665, Issue 2, Pp. 129-145.

[19] Meyer D., (2003) *The Support Vector Machine under test* Neurocomputing, Vol. 55, issues 1-2, Pp. 169-186.

[20] Müller K., Smola A., Rätsch G., Schölkopf B., Kohlmorgen J., Vapnik V., (1998) *Predicting Time Series with Support Vector Machines*, ICANN'97.

[21] Osuna,E., Freund R., Girosi, F., (1997) *Improved Training Algorithm for Support Vector Machines* Proc IEEE NNSP '97.

[22] Platt J., (1998) *A fast algorithm for training Support Vector Machines* Microsoft research.

[23] Prybutok V., Junsub Y., Mitchell D., (2000), *Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations*, European Journal of Operational Research, Vol. 122, Issue 1., Pp. 31-40.

[24] Radhika Y. and Shashi M., (2009) *Atmospheric Temperature Prediction using Support Vector Machines*, International journal of Computer Theory and Engineering, vol. 1, No. 1. Robenson S., Steyin D., (1989) *Evaluation and Comparison of Statistical Forecast Models for Daily Maximun Ozone Concetrations*, Atmospheric Environment Vol. 24B, No. 2.

[25] Rüping S., Morik K., (2003) *Support Vector Machine and learning about time* Proceedings ICASSP '03, 864-7, vol. 4.

[26] B. Scholkopf, C.J. Burges and J. Smola, editors. *Advances in Kernel Methods-Support Vector Learning* MIT Press, Cambridge, MA, 1999a.

[27] Smola, A., Scholkopf, B. (1998) *A tutorial on support vector regression*, ISIS Technical Report, University of Southampton.

[28] Taylor J., Cristianini N., (2000) *Support Vector Machines and other kernel-based methods*, Cambridge University Press.

[29] Tektas M., (2010) *Wheather Forecasting Using ANFIS and ARIMA MODELS. A Case Study for Istanbul*, Environmental Research, Engineering and Management, No. 1(51), Pp. 5-10.

[30] Vandervei J., (1997) *LOQO user's manual-version 3.10* Technical Report SOR-97-08, Princeton University, Statistics and Operations Research.

[31] Vapnik, V.N. (1995) *The nature of Statistical Learning Theory.* New York: Springer-Verlag.

[32] Vapnik, V.N. (1999) *An Overview of Statistical Learning Theory*, IEE Transactions on Neural Networks, Vol. 10, No. 5.

[33] Velasquez J., Olaya Y., Franco C., (2010) *Predicción de Series Temporales usando Máquinas de Vector Soporte*, Ingeniare. Revista chilena de ingeniería, Vol. 18, Pp. 64-75.