Capitulo 2 - Estado del arte

Aprendizaje supervisado

El aprendizaje inductivo es aquel que a partir de la experiencia y observación permite al hombre comprender los fenómenos que lo rodean. Para realizar este tipo de aprendizaje es necesario recolectar las observaciones para luego ser analizadas y así deducir una hipótesis la cual explique un fenómeno en particular. Los medios para recolectar la información suelen ser la memoria humana, hojas de papel, archivos planos, bases de datos.(Baranauskas y Monard, 2000).

Cuando la cantidad de datos recolectados es grande, el análisis de los mismos para comprender un fenómeno determinado es una tarea dificultosa. Hoy en día los avances en las tecnologías, brindando una alta capacidad de almacenamiento y procesamiento, han hecho de la computadora un elemento muy importante en el proceso de descubrimiento. Esto implica no usar la computadora para que la misma nos indique gráficos o estadísticas, sino utilizarla para que ella misma aprenda de los datos.

El aprendizaje de maquina es una disciplina de la inteligencia artificial dedicada al desarrollo de algoritmos que permiten a la computadora aprender a partir de la experiencia. Ejemplo de tareas que abarca el aprendizaje de maquina pueden ser: descubrir que tratamientos se adaptaran mejor a futuros pacientes a partir del archivo de historias clínicas de un hospital; como hacer que un motor de búsqueda se adapte automáticamente a las preferencias de sus usuarios o como hacer que un robot móvil aprenda a navegar a partir de la experiencia (Mitchell, 2006).

El aprendizaje de maquina supervisado está dedicado al desarrollo de algoritmos que aprenden a partir de ejemplos. Un ejemplo está representado por una instancia y una clase a la que esa instancia pertenece. Una instancia representa un conjunto de observaciones sobre un hecho determinado. Una clase representa la clasificación que se le da al hecho. A partir de un conjunto de ejemplos y un algoritmo de aprendizaje supervisado podemos entrenar un clasificador para que el mismo pueda predecir cuál será la clase de una instancia que desconocemos.

En la figura 2-1 se muestra un conjunto de ejemplos los cuales son utilizados como entrada a un algoritmo de aprendizaje de máquina para generar un clasificador.

Las instancias están compuestas por un conjunto de atributos correspondientes a las observaciones del ambiente en una cancha de golf. El atributo de clase representa si el día es apropiado o no para la práctica de golf en la cancha en cuestión. El usuario incorpora conocimiento acerca del problema seleccionando los datos a usar como entrenamiento así como también en la selección y parametrización de los algoritmos de aprendizaje de máquina.



Fig. 2-1. Proceso de clasificación. (adaptado de A.Baranauskas y C.Monard, 2000)

La selección y preparación de los datos a utilizar tiene mucha importancia dado que el rendimiento del clasificador se verá sumamente condicionado por los mismos. Los datos suelen tener impurezas, como ruidos o valores faltantes, haciendo necesario definir una estrategia para tratar los mismos. Además, los problemas de aprendizaje de maquina suelen tener instancias con un gran número de atributos y no todos suelen ser relevantes para el problema en cuestión. La cantidad de estos afectan negativamente el desempeño, en cuanto a tiempo y precisión, de los algoritmos de clasificación. Para lidiar con este problema suelen ser utilizadas las técnicas de Selección de Atributos y Extracción de Atributos.

El resultado del proceso de aprendizaje es un clasificador, el cual representa un concepto aprendido a partir de los ejemplos. Cuando el formato de este concepto es comprensible por el humano se dice que el sistema de aprendizaje es orientado al conocimiento, de lo contrario el sistema se denomina de caja negra (Kubat y Michalski, 1998 citado en Baranauskas y Monard, 2000). Los sistemas orientados al conocimiento tienen la ventaja de poder ser utilizados no solo para la predicción de la clase de las instancias desconocidas sino que también permiten ser utilizados para comprender el fenómeno subyacente a los ejemplos. Estructuras con forma de árbol o de reglas de decisión son algunas de las usadas para describir los conceptos adquiridos.

El clasificador obtenido es resultado de un proceso inductivo y como tal su efectividad no está garantizada.(Baranauskas y C.Monard, 2000). Además de evaluar el clasificador con el sentido crítico del usuario es necesario establecer métricas claras del desempeño del mismo. Para esto se prueba el clasificador con instancias cuya clase se conoce previamente y se comparan la clase verdadera y la predicha.

En clasificación una métrica común consiste en contar la cantidad de instancias bien clasificadas y las mal clasificadas para luego obtener la taza de error. Cuando el atributo de clase es numérico otras medidas de error son tomadas. Medidas utilizadas frecuentemente son el error absoluto medio, el error cuadrático medio y el coeficiente de correlación. (Witten y Frank, 2005)

Es importante que la evaluación sea realizada con un conjunto de datos independiente del usado para entrenamiento, de lo contrario la evaluación no sería representativa del desempeño del clasificador ante datos nuevos. El ideal es contar con tres conjuntos de datos: uno para entrenamiento, uno para validación y otro para pruebas. El conjunto de entrenamiento es utilizado para obtener el clasificador. El conjunto de validación para ajustar los parámetros del clasificador. El conjunto de prueba para obtener las medidas de performance finales del clasificador. En situaciones donde el número de ejemplos no resulta abundante para separar en estos tres conjuntos las técnicas de re muestreo resultan útiles. Ejemplo de ellas son: dejar-uno-afuera, validación cruzada de 10 particiones estratificada. Siendo esta ultima la técnica de evaluación estándar cuando la cantidad de datos es limitada. (Witten y Frank, 2005)

Métodos de regresión:

a. Regresión Lineal Simple:

El análisis de regresión de línea recta implica una variable de respuesta Y y una sola variable predictora X. Esta es la forma de regresión más simple que podemos encontrar, donde se modela Y como una función lineal de X. (Han y Kamber, 2000) Esto es,

*y* = *b* + a*x*; (Ec. 2-1)

donde la varianza de y es supuesta constante, b y a son los coeficientes de regresión que especifican la intercepción con el eje Y, y la pendiente de la recta respectivamente.

Estos coeficientes pueden ser resueltos mediante el método de mínimos cuadrados, el cual estima la recta de mejor ajuste como aquella que minimiza el error entre el dato verdadero y la estimación dada por la línea recta. Sea |D| el conjunto de entrenamiento que contiene puntos de datos de la forma (, ), (, ),…, (, ), podemos definir el error en función de los parámetros como:

(Ec. 2-2)

Donde , es el valor verdadero e el valor estimado por la función de regresión.

Queremos encontrar los valores de a y b que minimicen la función de error. Diferenciando esta función con respecto a cada parámetro hayamos las siguientes ecuaciones las cuales nos proporcionan los parámetros de la recta de mejor ajuste:

(Ec 2-3)

(Ec 2-4)

Donde es el promedio de x1, x2, ..., x|D|, y el correspondiente a y1, y2, ..., |D|.

*Ejemplo 2-1:*

Regresión lineal usando el método de mínimos cuadrados. La tabla 2-1 muestra los pares de datos: años de experiencia laboral de un graduado universitario(X) y el salario correspondiente(Y).

|  |  |
| --- | --- |
| años de experiencia (X) | salario(Y)(miles) |
| 3  8  9  13  3  6  11  21  1  16 | 30  57  64  72  36  43  59ª  90  20  83 |

Tabla 2-1: Relación años de experiencia, salario percibido. . (Han y Kamber, 2000)

Fig. 2-2: Gráfico de los datos de la tabla 2-1 junto con la recta de regresión

calculada utilizando el método de regresión lineal simple. . (Han y Kamber, 2000)

El grafico de la figura 2-2 sugiere una relación lineal entre las dos variables X e Y. De esta manera podemos modelar la relación entre el salario y la cantidad de años de experiencia mediante la ecuación de regresión lineal 2-1. Utilizando los datos de la tabla 1 junto con las ecuaciones (2-3) y (2-4) obtenemos el valor de los coeficientes:

a = 23,6

b = 3,5:

Reemplazando en 2-1 obtenemos la función de la recta de regresión:

Y = 23,6 + 3,5X. (Ec. 2-5)

Utilizando esta función podemos predecir el valor de salario para un nivel de experiencia de la cual no tenemos información. Por ejemplo podemos decir que se estima que un egresado universitario con X = 10 años de experiencia, posee un salario de: Y = 23,6 + 3,5 \* 10.= $58600.

La regresión lineal múltiple es una extensión de la regresión lineal simple de manera de poder incorporar más de una variable predictora. La misma permite modelar la variable de respuesta como una función lineal de N variables predictoras, formando una tupla, ( X = (x1, x2, : : : , xn).) Nuestro conjunto de entrenamiento D contiene datos de la forma (X1, y1), (X2, y2), : : : , (XjDj, yjDj), donde las Xi son tuplas de entrenamiento N dimensionales, con etiquetas de clases asociadas Yi. Un ejemplo de un modelo de regresión múltiple basado en dos variables predictoras es:

Y = w0 + w1x1+w2x2, (Ec. 2-6)

El método de cuadrados mínimos es extendido para encontrar los parámetros w0, w1, y w2 que minimicen la función de error asociada a este modelo. Las ecuaciones de regresión múltiple son tediosas para resolver a mano y este tipo de problemas son comúnmente resueltos usando paquetes estadísticos o de minería de datos.

La regresión lineal es un método simple pero poderoso para ser utilizado en la predicción numérica, el mismo ha sido usado ampliamente en aplicaciones estadísticas durante décadas. La desventaja que presenta este método es la linealidad. Cuando los datos exhiben una dependencia no lineal, la mejor recta de ajuste será encontrada. Esta recta no se ajustará demasiado bien a este tipo de datos. A pesar de esto los modelos lineales son interesantes ya que sirven como base para el desarrollo de métodos de aprendizaje más complejos(Witten y Frank, 2005).b. Redes neurales para regresión

Una Red Neuronal Artificial (Red Neuronal o ANN), es un modelo computacional inspirado en las redes neuronales biológicas. La red está conformada de elementos de procesamiento (neuronas), conexiones entre los elementos de procesamiento y coeficientes (pesos) asociados a cada conexión. Todos estos elementos forman la estructura neuronal.

Las redes neuronales, en su mayoría, tienen la particularidad de ser sistemas adaptativos, es decir que adaptan su comportamiento de acuerdo al ambiente en donde se encuentran. Así la red es capaz de modificar su estructura, en la fase de aprendizaje, de acuerdo a la información que se le presenta a la misma.

En términos prácticos las redes neuronales son técnicas de modelamiento no lineales capaz de modelar funciones complejas. Estas pueden ser aplicadas a problemas de regresion, clasificación o control en un amplio espectro de campos como finanzas, neurociencia, medicina, ingeniería y física.

El bloque fundamental para la construcción de la red neuronal artificial es el modelo matemático de una neurona como se muestra en la figura 2-3. Los tres componentes básicos de una neurona artificial son:

* Las conexiones que proveen pesos Wj a los valores de entrada Xn.
* Una función de transferencia para computar el valor de entrada a la función de activación.
* Una función de activación g que mapea v a g(v), el valor de salida de la neurona.



Fig. 2-3 Modelo de una neurona artificial.

Mientras existen numerosas arquitecturas de redes neuronales, las aplicaciones con mejores resultados en el área de minería de datos han sido las redes Multilayer Feedfoward (MIT, 2009). Estas son redes en las cuales existe una capa de nodos que simplemente aceptan los valores de entrada y capas sucesivas que son neuronas como las de la figura 2-3. Las salidas de las neuronas en una capa son entrada a las neuronas en la capa sucesiva. La última capa es denominada capa de salida. Las capas entre la entrada y la salida son denominadas capas ocultas, ya que no interactúan con el medio externo. La figura 2-4 muestra un diagrama para esta arquitectura.



Fig. 2-4 Arquitectura de red neuronal Multilayer Feedforward.

Una de las características más importantes de las redes neuronales es su habilidad para aprender. Si consideramos una red neuronal como una función de mapeo F: X→Y, siendo X un vector de entrada a la red e Y un vector de salida de la misma. Dada una tarea específica para resolver y una clase de función F, aprender significa usar un conjunto de observaciones para encontrar un f\* F, que resuelva la tarea específica de manera óptima. La solución óptima será aquella que haga la función de costo mínima. Esta función representa el error del clasificador entre el valor de la predicción y el valor verdadero.

A pesar que la función de costo puede ser elegida de manera arbitraria, la elección de la misma suele ser realizada basada en las propiedades de la misma (convexidad) y también en las particularidades del problema que se intenta resolver. Finalmente la elección de la función de costo dependerá del tipo de tarea que intentemos resolver. La función de costo más utilizada en la práctica resulta ser la función de mínimos cuadrados. Esta resulta ser la misma función utilizada en la regresión lineal simple (Ver ecuación 2-2).

El problema de entrenamiento puede dividirse en dos: aprender la estructura de la red y aprender los pesos de las conexiones (Witten y Frank, 2005).

Existen numerosos algoritmos de entrenamiento, que resuelven de manera simple los valores de los pesos, dada una estructura de red fija. Ejemplo de ellos son: Propagación hacia atras, Propagación rápida (ver Patterson, 1996; Haykin, 1994; Fausett, 1994), Gradiente Conjugado descendiente (ver Bishop, 1995). Si bien existen algoritmos para encontrar una estructura de red adecuada este aspecto del problema suele ser resuelto a través de la experimentación.

Los algoritmos de estimación de pesos son, en su mayoría, aplicaciones de la teoría de optimización y de estimación estadística. Estos utilizan alguna variante de la técnica de optimización [Gradiente](http://en.wikipedia.org/wiki/Gradient_descent) descendente. El algoritmo más conocido es denominado “Propagación hacia atrás”

La técnica de “Propagación hacia atrás” se compone de un ciclo de dos fases(MIT, 2009): una fase hacia adelante, donde un dato de entrenamiento se introduce en la red y se calculan las salidas de todos los nodos hasta llegar al nodo final que produce el resultado de predicción. Una fase de retroceso en la cual se van actualizando los pesos de las conexiones desde los nodos de la capa de salida hasta la capa de entrada.

“Propagacion hacia atrás” utiliza Gradiente Descendente, una técnica de optimización iterativa que usa la información de la derivada de primer orden de la función de costo para ajustar los pesos de la red. A partir del valor de las derivadas, las multiplica por una pequeña constante llamada **tasa de aprendizaje** y luego sustrae el resultado al valor actual del peso. Esto es repetido en cada ciclo hasta que el cambio en el valor del peso se torna muy pequeño, de esta manera hemos encontrado la configuración de los pesos que logran un mínimo de la función de costo elegida.

La **tasa de aprendizaje** determina el incremento en dirección al mínimo y por lo tanto que tan rápido la búsqueda converge. Si esta tasa es muy grande y la función tiene múltiples mínimos, la búsqueda puede pasar por alto algún mínimo, o puede oscilar fuertemente. Si la tasa es pequeña el progreso hacia un mínimo puede volverse demasiado lento. Cabe destacar que el método de Gradiente Descendente solo puede encontrar un mínimo local. Si la función de costo tiene varios mínimos puede ser que no se encuentre el mínimo óptimo. Para aliviar este problema suelen realizarse múltiples corridas inicializando los valores de pesos en forma aleatoria.

Como en cualquier otra técnica de aprendizaje de maquina en las redes neuronales podemos sufrir el problema de sobre entrenamiento (del inglés overfitting), es decir que la red puede reflejar una buen desempeño con los datos de entrenamiento, pero no así con datos nunca vistos.

**Detención temprana** es una modificación a la técnica de Gradiente Descendente la cual consiste en tener un conjunto de datos separado para verificar el desempeño de la red en cada iteración del ciclo de propagación hacia atras. Cuando el desempeño medido con este conjunto de datos empieza a decaer, indicando sobre entrenamiento, el algoritmo es terminado.

Una pasada por todos los datos de entrenamiento se denomina una **Epoca**. La mayoría de las redes del tipo feedforward requieren de varias épocas antes de que los errores sean razonablemente pequeños.

El **momento** es una solución para minimizar el número de épocas necesarias para encontrar un mínimo aceptable. La misma consiste en agregar al peso que se está actualizando una proporción del incremento agregado en la iteración previa. Esto genera que el proceso de búsqueda sea más suave haciendo los cambios en dirección menos abruptos y favoreciendo una convergencia más rápida. Valores altos en el parámetro de momento forzaran a que los ajustes sucesivos sean en direcciones similares. Otra idea es variar el parámetro de **tasa de aprendizaje** para que este comience con un valor alto e ir decrementandolo a medida que se avanza de época(Witten y Frank, 2005).

Las redes neuronales pueden verse como una especie de sistema de procesamiento no lineal capaz de resolver un amplio espectro de problemas. Las redes neuronales son útiles cuando existen datos en abundancia pero se carece de una base teórica completa, es decir, no hay un modelo causal o una representación matemática. Los datos disponibles suelen ser no lineales, no estacionarios, o caóticos haciéndolos difíciles de modelar. Las redes neuronales no suponen ningún conocimiento previo acerca del espacio del problema, tampoco necesitan conocimientos previos en cuanto a la distribución estadística de los datos.

Las tareas en la cuales las redes neuronales son aplicadas se encuentran dentro de las siguientes categorías: [Aproximación de funciones](http://en.wikipedia.org/wiki/Function_approximation), o análisis de regresión, incluyendo predicción de series de tiempo y modelamiento. [Clasificación](http://en.wikipedia.org/wiki/Statistical_classification), incluyendo reconocimiento de patrones y reconocimiento de secuencias Procesamiento de datos, incluyendo filtrado, clusterización y compresión.

Las áreas de aplicación de las redes neuronales incluyen: Sistemas de control(control de vehículos), juegos(backgammon, ajedrez), reconocimiento de patrones(sistemas de radares, identificación de caras, reconocimiento de objetos), reconocimiento de secuencias(gestos, habla, escritura), diagnósticos médicos, aplicaciones financieras, descubrimiento de conocimiento en bases de datos, visualización y filtrado de email spam.

c. Árboles modelos y de regresión.

La regresión lineal es un modelo global, donde existe una única ecuación predictiva que se mantiene para todo el espacio de datos. Cuando los datos tienen muchas características los cuales interactúan de una forma complicada, en formas no lineales, encontrar un único modelo global puede ser muy difícil.(Shalizi, 2009) Inclusive una vez encontrado este modelo, el mismo suele ser confuso.

Una alternativa al enfoque no lineal es la de subdividir o particionar el espacio en regiones más pequeñas donde las interacciones son más accesibles. Luego estas particiones se vuelven a sub dividir y así sucesivamente. Finalmente obtenemos porciones del espacio en donde podemos utilizar modelos sencillos para encajar los datos. De esta manera el modelo global tiene dos partes: una consiste en la partición recursiva del espacio, la otra en aplicar un modelo simple a cada celda de la partición. La alternativa mencionada no es más que la aplicación de la estrategia “Divide y vencerás”. Este enfoque conlleva a adoptar un estilo de representación de los datos en forma de árbol. En cada nodo de un árbol se evalúa un atributo en particular. Generalmente se compara el nodo con un atributo constante. Los nodos hojas dan una clasificación que se aplica a todas las instancias que alcanzan la hoja. Para clasificar una instancia desconocida, la misma es encaminada desde la raíz del árbol hacia abajo de acuerdo a los valores de los atributos que se evalúan en cada nodo y cuando una hoja es alcanzada la instancia es clasificada de acuerdo a la clase asignada para esa hoja.

Existen dos clases de arboles utilizados para predicción numérica, los arboles de regresión (Breiman et al., 1984 citado en Witten y Frank, 2005) y los arboles modelo (Quinlan, 1992 citado en Witten y Frank, 2005). La única diferencia entre ambos es que el primero almacena en sus nodos hojas el promedio de los datos que alcanzaron ese nodo, mientras que el segundo almacena un modelo de regresión lineal el cual es usado para predecir el valor de las instancias que alcanzan ese nodo (Witten y Frank, 2005). En lo que sigue describiremos los arboles modelos, ya que los arboles de regresión son un caso especial.

*Construcción del árbol modelo:*

La construcción del árbol modelo es un proceso recursivo. Comenzando del nodo raíz, se selecciona el atributo que mejor separa los datos de entrenamiento. Evaluando este atributo para cada instancia del conjunto de entrenamiento el mismo quedara separado en diferentes subconjuntos. Existirán tantos subconjuntos como posibles alternativas presente el nodo de decisión. Este proceso es repetido para cada subconjunto de los datos de entrenamiento hasta que todas las instancias que alcanzan un nodo tienen la misma clasificación.

Para determinar que atributo es el que mejor separa la porción T de los datos de entrenamiento que alcanzan un nodo en particular se utiliza el criterio de particionamiento. El mismo está basado en utilizar la desviación estándar de los valores de clase de T como una medida del error en ese nodo. El atributo que maximiza la reducción del error esperado es elegido para particionar los datos que llegan al nodo. La reducción del error esperado esta dado por la siguiente fórmula: (Witten y Frank, 2005).

(Ec.2-5)

donde T1, T2, . . . son los conjuntos que resultan de separar el nodo de acuerdo al atributo elegido. El proceso de particionamiento termina cuando el valor de clase de las instancias que alcanzan un nodo varían muy poco, es decir cuando su desviación estándar es solo una pequeña fracción (Ej.: %5) de la desviación estándar del conjunto de instancias original. El particionamiento también termina cuando quedan unas pocas instancias en un nodo, por ejemplo: 4 instancias. La experimentación indica que los resultados obtenidos no son muy sensible al valor de estos parámetros.

Para predecir el valor de una instancia de prueba el árbol es atravesado hasta las hojas usando los valores de los atributos para decidir qué camino tomar en cada nodo. La hoja tendrá un modelo lineal el cual será utilizado para obtener el valor de la predicción. En vez de utilizar este valor directamente, resulta ser beneficioso realizar un proceso de suavizado para compensar las discontinuidades que inevitablemente ocurren entre los modelos adyacentes en las hojas del árbol. Este proceso es llevado a cabo implementando un modelo lineal en cada nodo interno del nodo, además del de las hojas. Luego una vez obtenida la predicción dada por el modelo en la hoja, este valor es filtrado durante el camino hacia el nodo raíz, suavizándolo en cada nodo que es atravesado mediante la combinación con el valor predicho por el modelo de cada nodo. Un modelo apropiado para calcular este suavizado esta dado por:

(Ec. 2-6)

“p’” es la predicción pasada al nodo superior en el árbol, “p” es la predicción pasada al nodo actual proveniente del nodo inferior, “q” es el valor predicho por el modelo en el nodo actual, “n” es el número de instancias de entrenamiento que alcanzaron el nodo inferior, y “k” es una constante de suavizamiento. La experimentación muestra que realizando este proceso de suavización se mejora substancialmente la precisión de las predicciones.

*Poda:*

A pesar que los arboles construidos mediante el enfoque de “divide y vencerás” tienen un buen rendimiento sobre los datos de entrenamiento, los mismos suelen tener problemas de sobre entrenamiento y no pueden generalizar correctamente con conjuntos de pruebas independientes. Una solución a este problema es la realización de podas sobre el árbol.

Existen dos maneras para realizar la poda: **pre-poda** y **pos- poda**. En pre-poda cuando se va construyendo el árbol y se encuentra una estructura que es lo suficientemente compleja, se detiene la construcción en esa rama. En pos-poda primero se construye todo el árbol y luego las descripciones complejas son extraídas (Witten y Frank, 2005).

Pre-poda implica decidir, durante la construcción del árbol, cuando parar de desarrollar los sub-arboles- esta particularidad resulta interesante ya que, a diferencia de pos-poda, no se hace trabajo de mas desarrollando sub arboles que luego podrían ser desechados. A pesar de esto pos-poda ofrece ciertas ventajas. Por ejemplo, existen situaciones en la que dos atributos considerados individualmente no tienen ningún aporte significativo, mientras que los mismos considerados en conjunto resultan ser muy informativos. De esta manera para asegurarnos de poseer la mayor información posible es necesario construir el árbol completo para luego desechar las partes que provocan sobre entrenamiento. La mayoría de los algoritmos de construcción de arboles utilizan pos-poda. Actualmente es una pregunta abierta si las estrategias de pre-poda pueden ser desarrolladas para alcanzar el mismo rendimiento que las estrategias de pos-poda.



Fig. 2-5 Árbol modelo.(Witten y Frank, 2005)

En la figura 2-5, se muestra un ejemplo de un árbol modelo: En cada nodo se evalúa un atributo diferente y en las hojas se encuentra identificado el modelo lineal a utilizar.

d. Maquinas de Soporte Vectorial para regresión:

Las maquinas de soporte vectorial fueron desarrolladas como se conocen hoy en día en los laboratorios AT&T por Vappnik y colaboradores. (Boser et al.,1992, Guyon et al.1993, Cortes y Vapnik,1995, Scholkopf et al.,1995, Vapnik et al., 1997, citados en A. Smola y B. Scholkopf, 1998) Debido a este contexto industrial la investigación fue orientada a aplicaciones del mundo real, específicamente al reconocimiento de caracteres. En un corto periodo de tiempo estos clasificadores se convirtieron en competidores de las mejores técnicas existentes del momento. Así también las maquinas de soporte vectorial utilizadas para clasificación fueron extendidas para soportar problemas de regresión, obteniéndose también muy buenos resultados. (Müller et al., 1997, Drucker et al., 1997, Stitson et al., 1999, Matera y Haikin, 1999 citados en A. Smola y B. Scholkopf, 1998). Actualmente los algoritmos de maquinas de soporte vectorial forman parte de cualquier herramienta estándar para minería de datos. A continuación explicaremos los conceptos detrás de las maquinas de soporte de vectores utilizadas para regresión:

Como con la regresión lineal, la idea básica es encontrar una función que aproxime los puntos de entrenamiento minimizando el error en la predicción. La diferencia crucial es que todas las desviaciones hasta un parámetro ɛ dado son descartadas. (Witten y Frank, 2005).

Un parámetro ɛ especificado por el usuario define un tubo alrededor de la función de regresión en los cuales los errores son ignorados: para soporte de vectores lineal el tubo es un cilindro. Si todos los puntos de entrenamiento caben dentro de un tubo de 2ɛ, el algoritmo obtiene una función en el medio del tubo más horizontal que los encierra. En este caso el error percibido es cero. La figura 2-6 muestra un problema de regresión con un atributo, una clase numérica, y ocho instancias. En este caso el valor de ɛ fue configurado en 1, siendo el ancho del tubo alrededor de la función de regresión igual a 2. La figura 2-6 b muestra la salida del proceso de aprendizaje con el valor de ɛ configurado en 2. Como se puede apreciar un tubo más ancho hace posible aprender una función más horizontal.

El valor de ɛ controla que tan cerca la función encajaralos datos. Un valor demasiado grande producirá un predictor sin sentido – en el caso extremo, cuando 2ɛ excede el rango de valores de la clase de los datos de entrenamiento, la línea de regresión es horizontal y el algoritmo solo predice el valor promediode clase. Por otro lado, para valores pequeños de ɛ puede no haber un tubo que encierre todos los datos. En este caso algunos puntos de entrenamiento tendrán un error diferente de cero, y existirá un compensación (del inglés “trade-off”) entre el error de predicción y la horizontalidad del tubo. En la figura 2-6aɛ fue configurado en 0.5 y no existe ningún tubo de ancho 1 que pueda encerrar todos los datos.

Para el caso lineal, la función de regresión con soporte de vectores puede ser escrita:

(Ec. 2-7)

Los vectores de soporte son aquellos puntos que no caen estrictamente dentro del tubo –o sea, los puntos afuera del tubo y sobre el borde (Ver figura 2-6a). Todos los puntos dentro del tubo se les asigna coeficiente 0 y pueden ser eliminados de los datos de entrenamiento sin cambiar la salida del proceso de aprendizaje.

En la mayoría de los casos los datos no son linealmente separables, son problemas no lineales, las MVS proveen soporte para este tipo de casos. Cuando los datos no son linealmente separables en el espacio de entradas, son problemas no lineales; En este caso se puede realizar una transformación no lineal del espacio de entradas, en un espacio de características. Este nuevo espacio permite que los datos puedan ser separados linealmente de manera que se pueden aplicar los mismos razonamientos que para las MVS lineales. Así podemos utilizar la ecuación de regresión 1, reemplazando el producto escalar por una función de núcleo. Una función núcleo o kernel se puede definir como aquella que permite realizar la separación y el traslado de los datos al espacio de características. Existen diversos kernels predeterminados conocidos entre los cuales se destaca el lineal, el RBF (Función de Base Radial), el polinomial, el sigmoidal.

El algoritmo de SVM para regresión funciona buscando simultáneamente una minimización del error y una maximización de la horizontalidad de la función de regresión. Sin embargo, cuando los datos no son linealmente separables incluso luego de aplicar una transformación, algunos puntos no encajan en el tubo (figura 2-6 c)y no existe un tubo con error igual a 0. Luego deberá realizarse una compensación entre el error en la predicción y la horizontalidad del tubo. Esta compensación es controlada forzando un límite superior C en el valor absoluto de los coeficientes . El límite superior restringe la influencia de los vectores de soporte en la forma de la función de regresión y es un parámetro que el usuario debe especificar en adición a ɛ. Mientras más grande sea C lo más cerca la función encajaralos datos. En el caso degenerado (ɛ=0) el algoritmo simplemente realiza una regresión de *error absoluto mínimo* utilizando la restricción del coeficiente y todas las instancias de entrenamiento se transforman en vectores de soporte. Contrariamente si ɛ es suficientemente grande como para que el tubo acomode todos los datos, el error se vuelve 0, no hay compensación para hacer y el algoritmo obtiene el tubo mas horizontal que encierra a los datos indiferentemente del valor de C.

Comparado a otros métodos, como los arboles de decisión, incluso los algoritmos más veloces para soporte de vectores son lentos cuando son aplicados en un contexto no lineal. Por otro lado, suelen producir clasificadores muy precisos debido a los detalladosy complejos limites de decisión que pueden ser obtenidos.

1. **
2. **
3. **

Fig. 2-6: Regresión con soporte de vectores: (a) ɛ = 1, (b) ɛ = 2, y (c) ɛ = 0.5.

(Witten y Frank, 2005)

3 Problema de predicción de oleaje:

En la actualidad existen una variedad de deportes acuáticos que se ven beneficiados por los pronósticos acerca del oleaje en una zona cercana a la costa. Ejemplos de estos pueden ser el surf, windsurf, kitesurf, kayac, etc. Los deportistas están sumamente atentos a las condiciones que presenta el mar para evaluar la playa en la que van a practicar el deporte. Utilizando un pronóstico de oleaje el deportista puede planificar sus actividades para lograr estar presente en el lugar con mejores condiciones. Eventualmente el deportista puede evitarse recorridas extensas buscando el sitio con las mejores condiciones del día.

Actualmente los pronósticos de oleaje son derivados a partir de la resolución de un modelo matemático global sumamente complejo. Este provee de la predicción de la altura de las olas, la dirección de las mismas y el tiempo entre una ola y otra (período). Si bien este pronóstico tiene buen rendimiento en alta mar, no sucede lo mismo en la cercanía de la costa. Esto se debe a que la interacción de las olas con el lecho marino cercano no es contemplado por estos modelos. De acuerdo a la geografía de la costa puede suceder que en un rango de 15 km un lugar presente condiciones apropiadas para la práctica del deporte mientras que en otro las condiciones sean desfavorables. Se sabe que dentro de una misma costa expuesta a las mismas condiciones de mar las condiciones del lecho marino de una determinada playa puede aumentar o disminuir el tamaño de las olas significativamente. Por esta razón es que la utilización de un pronóstico preciso en las cercanías de la costa es de suma utilidad.

Existen modelos para contemplar el comportamiento de las olas en las cercanías de la costa. Los mismos reciben como entrada los datos de pronóstico de alta mar e información acerca del lecho marino. Utilizando estos datos los mismos obtienen un refinamiento del pronóstico inicial. Si bien estos modelos poseen un buen comportamiento, su limitación está relacionada con la obtención de los datos del lecho marino. Usualmente la obtención de estos datos es costosa y no se encuentra disponible en todos los lugares.

Nuestra propuesta consiste en utilizar modelos de aprendizaje de máquina para aprender de casos pasados como un determinado pronóstico de alta mar se vio reflejado en las cercanías de la costa. De esta manera ante un nuevo pronóstico de alta mar se podrá predecir cuál será el efecto del mismo en las cercanías de la costa. Nos centraremos en la predicción de la altura de la ola en la cercanía de la costa, mejorando la predicción del pronóstico de alta mar. Para realizar esta predicción utilizaremos métodos de regresión los cuales tendrán como datos de entrada: el pronóstico de alta mar y la observación de la altura en la cercanía de la costa.

4. Trabajos relacionados a la predicción de oleaje

A continuación se describen las características de los principales trabajos relacionados a la predicción de la altura de la ola mediante el uso de técnicas de regresión.

(Caldwell y Aucan, 2007) y (Browne et al.,2006) se caracterizan por enfocarse en la predicción de la altura de la ola en la zona cercana a la costa donde se realiza la práctica de surf. Estos a partir de la información del estado del mar en una zona de alta mar(X) infieren el tamaño que presentara este mismo estado en la zona de surf (Y). Es decir el objetivo es inferir todo el proceso de transformación que experimenta un estado del mar desde el punto X, hasta el punto Y. Para lograr esto definen un modelo de regresión en donde las variables predictivas están dadas por los datos en alta mar (X), y la variable de respuesta es la altura de la ola en la playa (Y). (Caldwell y Aucan, 2007) utiliza como variables predictivas los datos de alta mar recolectados por una boya marina, mientras que (Browne et al.,2006) utiliza las previsiones oceánicas brindadas por la resolución del modelo numérico WaveWatch3 (ver Capítulo 3). En ambos trabajos la variable de respuesta está representada por las observaciones visuales recolectadas por un observador experto el cual registra el tamaño de ola promedio en un determinado momento del día.

Los trabajos (Deo y Sridhar, 1998) y (Makarynskyy, 2004) se caracterizan por enfocarse en la predicción de la altura de la ola en el mismo lugar donde se recolectan los datos. Estos utilizando el estado del mar de las últimas horas en un lugar X, infieren cual será la altura de la ola en ese mismo lugar con horas de anticipación. Para lograr esto se plantea un modelo de regresión donde las variables predictivas están dadas por las lecturas de una boya y la variable de respuesta está dada por la altura de la ola que se presentara en ese mismo lugar con ‘n’ horas de anticipación. Estos trabajos difieren del objetivo que presenta nuestro caso de estudio de predecir la altura de la ola en una zona diferente de donde son leídos los datos del mar. A pesar de esta diferencia los trabajos mencionados son validos como referencia por estar relacionados con el mismo fenómeno natural.

(Deo y Sridhar, 1998), (Browne et al.,2006), y (Makarynskyy, 2004) desarrollan modelos de regresión en base a redes neuronales, mientras que (Caldwell y Aucan, 2007) utilizan un modelo de regresión polinomial.

Las redes neuronales desarrolladas, son del tipo multilayer feedforward. En todas ella se usa una función de transferencia del tipo sigmoidal. Los algoritmos de aprendizaje utilizados son diversos. (Deo y Sridhar, 1998) evalúa los algoritmos de Propagación hacia atrás, Gradiente conjugado y Correlación en cascada, obteniendo similares resultados en términos de correlación entre la altura observada y la altura predicha, aunque se observa un comportamiento superior en términos de tiempo de convergencia mediante el algoritmo de correlación en cascada. (Browne et al.,2006) experimenta con el algoritmo de aprendizaje de Levenburg-Marqhardt, el mismo es una forma más robusta del algoritmo de Gauss-Newton. (Makarynskyy, 2004) utiliza el algoritmo de “resilient backpropagation”, el cual junto con Correlación en cascada y Levenburg-marqhardt es uno de los más rápidos para hallar los pesos óptimos. La cantidad de nodos a utilizar en la capa oculta (h) es obtenida mediante la fórmula empírica: h = (2z + 1)(Huang and Foo, 2002). (Browne et al.,2006) utiliza una red con 6 nodos en la capa oculta, mientras que (Deo y Sridhar, 1998) utiliza el algoritmo de correlación en cascada el cual además de obtener los pesos óptimos se encarga de la construcción de la red en la capa oculta para hallar la estructura de red optima.

(Makarynskyy, 2004) experimenta, obteniendo resultados positivos, una técnica especial en la cual se conectan dos redes neuronales. Donde la salida de la primer red es la entrada de la segunda. Según los autores esto permite a la primer red concentrarse en aprender la interacciones complicadas, mientras que la segunda red realiza un refinamiento a las predicciones de la primera.

Diversas comparaciones entre los modelos desarrollados y modelos de predicción ya existentes son realizadas en los trabajos. En (Makarynskyy, 2004) se compara el modelo de red neuronal desarrollado con un modelo estadístico auto regresivo obteniéndose mejores resultados con el primero. (Browne et al.,2006) realiza una comparación entre el modelo de red neuronal y el modelo SWAN, el cual es un modelo numérico para predicción de olas en la zona costera, obteniendo mejores resultados con el primero. También se realizan comparaciones entre un modelo de regresión lineal y uno no lineal, obteniendo mejores resultados con el modelo no lineal.

Todos los trabajos analizados presentan un desempeño satisfactorio en la predicción de olas, de esta manera dejando en evidencia que los métodos de regresión para la predicción de olas son una alternativa viable a los métodos de modelado numérico, y que además los mismos son útiles para refinar los pronósticos dados por los modelos de oleaje de alta mar en la cercanía de la costa.

Nuestro trabajo estará alineado con los objetivos de los trabajos de (Caldwell y Aucan, 2007) y (Browne et al.,2006), (predecir la altura de la ola en la zona de surf a partir de datos de alta mar) donde además de los métodos de regresión lineal y redes neuronales evaluaremos métodos no tenidos en cuenta como los arboles de regresión y las maquinas de soporte vectorial. Luego utilizando el clasificador más eficiente construiremos un sistema que permita al usuario, a partir de un pronóstico más preciso, decidir cuál será la playa que mejor se adapte a sus preferencias.

En la siguiente tabla se resumen los diferentes trabajos analizados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trabajo | Modelo de regresión | Algoritmo de Aprendizaje | Variables predictoras | Variables de respuesta |
| [1] | Red neuronal Multilayer Feedforward | Propagación hacia atrás;  Correlación en cascada;  Gradiente conjugado | Lecturas de la altura de la ola en la boyad de las ultimas N horas. | Valor de la altura de la ola con tiempo de anticipación de N horas. |
| [2] | Red neuronal Multilayer Feedforward | Resilient back- propagation | Lecturas de las ultimas 48 hs de la altura de la ola en la boya | 24 salidas cada una representando el valor de la altura de la ola en un periodo de anticipación. |
| [3] | Red neuronal Multilayer Feedforward | Levenburg-marqhardt | Parámetros del modelo Wave Watch 3 | Observación visual desde la playa de la altura de la ola. |
| [4] | Regresión polinomial | - | Lectura del período y tamaño de ola en una boya en alta mar. | Observación visual desde la playa de la altura de la ola. |

Tabla 2-2. Modelos de regresión utilizados en trabajos relacionados a la predicción de la altura de ola.

Referencias

A.Baranauskas, C.Monard. *Reviewing Some Machine Learning Concepts and Methods.* Univeristy of Sao Paulo.2000.

Bishop, C. *Neural Networks for Pattern Recognition.* Oxford: University Press. 1995.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor 5th Annual ACM Workshop on COLT, pp 144-152, Pittsburgh, PA, 1992. ACM Press.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Monterey, California, U.S.A.: Wadsworth, Inc. 1984.

[3] Browne, M. Strauss, D. Castelle, B. Blumenstein, M. Tomlinson, R. Lane, C.: *Empirical Estimation of Nearshore Waves From a Global Deep-Water Wave Model*. IEEE Geoscience and Remote Sensing Letters. Vol. 3(4), pp. 462-466. 2006.

[4] Caldwell, P. C. and J. P. Aucan. 2007. An empirical method for estimating surf heights from deepwater significant wave height and peak periods in coastal zones with narrow shelves, steep bottom slopes, and high refraction. *Journal of Coastal Research* 23/5:1237–1244.[Bioone](http://www.bioone.org/servlet/linkout?suffix=i1551-5036-25-3-539-b37&dbid=4&doi=10.2112%2F07-0958.1&key=10.2112%2F04-0397R.1)

[1]Deo M.C., Sridhar Naidu C. Real time wave forecasting using neural networks. [Ocean Engineering](http://www.ingentaconnect.com/content/els/00298018;jsessionid=3ldc1bvjg94j9.alice), Volume 26, Number 3, August 1998 , pp. 191-203(13). [Elsevier](http://www.ingentaconnect.com/content/els;jsessionid=3ldc1bvjg94j9.alice).

H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. En M. Mozer, M Jordan, y T. Petsche, editores, Advances in Neural Information Processing Systems 9, paginas 155-161, Cambridge, MA, 1997, MIT Press.

Fausett, L. (1994). *Fundamentals of Neural Networks*. New York: Prentice Hall.

I. Guyon, B. Boser and V. Vapnik. Automatic capacity tuning of very large VCdimension classiers. En Stephen Jose Hanson, Jack D. Cowan, y C. Lee Giles, editors, Advances in Neural Information Processing Systems, volumen 5, paginas 147-155, Morgan Kaufmann, San Mateo, CA, 1993.

J. Han y M. Kamber. Data mining: concepts and techniques, Morgan Kaufmann; 1st edition 2000

Haykin, S. (1994*). Neural Networks: A Comprehensive Foundation.* New York: Macmillan Publishing.

Kubat, M., Bratki, I., y Michalski, R.S(1998). A review of Machine learning Methods

[2]Makarynskyy O. Improving wave predictions with artificial neural networks. Ocean engineering  2004, vol. 31, no5-6, pp. 709-724.Elsevier.

D. Mattera y S. Haykin. Support vector machines for dynamic reconstruction of a chaotic system. In B. Scholkopf, C.J.C. Burges, y A.J. Smola, editores, Advances in Kernel Methods - Support Vector Learning, paginas 211-242, Cambridge, MA, 1999. MIT Press.

T. M. Mitchell (2006). *The Discipline of Machine Learning*

Huang, W., Foo, S., 2002. Neural network modelling of salinity variation in Apalachicola River. Water Research36, 356–362.

MIT resource, Artificial Neural Networks – Lecture 6, <http://ocw.mit.edu/NR/rdonlyres/Sloan-School-of-Management/15-062Data-MiningSpring2003/650A194A-828C-4990-98CE-7EB966628437/0/NeuralNet2002.pdf> (accessed August 29, 2009).

K.R. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. En W. Gerstner, A. Germond, M. Hasler, y J.D. Nicoud, editores, Artificial Neural Networks - ICANN'97, paginas 999-1004, Berlin, 1997. Springer Lecture Notes in Computer Science, Vol. 1327.

Patterson, D. (1996*). Artificial Neural Networks*. Singapore: Prentice Hall.

Quinlan, J. R. Learning with continuous classes. In N. Adams and L. Sterling,

editors, *Proceedings of the Fifth Australian Joint Conference on Artificial Intelligence*,

Hobart, Tasmania. Singapore: World Scientific, pp. 343–348.1992.

B. Scholkopf, C. Burges, Y V. Vapnik. Extracting support data for a given

task. En U. M. Fayyad y R. Uthurusamy, editores, Proceedings, First International Conference on Knowledge Discovery and Data Mining, Menlo Park 1995. AAAI Press.

A. Smola y B. Scholkopf (1998). *A tutorial on support vector regression.*

[Shalizi](file:///C:\Users\esteban\workspace\arfgen\docs\Entregas\Informe%20Final\capitulo%202\Shalizi) Cosma, Regression Trees– Lecture 10, <http://www.stat.cmu.edu/~cshalizi/350-2006/lecture-10.pdf> (accessed August 29, 2009).

M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, y J. Weston. Support vector regression with ANOVA decomposition kernels. En B. Scholkopf, C.J.C. Burges, y A.J. Smola, editores, Advances in Kernel Methods - Support Vector Learning, paginas 285-292, Cambridge, MA, 1999.MIT Press.

V. Vapnik, S. Golowich, y A. Smola. Support vector method for function approximation regression estimation, and signal processing. En M. Mozer, M. Jordan, y T. Petsche, editores, Advances in Neural Information Processing Systems 9, paginas 281-287, Cambridge, MA, 1997. MIT Press.

Witten, I. H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, 2005.