Introduccion

Data Mining - Machine Learning - Regression techniques.

La mineria de datos es interdisciplinaria. Encontrar patrones en los datos. Dentro de las tareas de El aprendizaje de maquina es...Los algoritmos de se dividen por la siguiente taxonomía. El aprendizaje supervisado es aprender: ....CUando se denomina clasificacion. Sino se denomina regresion.Dentro de La mineria de datos tiene como objetivo encontrar patrones en un conjunto de datos. Su fin ultimo es poder utilizar estos patrones para poder obtener conocimiento nuevo a partir de un conjunto de datos. La mineria de datos es aprender de los datos. Existen dos enfoques para aprender a partir de un conjunto de datos. El aprendizaje supervisado y el no supervisado. En el aprendizaje supervisado tenemos un conjunto de variables predictoras y un conjunto de y una o mas variables dependientes. El proposito es encontrar una relacion entre estos dos conjuntos que nos permitan predecir cual sera el valor de la variables dependientes a partir de las variables predictoras. Cuando la variable a predecir es del tipo numerica, la tarea de aprendizaje se denomina regresion, cuando esta variable es nominal la tarea se denomina clasificacion.El aprendizaje de maquina, en ingles Machine Learning, es una rama de la inteligencia artificial especializada en desarrollar algoritmos que permitan a la maquina aprender a partir de un conjunto de datos. Dentro de esta area se han desarrollado numerosos algoritmos, tanto en el grupo aprendizaje supervisado como no supervisado. Dentro del primer grupo tenemos algoritmos de regresion como se clasificacion. Los algoritmos de regresion, buscan obtener una funcion f(x) que establesca la relacion

El aprendizaje supervisado(aprendizaje dirigido) consiste en aprender un caracteristica especifica a partir de un conjunto de caracteristicas explicatorias.

Capitulo 2 - Estado del arte

1. Aprendizaje supervisado

Las técnicas de aprendizaje de maquina pueden dividirse en tres grupos. Aprendizaje supervisado, no supervisado, por retroalimentación.

El aprendizaje supervisado consiste en aprender el mapeo entre un grupo de variables predictivas y una o más variables dependientes. Podemos ver este proceso de aprendizaje como dos fases: 1) aprendizaje, 2) utilización de lo aprendido. En la fase 1 se presentan tuplas(X, Y) conformadas por las variables predictivas(X) junto con la variable dependiente (Y). En esta fase cada técnica ajustará su modelo a las tuplas presentadas, dando como resultado un modelo (F). Este modelo (F) será utilizado en la siguiente fase para predecir el valor de la variable dependiente a partir de una tupla conformada únicamente por las variables predictivas. Las técnicas de aprendizaje supervisado pueden subdividirse en dos clases: clasificación, si la variable dependiente es discreta, y regresión cuando la variable dependiente es un valor continuo.

2. Métodos de regresión:

Regresión Lineal

Cuando el conjunto de variables independientes y dependientes son numéricas, la regresión lineal es una técnica a considerar. La idea es expresar la variable dependiente como una combinación lineal de: las variables independientes y de coeficientes predeterminados:

Y = w0+w1X1+w2X2+...+WnXn

Una vez que se calculan los coeficientes los mismos se pueden utilizar para predecir el valor de la variable dependiente a partir de datos nuevos. Existen diversos métodos para calcular el valor de los coeficientes, siendo el más común el método de los cuadrados minimos.

La forma más simple de regresión lineal involucra una variable dependiente "y" y una única variable independiente "x" de la siguiente manera:

y = b + mx

Donde la variación de y se asume constante, b y m son coeficientes de regresión que especifican la intercepción con el eje Y y la pendiente de la recta respectivamente. Utilizando el método de mínimos cuadrados estimamos la recta que minimiza el error entre los datos actuales y la estimación de la línea.

El conjunto de entrenamiento contiene |D| puntos de datos de la forma (x1, y1), (x2, y2), ... , (x|D|, y|D|). Los coeficientes de regresión, pueden ser estimados usando este metodo con las siguientes ecuaciones:

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

Ejemplo 1,Regresion lineal simple:

Regresión lineal usando el método de mínimos cuadrados. La tabla 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 1

Figura 1

El grafico de la figura 1 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

y = w0+w1x.

Utilizando los datos de la tabla 1 junto con las ecuaciones (1) y (2) obtenemos el valor de los coeficientes.

w0 = 23,2

w1 = 3,5:

Luego la recta que mejor se ajusta a los datos de la tabla 1 estará dada por: Y = 23,6 + 3,5X. Utilizando esta ecuación 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 10 años de experiencia posee un salario de (= 23,6 + 3,5X \*10 )\*1000= $58600

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, mediante el método de mínimos cuadrados. 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.

b. Redes neurales para regresión

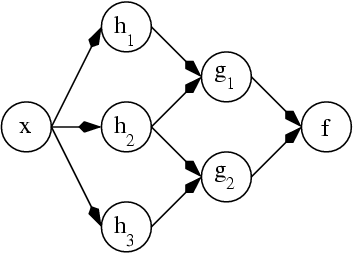
Una Red Neuronal Artificial (Red Neuronal o ANN), es un modelo computacional inspirado en las redes neuronales biológicas. La red está conformada de elementos de procesamiento (neuronas), conexiones entre los elementos de procesamiento y coeficientes (pesos) asociados a cada conexión. Todos estos elementos 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 predicción, clasificación o control en un amplio espectro de campos como finanzas, neurociencia, medicina, ingeniería y física.

*Modelos*

Las redes neuronales artificiales esencialmente son modelos matemáticos que definen una función F : X -> Y. Cada modelo de red neuronal se corresponde con una clase de esta función.

La función f(x) es una composición de otras funciones que a su vez pueden ser definidas como composición de mas funciones. La composición de funciones se puede representar mediante un grafo, dando la apariencia de una red, en donde cada nodo o neurona representa una función y las flechas entrantes representan las dependencias de esa función.

Una composición de funciones usada ampliamente es la suma no lineal de los pesos: ). Donde K es una función cualquiera y es el peso asociado a la función . La función K suele denominarse función de activación. La siguiente figura muestra esta descomposición, con las flechas indicando la dependencia de variables.



En general existen dos arquitecturas de conexión entre las neuronas, las arquitecturas feed-forward o las arquitecturas feed-back:

1. Feed-Forward: En este tipo de arquitectura no se presentan ciclos dentro del grafo de conexiones. Las redes del tipo Perceptron pertenecen a esta categoría.
2. Feed-Back: En este tipo de arquitectura el grafo posee ciclos. De esta manera la red posee memoria y su próximo estado depende no solo de las señales de entrada sino que también de los estados anteriores. Ejemplo de estas son la redes Hopfield.

*Aprendizaje:*

Una de las características más importantes y que más interesante hace a 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\* Ȇ F, que resuelva la tarea específica de manera óptima.

Esta definición implica definir una función de costo C: F→R tal que para la solución optima f\*, C (f\*) <= C (f) para todo f Ȇ F. Es decir que ninguna otra solución tiene un costo menor al costo de la solución óptima. Los algoritmos de aprendizaje realizan una búsqueda en todo el espacio de soluciones para encontrar la solución que menor costo produce.

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.

*Algoritmos de aprendizaje:*

Entrenar una red neuronal significa encontrar una función f\* Ȇ F, siendo F: X→Y, tal que minimice el criterio de costo utilizado. Existen numerosos algoritmos de entrenamiento, en su mayoría son aplicaciones de la teoría de optimización y de estimación estadística. La mayoría de los algoritmos usados para entrenar redes neuronales utilizan alguna variante de la técnica de optimización [Gradient Descent.](http://en.wikipedia.org/wiki/Gradient_descent) Esto es realizado utilizando la derivada de la función de costo con respecto a los parámetros de la red y luego actualizando los parámetros en dirección al gradiente.

Algunos algoritmos de aprendizaje comúnmente utilizados en aprendizaje supervisado son: BackPropagation, Quick Propagation, [Conjugate Gradient Descent](http://en.wikipedia.org/wiki/Conjugate_gradient_method), [Projection Operator](http://en.wikipedia.org/wiki/Radial_basis_function), Delta-Bar-Delta.

*Aplicaciones:*

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, clustering y compresión

Las áreas de aplicación de las redes neuronales incluyen: Sistemas de control(control de vehículos), juegos(backgammon, ajedres), 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 de regresión

d. SVM para regresión

Las maquinas de soporte vectorial(SVM) son un conjunto de métodos relacionados al aprendizaje supervisado usados para clasificación y regresión. SVMs pertenecen  a la familia de clasificadores lineales.

Una propiedad especial es que simultaneamente minimizan el error de predicción y maximizan the geometric margin; por esto son conocidos como: maximum [margin classifiers](http://en.wikipedia.org/wiki/Margin_classifier).

*MVS no lineal*

En la mayoría de los casos los datos no son linealmente separables, son problemas no lineales, la ventaja de MVS es que estos problemas pueden ser resueltos mediante la utilizacion de funciones de nucleo(kernel) las cuales permiten hallar la solución sin la necesidad de aplicar explicitamente algoritmos no lineales. Existen dos casos principales para la solución de datos que no son linealmente separables en el espacio de entradas; el primero, cuando los datos pueden ser linealmente separables pero en un espacio de características y, el segundo, cuando no es posible separar dichos datos linealmente incluso en un espacio de características. Ambos casos se explicarán en las subsecciones siguientes.

*MVS lineal en el espacio de características:*

Existen casos en los cuales los datos, debido a su naturaleza, no pueden ser ajustados mediante una funcioin lineal. Sin embargo, en muchas situaciones los datos, a través de una transformación no lineal del espacio de entradas, pueden ser separados linealmente pero en un espacio de características, en el cual se pueden aplicar los mismos razonamientos que para las MVS lineales.

Aunque la dimensión del espacio de características puede ser bastante grande, para ciertas transformaciones y ciertos espacios de características, se puede calcular el producto escalar usando las denominadas funciones núcleo.

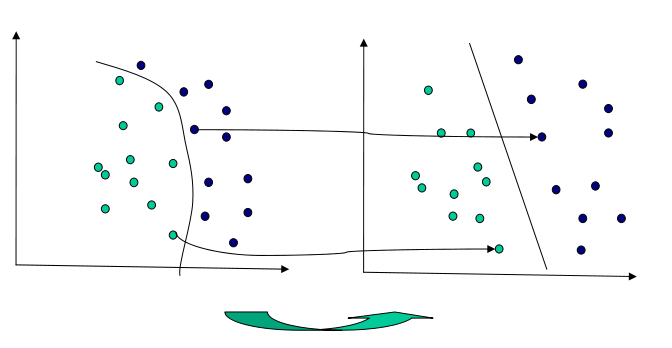
En la figura 5.3 se puede observar un ejemplo donde se puede encontrar un híper-plano óptimo en el espacio de características, a través de un kernel. 

Figura 5.3: Clasificación de MVS para el caso no lineal

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, entre otros más. Matemáticamente, un kernel se puede definir como una función K : X × X → R tal que K ( x, y) = < Φ( x), Φ( y) > , donde Φ es una transformación de X en un cierto espacio de Hilbert I . Algunas funciones de kernel son:

Polynomial (homogeneous): 

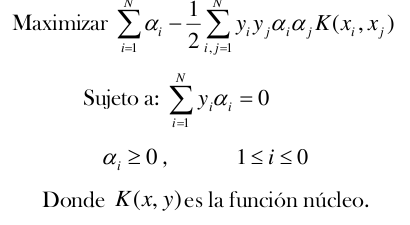
Polynomial (inhomogeneous): 

Radial Basis Function: , for γ > 0

Gaussian Radial basis function: 

[Sigmoid](http://en.wikipedia.org/wiki/Sigmoid_function): , for some (not every) κ > 0 and c < 0

Carreras, Márquez y Romero (2004) afirman que la función núcleo permite calcular el producto escalar < Φ( x), Φ( y) > en el espacio de características sin necesidad de usar ni conocer la transformación Φ . Luego de conocer la idea de lo qué es una función núcleo y qué se puede lograr con ella, se puede proceder a especificar el problema de optimización a resolver para las MVS con margen máximo en el espacio de características. El problema de programación cuadrática con restricciones a resolver es el siguiente:



*MVS con margen blando*

Las MVS con margen blando se refieren a aquellos casos en los cuales los datos no son linealmente separables, ni siquiera en el espacio de características. También es necesario este enfoque cuando los datos pueden ser clasificados linealmente en el espacio de características pero las soluciones son sobreajustadas a los ejemplos y, con esto, se tiende a tener mala generalización. Este modelo es el que precisamente funciona bien con la inclusión de ruido pues resulta ser mucho más robusto que los anteriores modelos explicados.  
 Las MVS con margen blando incorporan variables de holgura, ξi,  al modelo permitiéndole a las máquinas seleccionar un clasificador con cierto margen de error pero que a su vez tienda a tener una mejor generalización. En otras palabras, estas variables, cuyo valor será siempre positivo o igual a cero, permitirán que con cierto margen no se cumplan las restricciones estrictamente. Además, será necesario incluir tantas variables de holgura como datos se tengan, pues para cada instancia de datos se tendrá su respectiva variable de holgura.

La inclusión de variables de holgura a las restricciones en el modelo debe equilibrarse incluyendo en la función objetivo un término de regularización que depende de dichas variables. Este término es el que se conoce como el parámetro C, el cual determina la holgura del margen blando; es decir, el parámetro C es el que permite al clasificador un cierto margen de error en el momento de clasificación. Con esto, el problema a optimizar resulta muy similar al de los modelos con margen máximo, con la diferencia de que en este modelo de margen blando se incluye un término en la función objetivo dependiente del parámetro C y las variables de holgura y, a su vez, a cada restricción se le añadirá las respectivas variables de holgura. Además, se incluirá una nueva restricción que es la que limita el valor del parámetro C.

  
  
Donde ξi >= 0. El híper-plano de separacion optima queda determinado por el vector w que minimiza la funcion:.donde C es un valor dado sujeto a las restricciones de la ecuacion 2.

*Maquinas de Soporte Vectorial para regression :*

El concepto de híper-plano de margen máximo solo puede ser aplicado para clasificación. Sin embargo, se han desarrollado maquinas de soporte vectorial para predicción numérica que comparten muchas de las propiedades descriptas en el caso de clasificación: estas producen un modelo que puede ser expresado en término de unos pocos vectores y puede ser utilizado para problemas no lineales usando una función de núcleo.

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.

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 muestra un problema de regresión con un atributo, una clase numérica, y ocho instancias. Es 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 X 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.

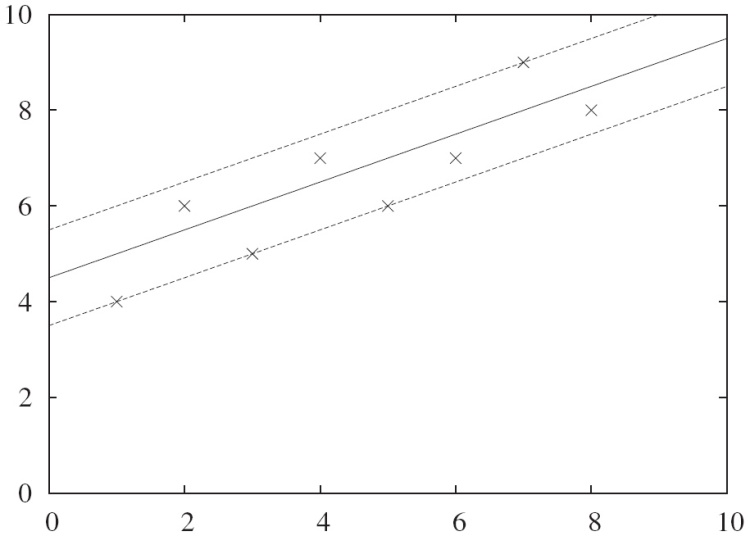
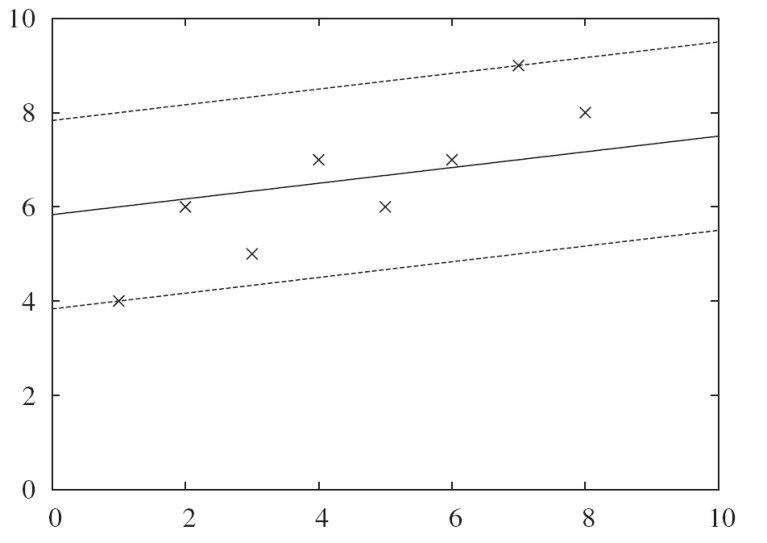
El valor de ɛ controla que tan cerca la función **aproximara(fit?)** los datos. Un valor demasiado grande producirá un predictor sin sentido – en el caso extremo, cuando 2ɛ excede el rango de valores de la clase de los datos de entrenamiento, la línea de regresión es horizontal y el algoritmo solo predice el **valor promedio** (mean) de 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 “trade-off” entre el error de predicción y la horizontalidad del tubo. En la **figura** **¿?** ɛ 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:

Como en el caso de clasificación el producto escalar puede ser remplazado por una función de núcleo para problemas no lineales. 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. Como en la clasificación todos los demás puntos se les asigna coeficiente 0 y pueden ser eliminados de los datos de entrenamiento sin cambiar la salida del proceso de aprendizaje.

El algoritmo busca simultáneamente una minimización del error y una maximización de la horizontalidad de la función de regresión. Sin embargo, cuando algunos puntos no encajan en el tubo**(Figu)** y no existe un tubo con error igual a 0, estamos en presencia de un trade-off entre el error en la predicción y la horizontalidad del tubo. Este “trade-off” es controlado 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 **encajara(fit?)** los datos. En el caso degenerado (ɛ=0) el algoritmo simplemente realiza una regresión de *error absoluto minimo* 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 “trade-off” para hacer y el algoritmo obtiene el tubo mas horizontal que encierra a los datos indiferentemente del valor de C.

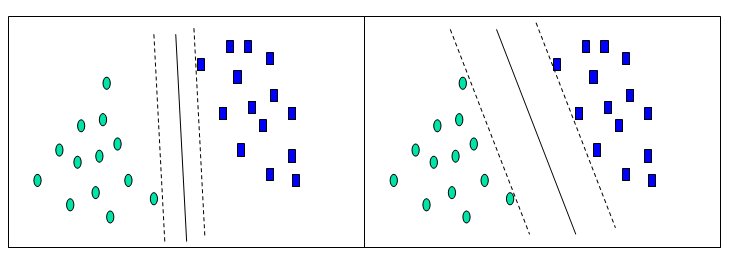




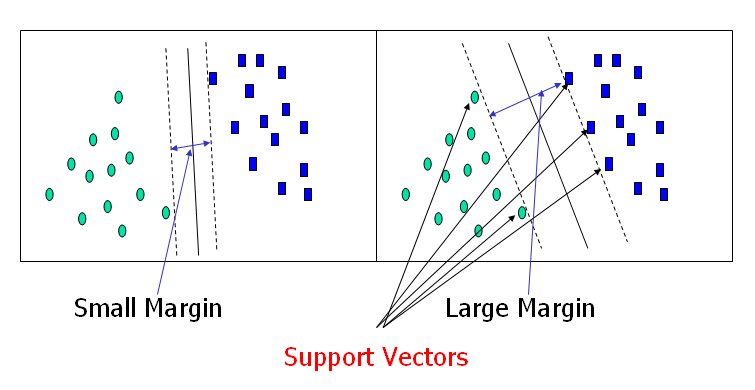
Los datos de entrada se pueden ver como dos conjuntos de vectores en un espacio de n dimensiones, una SVM construye un híper-plano separador en ese espacio que maximizara el margen entre los dos conjuntos de datos. Para calcular dicho margen, se construyen otros dos híper-planos paralelos a este uno a cada lado del original y se van empujando hacia los grupos de datos, en general con márgenes mas grandes se logra mejorar el error del clasificador.  
Los modelos SVM están estrechamente relacionados con las redes neuronales. De hecho, un modelo de SVM que utiliza un sigmoid kernel es equivalente a una red neuronal de dos capas.

Motivation

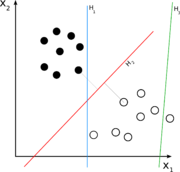
Clasificar datos es una necesidad comun en aprendizaje de maquina. Supongamos algunos datos pertenecientes a dos clases distintas (Espacio de 2-dimensiones), y el objetivo es decidir a que clase pertenece cada dato y lograr separarlos con un hyperplano uni dimensional (una recta). En este caso los vamos a representar con ovalos y rectangulos. Hay un nro infinito de híper-planos posibles pero solo uno es el optimo que maximiza los margenes.



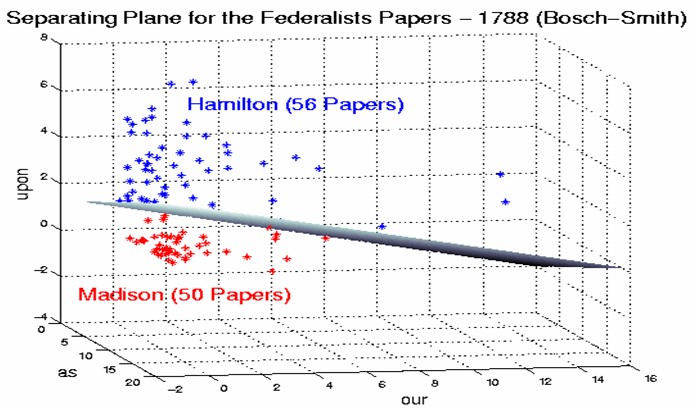
En ambas imágenes, en la figura anterior, vemos un híper-plano de separación posible pero en la segunda vemos el optimo. Las líneas punteadas muestran la distancia entre el híper-plano y los puntos más cercanos a él. Los puntos ubicados al lado de las líneas punteadas que delimitan el margen máximo de dicho híper-plano se denominan vectores de soporte, y se destacan en la figura siguiente:



Estamos interesados adicionalmente en encontrar una máxima separación (margen) entre las dos clases de datos. Es decir que queremos encontrar el híper-plano cuya distancia a un dato sea la mayor. Si encontramos este híper-plano lo llamaremos híper-plano de máximo margen.

  
H3 (verde) no separa las dos clases. H1 (azul) las separa con un margen pequeño y H2 (rojo) lo hace con el máximo margen.

Si todos los análisis consistieran en dividir dos clases distintas de datos, todo resultaría muy sencillo. Por lo general esto no sucede por lo que SVM tiene que lidiar (a) con más de dos variables predictivas, (b) separando los datos con curvas no lineales, (c) manejando casos donde los datos no puedes ser completamente separados y (d) manejando clasificaciones con más de dos categorías.  
  
En el ejemplo previo solo teníamos dos variables predictivas, y pudimos exponer los datos en un plano de dos dimensiones. Si tuviéramos 3 variables predictivas usaríamos un cubo de 3 dimensiones. Datos en un plano de 2 dimensiones pueden separarse con un un híper-plano unidimensional, similarmente datos en un cubo de 3 dimensiones pueden separarse con un híper-plano bidimensional:



A medida que vamos agregando variables predictivas(atributos), los datos pueden ser representados en un N-dimensional space, y un (N-1)dimensional hyperplane pueden separarlos.

En el caso de SVM, un dato es visto como un vector p-dimensional, y si queremos podemos separar esos puntos con un híper-plano de p-1-dimensiones. Esto es llamado clasificación lineal. hay muchos híper-planos que pueden clasificar los datos, sin embargo.

Formalization

Dado algunos datos de entrenamiento, un conjunto de puntos de la forma:



donde ci es tanto 1 or −1, indicando la clase del punto  . Cada  es un p-dimensional [real](http://en.wikipedia.org/wiki/Real_number) vector. Queremos ofrecer el maximum-margin hyperplane que divide los puntos con clase ci = 1 de los ci = − 1. Cualquier híper-plano puede ser escrito como un conjunto de puntos  que satisface:



El vector  es un [normal](http://en.wikipedia.org/wiki/Surface_normal) vector: y perpendicular al híper-plano. El parametro  determina el desplazamiento del híper-plano desde el origen a lo largo del normal vector .

Queremos elegir el  y b para maximizar el margen entre los híper-planos paralelos para que esten los mas separados posible manteniendo los conjuntos de datos separados. Estos híper-planos pueden ser descriptos con las ecuaciones

  
y

Notar que si los datos de entrenamiento son linealmente separables podemos seleccionar los dos híper-planos del margen de manera que no queden datos entre estos y luego intentar maximizar su distancia. Usando geometria, podemos encontrar que distancia entre estos dos híper-planos es , por lo tanto queremos minimizar . Al mismo tiempor tenemso que prevenir que los datos caigan dentro del este margen por lo que agregaremos las siguientes restricciones : para cada  i

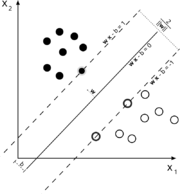
 para  de la primera clase o  
 para  de la segunda.

Esto se puede reescribir asi:



Podemos poner esto juntos para obtener el problema de optimizacion:

elegir  para minimizar sujeto a 





Maximum-margin hyperplane and margins for a SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Primal Form

El problema de optimizacion presentado en la seccion anterior es dificil de resolver porque depende del valor absoluto de |w|. Afortunadamente es posible alterar la ecuacion sustituyendo  ||w|| por  sin cambiar la solucion (the minimum of the original and the modified equation have the same w and b). Esto es un [quadratic programming](http://en.wikipedia.org/wiki/Quadratic_programming) (QP) [optimization](http://en.wikipedia.org/wiki/Optimization_(mathematics)) problem. Mas claramente,

minimizar , sujeto a .

El factor 1/2 es usado por convenienza matematica. This problem can now be solved by standard quadratic programming techniques and programs.

Dual Form

La escritura de la regla de clasificacion en su forma dual sin restricciones revela que el maximum margin hyperplane asi como la tarea de clasificacion es solo una funcion de los support vectors, los datos que se encuentran en el margen. The dual of the SVM can be shown to be:

 subject to , and 

where the α terms constitute a dual representation for the weight vector in terms of the training set:



3. Problema de Predicción de oleaje

El problema de prediccion de oleaje es el siguiente:

4. Trabajos relacionados a la predicción de oleaje (con redes neurales u otros métodos)

Los trabajos relacionados a la prediccion de oleaje: