Revista Tecnocientífica URU
Universidad Rafael Urdaneta
Facultad de Ingeniería
No. 14 Enero - Junio 2018
Depósito Legal: PPI 201402ZU4464
ISSN: 2343-6360

Estructura matemática del modelo factores latentes fortalecidos

Eddy Jackeline Rodríguez
Escuela de Ingeniería Industrial. Facultad de Ingeniería.Universidad del Zulia. Maracaibo Venezuela.
Correo electrónico: eddyjackeline@yahoo.es

Recibido: 14/11/2017 Aceptado: 03-07-2018

Resumen

El método de factores latentes fortalecidos busca crear una serie de características ortogonales que abarcan todo el espacio de interés con el propósito de reducir la dimensionalidad del espacio de características haciéndolas ortogonales, esto puede aumentar drásticamente la velocidad de convergencia del algoritmo. De esta forma deben construirse menos funciones para obtener la misma disminución en la función de costo. Este conjunto de características, relativamente pequeñas en comparación con los datos originales, se pueden utilizar para diversos propósitos, tales como, modelos predictivos, visualización y detección de variables. El objetivo de este trabajo es describir y analizar la estructura matemática del algoritmo de los factores latentes fortalecidos, enfocado en la obtención de un espacio ortogonal de entrenamiento y en la minimización de la función de costo diferenciable, para esto se parte de la revisión bibliográfica de trabajos de investigación Bühlmann, Hothorn [1]; Duffy, Helmbold [2]; Kramer, [3]; Momma And Bennett [4].

Palabras clave: Factores latentes fortalecidos, características ortogonales, estructura matemática, Boosting.

Mathematical structure of the model latent factors strengthened

Abstract

The latent strength method seeks to create a series of orthogonal features that span the entire space of interest with the purpose of reducing the dimensionality of the feature space making them orthogonal, this can drastically increase the convergence speed of the algorithm. In this way, fewer functions must be built to obtain the same decrease in the cost function. This set of features, relatively small compared to the original data, can be used for various purposes, such as predictive models, visualization and variable detection. The objective of this work is to describe and analyze the mathematical structure of the algorithm of strengthened latent factors, focused on obtaining an orthogonal training space and minimizing the function of differentiable cost, for this is part of the bibliographic review of Research work Bühlmann, Hothorn [1]; Duffy, Helmbold [2]; Kramer, [3]; Momma And Bennett [4].

Keywords: Boosting latent factors, orthogonal features, mathematical structure, boosting.

Introducción

Un problema común en el campo de la ingeniería es estimar relaciones lineales y no lineales entre conjuntos multivariables. Estos conjuntos se agrupan en variables dependientes o respuesta Y de tamaño nxq , y en variables independientes X de tamaño nxp que pueden ser muy numerosas (en algunos casos miles), en comparación con el número de observaciones que son pequeñas, lo que involucra trabajar con matrices mal condicionadas. Este problema de modelos de regresión multivariables se puede definir como:

$$Y = \alpha + X\beta + \in \text{ , } \alpha\ (nxq) intercepto, \beta\ (pxq)\ regresores\ y \in (nxq)\ error$$

$$Y = \alpha + X\beta +$$

Existen diferentes métodos para abordar y solucionar estos tipos de problemas, tales como: mínimos cuadrados parciales (PLS), máquina de vectores de soporte (SVM), regresión de componentes principales (PCR), resolución multivariante de curvas (MCR), factores latentes fortalecidos (FLF), entre otros.

En esta investigación se estudia el método de factores latentes fortalecidos que es usado para problemas de modelos de calibración y clasificación, Freund, Schapir [5], este método se fundamenta en la construcción de un algoritmo fortalecido que forma un conjunto de características llamadas hipótesis o características para encontrar un espacio de respuesta. En el caso de calibración o regresión se parte de: $Y \approx Tc$ y en el caso de clasificación $Y^T Tc > 0$.

Los Factores latentes fortalecidos crean un algoritmo que forma un conjunto de variables ortogonales que explican la respuesta. Escogiendo un *AnyBoost*[1] para producir hipótesis ortogonales, se puede forzar el *AnyBoost* para producir variables que factorice y explique el espacio de entrada. Este método usa funciones diferenciables para construir un espacio que contenga solo las variables necesarias para predecir el modelo de estudio. La meta es crear un conjunto de características ortogonales que expliquen la respuesta de acuerdo a alguna función de costo.

En la actualidad existe diferente bibliografía que presenta al método FLF como herramienta aplicada en la solución de problemas en la industria Bühlmann, Hothorn [1], Duffy, Helmbold [2], Kramer, [3], pero poca información como desarrollo de algoritmo. Es por ello que en este trabajo se explican las diferentes etapas del algoritmo del método FLF, describiendo y analizando la estructura matemática de las mismas, lo cual permitirá, en posteriores investigaciones, adaptar la formulación del algoritmo para la solución de otras aplicaciones.

Fundamento de los factores latentes fortalecidos

La estructura de aprendizaje general trata de estimar una relación funcional, basada sobre un conjunto finito de observaciones:

$$F : X \rightarrow Y \tag{1}$$

Donde X es un subconjunto de R^P e Y puede pertenecer a los reales si el estudio es de regresión o $\{\pm 1\}$ si el problema es de clasificación.

Dada una muestra $z = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ se estima a (1) usando una función $f \in F$.

El problema critico es escoger la clase de la función F con la cual se estimará a f, si F es muy complejo, habrá sobreajuste, la función se adapta perfectamente a los datos de entrenamiento dando un error de aproximación bajo, pero el error de estimación será alto. Una forma de evitar esto es restringir a sí mismo a las llamadas funciones “simples” o “débiles” de baja complejidad (por ejemplo polinomios de bajo grado o suavizado splines con una alta cantidad de suavidad). Sin embargo, se debe

1 AnyBoost: algoritmo fortalecido, se explica en detalle posteriormente en esta investigación.

tener presente que si F es demasiado simple, podría haber bajo ajuste, por lo tanto, hay que encontrar un equilibrio entre aproximación y estimación.

Los factores latentes fortalecidos se sustentan en *Boosting*, el cual consiste en la combinación de hipótesis débiles $\left(f_m\right)$,a través de una simple función de clase F , de tal manera que, se formen una hipótesis fortalecida:

$$g_m(x)= \sum_{m=1}^{M} \alpha_m f_m(x) \tag{2}$$

Donde f_m son llamadas la base de las hipótesis, M es el número de iteraciones en Boosting, α_m son los pesos y $\alpha_m f_m$ se eligen de forma adaptativa con los datos.

En general el método de *Boosting,* empieza con una simple función $f \in F$ y combina estas funciones en respuesta a los datos. De esta forma es posible que el estimador que proviene de la combinación resulte más complejo que las funciones individuales extraídas de F , para superar este inconveniente se debe elegir cuidadosamente las funciones de f_m y los coeficientes α_m , de manera que, el aumento de la complejidad sea menor en comparación con la mejora en la precisión.

El método FLF es un algoritmo fortalecido que combina propiedades de las componentes ortogonales con la flexibilidad del conjunto general *Boosting*. La meta es crear un conjunto de características ortogonales que expliquen la respuesta de acuerdo a alguna función de costo. Este método, modifica al *Boosting* reforzando las hipótesis débiles con las componentes ortogonales.

El algoritmo *Boosting* es equivalente al método de descenso del gradiente proyectado en un espacio de hipótesis, *Mason and Baxter* [6]. De esta manera, los FLF, pueden ser vistos como un *Boosting* ortogonal, que construye en cada iteración la característica que es ortogonal a las funciones anteriores maximizando el producto interno.

La estructura de los factores latentes fortalecidos se desarrolla a partir de un *Boosting* particular llamado *AnyBoost* como se describirá a continuación, Momma and Bennett [4].

AnyBoost:

Es un algoritmo fortalecido que construye un conjunto de características llamadas hipótesis para encontrar un espacio de respuesta, según sea el caso:

Problema de regresión: $Y \approx Tc$ (3)

Problema de clasificación: $Y^T Tc > 0$ (4)

Donde Y es la matriz respuesta, $T \in \Gamma$ son las hipótesis y c es el vector formado por los coeficientes de la combinación.

El *AnyBoost* construye una combinación lineal de hipótesis para adecuar la respuesta por el desarrollo de la función del gradiente de descenso o espacio de hipótesis.

Γ es un conjunto de funciones de valores reales que son las hipótesis, el espacio de Γ forma un espacio de funciones lineales y el producto interno en este espacio de funciones está definido por:

$$T^T f \equiv \sum_{i=1}^{m} t(x_i) f(x_i) \tag{5}$$

Donde es un vector de tamaño, m, $t_i = t(x_i)$

Se establece una combinación lineal de la hipótesis:

$$\sum_{i=1}^{N} t_i c_i = Tc \tag{6}$$

Los vectores c contienen los coeficientes de la combinación.

El objetivo es encontrar los elementos, t que pertenece al espacio de Γ, que aproximadamente minimice alguna función de costo $l(y, f)$.

AnyBoost lleva a cabo esto aplicando el gradiente de descenso en el espacio de hipótesis. Que representa el gradiente de la función de costo en el espacio función.

$$\nabla l(y, f) = \nabla_f l(y, T_c) \tag{7}$$

Idealmente el espacio de Γ es el mismo de Y, pero en general no es el caso, por ello el funcional lineal es minimizado con respecto a la función de costo para adecuar la respuesta.

Dado la función actual

$$f = Tc \tag{8}$$

Se crea una dirección de descenso en el espacio de hipótesis t con una función con producto interno positivo, con el gradiente negativo que debe ser una dirección de descenso: $\left(-\nabla l(y, Tc)^T t_{i+1}\right)$

En *AnyBoost,* un algoritmo de entrenamiento débil es usado para construir una hipótesis que aproximadamente maximice el producto interno de la hipótesis con el gradiente negativo en cada iteración. La hipótesis es agregada con un apropiado tamaño de paso a la función actual: $Tc + ct$

El algoritmo termina si el entrenamiento débil falla al producir una hipótesis débil que es una dirección de descenso que indica convergencia, o si este alcanza el máximo número de iteraciones.

Algoritmo (Anyboost):

Dada una clase de hipótesis débiles Γ, y la función $l(y, t)$ con gradiente $\nabla l(y, t)$, el entrenamiento para encontrar $t \in \Gamma$ maximizando $u^T t$ consiste en:

1. Dado que $f = contante\ (hipotesis\ nula)$

2. Computar $u1 = -\nabla l(y, f)$

3. Iniciar el ciclo para $i = 1\ hasta\ N$

4. dado $t_i \in \arg\ max_{t \in r} u_i^T t_i$

5. $si - u_i^T t_i < 0$ entonces, regresar a f

6. Escoger c_i para reducir $l\left(y, f + c_i t_i\right)$

7 Dado $f = \sum_{j=1}^{i} t_j c_j$

8. Computar $u_i + 1 = \nabla l\left(y, f\right)$

9. Finalizar el ciclo

10. retornar a f

Se puede cambiar en el paso 4 la función de costo y/o en el paso 6 el algoritmo a optimizar con el tamaño del paso.

AnyBoost con hipótesis débil ortogonal:

Es una variante del *AnyBoost* cuya meta es crear un conjunto de variables ortogonales que expliquen la respuesta, en este caso recibe el nombre de *orthoAnyBoost,* por ejemplo en el caso de regresión:

$$Y \approx Tc = XVc \tag{9}$$

$$T^T T = 1 \tag{10}$$

Donde V es una matriz proyección con $v_i, i = 1, ..., N$ como sus vectores columnas, de tal forma que minimice la función de costo.

Algoritmo (orthoAnyBoost):

Se basa en el algoritmo de *AnyBoost* cambiando los siguientes pasos:

4. dado $t_i \in \arg \quad max_{t \in r} u_i^T t_i$

$sujeto \; a : t^T t_j = 0, \qquad j = 1, ..., i-1$

6. optimizar c para reducir $l\left(y, \sum_{j=1}^{i} t_j c_j\right)$

Este simple cambio tiene varias ramificaciones:

- Se puede usar ahora una hipótesis más poderosa mientras que la hipótesis del sub-problema está regularizada.

- En segundo lugar la búsqueda ya no es mediante el algoritmo del gradiente de descenso. Se ha transformado el algoritmo de sub-espacio interno, similar, al algoritmo de gradiente conjugado no lineal. En cada iteración el algoritmo computa la solución óptima sobre el sub-espacio actual.

El método de sub-espacio tal como el gradiente conjugado puede ser mucho más rápido que el método de los gradientes, particularmente para problemas mal planteados.

En este caso los coeficientes de regresión deben ser nuevamente fijados en cada iteración, lo cual no se hace en el *boosting.*

La estructura del *orthoAnyBoost* puede ser aplicada para hipótesis lineales de la forma:

$$t_k = x_k^T v \tag{11}$$

En el espacio de entrada.

Para aplicar *orthoAnyBoost* a la hipótesis lineal se requiere que la hipótesis débil del paso 4 y la hipótesis fuerte calculada en el paso 6 sean modificadas adecuadamente. La gran ventaja de la hipótesis lineal es que la función lineal ortogonal óptima encontrada en el paso 4 puede ser eficientemente computada remodelando el problema dentro del espacio nulo de las restricciones usando procedimientos de deflación.

Al aplicar ortogonalidad en el espacio lineal se requiere que la nueva hipótesis sea ortogonal a toda restricción previa de las restricciones de igualdad lineal en el espacio de hipótesis. La hipótesis lineal, $t = Xv$, en el paso 4 en *orthoAnyBoost* se reduce a:

$$max_v \quad u_i^T Xv \tag{12}$$

$$sujeto\ a : t^T t_j = 0, j = 1,...,i-1$$

Esta combinación *de orthoAnyBoost* con hipótesis lineales es lo que genera al método de factores latentes fortalecidos.

Algoritmo de los factores latentes fortalecidos

Antes de establecer el algoritmo para los factores latentes fortalecidos se debe tomar en cuenta lo siguiente:

- Se requiere que la nueva hipótesis sea ortogonal a toda restricción previa de las restricciones de igualdad lineal en el espacio de hipótesis.

La hipótesis lineal, $t = Xv$, en el paso 4 en *orthoAnyBoost* se reduce a:

$$max_v u_i^T Xv$$

$$sujeto\ a : t^T t_j = 0, j = 1,...,i-1$$

La igualdad lineal se elimina mapeando el problema dentro del espacio nulo de las restricciones lineales.

De manera que:

Defina $T_i = [t_1,...,t_1]$ y defina a Z_i como la matriz de espacio nulo para la matriz $T_i^T X$.

Entonces para algún w:

$v_i = Z_{i-1} w_i$, lo que implica: $T_i^T X v_i = 0$.

En la iteración $i = 1, Z_0$ es definida como la matriz identidad I y $X^1 = X$.

El problema puede ser re-parametrizado como:

$max_w u_i^T XZ_{i-1} w$

sujeto $a : w^t \ w = 1$

Comenzar el algoritmo con la hipótesis débil igual a una hipótesis constante, $t \quad \alpha \quad e$, un vector de unos.

- El tamaño del paso sobre la columna constante es obtenido resolviendo:

$min_c l(y, ce)$

La primera hipótesis es entonces $f = ce.$

Se considera el primer $c = \mu_y$

La data de la matriz es deflactada por el escalar: $t_0 = \frac{e}{\|e\|}$

$X_1 = X - t_0 t_0^T X = X - e\mu_X^T$

Esta deflación centra cada valor.

Algoritmo FLF:

Introduzca la matriz X, la respuesta y, número de variables latentes N

1. Calcule $c = \mu_y = \arg\min_{\mu_y} \ l(y, \mu_y e)$

Dado $\mu_x = \frac{1}{m} X^T e$ *deflactar a* X *para obtener* $X_1 = X - e\mu_X^T$

calcular: $u_1 = -\nabla l(y, \mu_y e)$

2. Iniciar ciclo, para $i = 1$ *hasta N*

3. Calcular la solución optimal de $w_i = X_i^T u_i$

4. Calcular la hipótesis lineal $t_i = X_i w_i$, t_i *normalizada*

T e*s la matriz que almacena a los* $t : T = [T \quad t]$

5. Deflactar a $X : p_i = X_i^T t_i \quad X_{i+1} = X_i - t_i p_i^T$

6. Calcular $c = \arg\min_{(c)} l(y, Tc)$

7 Calcular el gradiente negativo: $u_{i+1} = -\nabla l\left(y, Tc\right)$

8. Finalizar el ciclo

9. Hipótesis final $T(x) = (x - \mu_x)^T W (P^T W)^{-1}$

P *y* W *son matrices que contienen los vectores* p *y* w *obtenidos en las iteraciones*

10. Calcular los coeficientes en el espacio original: $g = W\left(P^T W\right)^{-1} c$

11. Función final $f_{(x)} = (x - \mu_x)^T g$

Estructura matemático del algoritmo FLF

El método de factores latentes fortalecidos tiene por objetivo encontrar un modelo que describa el comportamiento de los datos, entrena un conjunto de hipótesis o funciones ortogonales usando la minimización de una función de costo. La ejecución de este algoritmo involucra procesos matemáticos que se detallan a continuación:

- En primer lugar se elige una función de costo $l(y,f)$ que debe ser diferenciable, asegurando que existe el gradiente de esa función $\nabla l(y,f)$. Es de notar que en factores latentes fortalecidos:

$f = Tc$, con el primer $c = \mu_y$ y el primer $t = e$.

- El trabajo de entrenamiento de los datos se realiza en un espacio de función lineal. Los datos se proyectan en este espacio siguiendo la dirección del vector gradiente, que minimiza la función de costo, el problema consiste entonces en:

A partir de:

$u_i = -\nabla l(y,f)$ y $w_i = X_i^T u_i$

Encontrar la hipótesis lineal: $t_i = X_i w_i$.

Lo que es posible al encontrar la solución del problema optimal:

$$\max_w u_i^T X Z_{i-1} w$$

$$\textit{sujeto a}: w^t w = 1$$

Usando el método de multiplicadores de *Lagrange*[2], Thomas and Finney [7], para resolver este problema, queda:

Como funciones a evaluar: $F(w) = u_i^T X Z_{i-1} w \quad g(w) = (w^T w - 1) = 0$

El sistema de ecuaciones:

$$\nabla F = \lambda \nabla g \qquad g = 0$$

$$\begin{cases} u_i^T X Z_{i-1} = \lambda w^T \\ w^T w - 1 = 0 \end{cases}$$

Despejando w de la primera ecuación del sistema: $w = \dfrac{z_{i-1}^T X^T u_i}{\lambda}$

Al En general, en la iteración el problema se acerca a la solución en: $w_i = \dfrac{Z_{i-1}^T X^T u_i}{\left\| Z_{i-1}^T X^T u_i \right\|}$

Luego se sustituye w_i para computar t_i.

---

2 Método de los multiplicadores de Lagrange: suponga que las funciones $F(x_1,\ldots,x_n)$ y $g(x_1,\ldots,x_n)$ son diferenciables. Para encontrar los valores máximos y mínimos locales de F sujetos a la restricción $g = 0$ encuentre los valores de , que simultáneamente satisface las ecuaciones:

$\nabla F = \lambda \nabla g \;\; g(x_1,\ldots,x_n) = 0$ con λ como el multiplicador de Lagrange.

Una equivalente forma para resolver el anterior problema consiste en usar la restricción de ortogonalidad del método *orthoAnyBoost* y la matriz de espacio nulo, Z_i, de $T_i^T X$:

Defínase a: $X_i = XZ_{i-1}$

Al evaluar la restricción de ortogonalidad:
$T_i^T X v_i = 0$

Con $T_i = [t_1 ... t_{i-1}]$ y considerándose para algún , se puede escribir: $v_i = Z_{i-1} w_i$

sustituir w en la segunda ecuación del sistema: $\lambda^2 = \left\| Z_{i-1}^T X^T u_i \right\|^2$

Lo que implica que: $T_i^T X v_i = 0$

Es equivalente a: $T_i^T X Z_{i-1} w_i = 0$

Luego al sustituir a $X_i = XZ_{i-1}$ queda: $T_i^T X_i w_i = 0$

Entonces la hipótesis lineal es: $t_i = X_i w_i$

Debido a que por ortogonalidad: $T_i^T t_i = 0$

Una ventaja de usar esta forma para el problema es que la solución óptima: $w_i = \alpha X_i^T u_i$

Se obtiene de sustituir a $Z_{i-1}^T X^T = X_i^T$ en:

$$w_i = \frac{Z_{i-1}^T X^T u_i}{\left\| Z_{i-1}^T X^T u_i \right\|}$$

- Los vectores t son ortogonales entre sí: para demostrar esto se trabaja con la matriz residual de X , al someter a X a un proceso de deflación en cada iteración:

Con $p_i = X_i^T t_i, X_{i+1} = X_i - t_i p_i^T$

El factor de ortogonalización es aplicado, de forma que: $T_i^T X_{i+1} = 0$ puede ser demostrado por inducción:

Considere en particular, la segunda iteración, donde $t_1 = \dfrac{Xw_1}{\|Xw_1\|}$

$$X_2 = (I - t_1 t_1^T) X = X - \frac{Xw_1}{\|Xw_1\|} t_1^T X = X(I - \frac{w_1}{\|Xw_1\|} t_1^T X)$$

Defina: $Z_1 = (I - \dfrac{Xw_1}{\|Xw_1\|} t_1^T X)$

Entonces: $t_1^T X_2 = t_1^T X Z_1 = 0$

Tal que la solución optimal del problema para la segunda iteración satisface la restricción de ortogonalidad.

En general para la i-ésima iteración asuma:

$X_1 = XZ_{i-1}$

$t_J^T XZ_{i-1} = 0 \quad j = 1,\ldots,i-1$

En la iteración $i+1$

$$X_{i+1} = (I - t_i t_i^T)X_i - \frac{X_i w_1}{\|X_i w_1\|} t_1^T X_i = XZ_{i-1}\left(I - \frac{w_1}{\|X_i w_1\|} t_1^T X_i\right)$$

Por la suposición,

$$t_J^T XZ_{i-1} = t_J^T XZ_{i-1}\left(I - \frac{w_1}{\|X_1 w_1\|} t_i^T X_i\right) = 0,\ j = 1,\ldots,i-1$$

Demostrándose la ortogonalidad.

- Encontrar los valores de , a través de la minimización de la función: $c = argmin_{(c)} l(y, Tc)$

Para la solución de este problema se parte de la definición de punto mínimo relativo y las condiciones necesarias de primer orden y segundo orden, Luenberger [8]:

Definición de punto mínimo relativo: un punto $c^* \in R^n$ es un punto mínimo relativo o punto mínimo local de $l(y,c) \in R^n$, si existe un $\varepsilon > 0$ tal que $l(y,c) \geq l(y,c^*)$ para toda $c^* \in R^n$ a una distancia de c^* menor que ε, es decir, $c \in R^n$ y $\left| c - c^* \right| < \varepsilon$.

Condición necesaria de primer orden: sea, $l(y,c) \in c^1$ una función en. R^n Si c^* es un punto mínimo relativo de $l(y,c)$ en, R^n entonces para cualquier $d \in R^n$ que sea una dirección factible[3] en c^*, resulta $\nabla l(y,c^*)\, d \geq 0$.

Condición necesaria de segundo orden: sea $l(y,c) \in C^2$, una función en. R^n Si c^* es un punto mínimo relativo de $l(y,c)$ en R^n, entonces para cualquier $d \in R^n$ que sea una dirección factible en c^* resulta:

$\nabla l(y,c^*)\, d \geq 0$

Si $\nabla l\left(y,c^*\right) d = 0$, entonces $d^T \nabla^2 l(y,c^*) d \geq 0$

Donde $\nabla^2 l\left(y,c^*\right)$ representa la segunda derivada evaluada en c^*

El cálculo de μ_y que corresponde al primer valor c se obtiene de considerar:

$$\nabla l(y,\mu_y) = \frac{\partial l(y,\mu_y)}{\partial \mu_y} = 0$$

3 Dirección factible: dado un $c \in R^n$, se puede decir que un vector d es una dirección factible en c si existe si existe algún $\alpha^* > 0$ tal que $\left(c + \alpha d\right) \in R^n$ para todo α, con $0 \leq \alpha \leq \alpha^*$.

Debido a que el punto mínimo relativo de una función debe cumplir con que la primera derivada evaluada en ese punto debe ser cero.

Para el cálculo de los siguientes valores de c, se procede con alguna técnica de minimización de programación lineal (la más común el método de Newton) junto con el punto de partida $c = \mu_y$ La técnica de minimización consistirá en encontrar en cada iteración el nuevo valor de c que minimiza a $l(y,c)$ cumpliendo las condiciones de primer y segundo orden:

$$c_{i+1} = c_i - \alpha_i d_i$$

Donde α_i es un escalar no negativo y d_i es una dirección de descenso.

En particular para el método de Newton: $\alpha_i d_i = \left(\nabla^2 l(y,c_i)\right)^{-1} \nabla l(y,c_1)$

Conclusiones

- El método de factores latentes fortalecidos puede ser usado en problemas de regresión y problemas de clasificación aplicando una función de costo adecuada.

- El uso de una función de costo con las propiedades de diferenciabilidad y convexidad asegura que el método *Boosting* y sus variantes como el *AnyBoost* y los factores latentes fortalecidos converjan a la solución óptima.

- El desarrollo matemático del algoritmo FLF muestra que su convergencia es alcanzada en pocos pasos debido a que se fundamenta en funciones diferenciables y métodos de optimización que tienen convergencia comprobada.

- Debido a que el método estudiado en esta investigación se fundamentan en estructuras algebraicas y técnicas de optimización se hace sencillo su implementación numérica.

Referencias Bibliográfícas

[1] Bühlmann H., Boosting algorithms: regularization, prediction and model fitting. Statistical science, Vol 22, No. 4 (2007), 477- 505. En: http://www.cs.princeton.edu/~schapire/boost.html. [Consultado 6 de Septiembre 2009].

[2] Duffy H., Boosting Methods for Regression, The Netherlands, Kluwer Academic Publishers, Vol. 27, (2002), 153-200.

[3] Kramer., An Introduction to Boosting in the Regresson Framework, Technical University Berlin, Institute de Quantitative Methods, (2007). En: https://scholar.google.co.ve/scholar?q=an+introduction+to+boosting+in+the+regression+framework&hl=es&as_sdt=0&as_vis=1&oi=scholart&sa=X&ved=0CBkQgQMwAGoVChMInsauttSSyQIViGImCh2UOgQ2 [Consultado 20 de Julio 2007].

[4] Momma M., y Bennett K., Constructing Orthogonal Latent Features for Arbitrary Loss, Studies in Fuzziness and Soft Computing, Vol 207, (2006), 1-28. En: http://basilo.kaist.ac.kr/papers/rensselaer/bennet/mb7.pdf [Consultado 20 Noviembre 2010].

[5] Freund S., A Short Introduction to Boosting, Journal of Japanese Society for Artificial Inteligence, Vol 5, No.14, (1999), 771-780. En: http://www.cs.princeton.edu/~schapire/boost.html [Consultado 6 de Septiembre 2009].

[6] Mason Ll., Baxter J., Boosting Algorithms as Gradient Descent in Function Space, MIT Press Cambridge (1999). En: http://citeseer.ist.psu.edu/124873.html [Consultado 13 de Noviembre 2007].

[7] Thomas G., y Finney R., Cálculo varias variables (9^{na} edición), Addison Wesley Longman, México, (1999).

[8] Luenberger D., Programación Lineal y no Lineal. Addison-Wesley Iberoamericana, (1989).