Algoritmo Robusto de Aprendizaje para el Modelo Mezcla de Expertos

Romina D. Torres[1,2]
Director: Dr. Héctor Allende; Correferente: Dr. Horst von Brand ; Externo: Dr. Max Chacón.
[1]Universidad Técnica Federico Santa María Dept. de Informática,
Av. España 1680, Casilla 110-V, Valparaíso-Chile;

[2] Motorola Valparaíso; Global Software Group
romina.torres@motorola.com

Abstract

El Modelo de Mezcla de Expertos (ME) es un tipo de Redes Neuronales Artificiales Modulares (MANN) especialmente adecuadas cuando el espacio de entrada se encuentra estratificado. La arquitectura está compuesta por diferentes módulos: redes expertas que compiten por aprender diferentes aspectos de un problema y una red de agregación que arbitra la competencia y aprende a asignar diferentes regiones del espacio de datos a diferentes expertos locales cuya topología parece ser la más apropiada. La regla de aprendizaje combina aspectos competitivos y asociativos y está diseñada para favorecer la competencia entre expertos locales, que permiten dividir el espacio 'automáticamente' en subregiones manejadas en lo posible por un único experto local.

El aprendizaje del modelo ME puede ser tratado como un problema de estimación de parámetros que maximizan la verosimilitud, donde el algoritmo de Máxima Expectación desacopla el proceso de estimación en una manera que calza con la estructura modular de la arquitectura ME.

Sin embargo, cuando los datos están expuestos a datos atípicos, el modelo es afectado debido a que el algoritmo es sensible a estas desviaciones obteniendo un bajo rendimiento. En esta tesis se propone robustificar el algoritmo EM para el modelo ME, obteniendo un algoritmo elegante, eficiente, de rápida convergencia debido a que aprovecha la modularidad del modelo (baja interferencia destructiva), y a la vez es insensible a los datos atípicos (acotando el impacto de ellos en la obtención de los estimadores pero sin eliminarlos). Para ésto se utiliza una generalización del estimador máximo verosímil conocido como M-estimadores.

En la fase de prueba se seleccionan problemas reales y con presencia de datos atípicos pertenecientes a la serie de problemas estándares DELVE y PROBEN1, para mostrar que el algoritmo Robusto de Máxima Expectación para Mezcla de Expertos (REM-ME) muestra mejoras significativas con respecto a los métodos clásicos.

Palabras Claves: Redes Neuronales Artificiales Modulares, Modelos de Mezcla, Modelo Mezcla de Expertos, M-estimadores, Algoritmo de Máxima Expectación.

1 Introducción

El cerebro es un sistema altamente complejo debido a que está compuesto de un gran número de componentes que interactúan entre sí. Estudios en Biología y Psicología han descubierto la arquitectura modular del cerebro, pero no han llegado a un acuerdo en el número de módulos, la naturaleza de su interacción o la manera en que ellos se desarrollan. Estos módulos se especializan en ciertas funciones de acuerdo a sus propiedades estructurales, presentando alta cohesión. Una pregunta válida es si la definición de cuál módulo realiza qué función es acorde al material genético o en base a la experiencia.

Un creciente número de investigadores cree que lo que realmente determina las propiedades funcionales de las distintas regiones del cerebro, es un proceso dependiente de la experiencia. Esta teoría combina dos grandes nociones:

- Debido a las diferentes propiedades estructurales de las regiones del cerebro, existe una correspondencia entre sus estructuras y las funciones que son capaces de realizar. Aunque diferentes regiones puedan realizar una misma tarea o puedan adaptarse para realizarla, algunas serán más eficientes que otras.

- Diferentes regiones del cerebro compiten por la habilidad de realizar un conjunto de tareas. Basados en la primera noción se puede establecer que la competencia está sesgada, ya que cada región tiende a ganar la competencia en aquellas funciones en las cuales su estructura es la más adecuada.

Jacobs et al. en [JJNH91a], proponen una arquitectura de redes neuronales, conocida como Modelo Mezcla de Expertos (ME) que implementan las nociones anteriores utilizando modelos conexionistas. Esta arquitectura es ilustrada en la Figura 1, y consiste en dos tipos de redes neuronales: redes expertas y una red de agregación (conocida como *gating* en la literatura inglesa). Las redes expertas compiten por aprender de los patrones de entrenamiento, mientras la red de agregación media esta competencia.

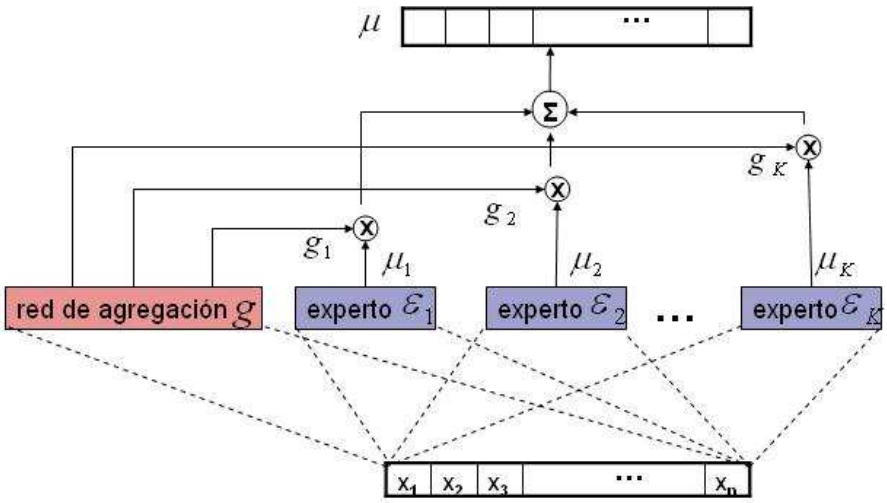

Figure 1: *Modelo Mezcla de Expertos(ME)*: consiste en un conjunto de redes expertas y una red de agregación. Los expertos compiten por aprender diferentes aspectos del problema y la red de agregación es la mediadora de esta competencia.

El modelo ME es un caso particular de una arquitectura compuesta por un conjunto de redes que tienen la capacidad de inhibirse entre sí, suprimiendo la salida de otras redes. La fuerza inhibitoria de la arquitectura depende del contexto del problema, es decir del valor actual de los patrones. Aunque en el cerebro, no existe evidencia física de una red de agregación, al finalizar el entrenamiento, aquellas redes que ganan la competencia, suprimen fuertemente la salida de las redes perdedoras (con pobre rendimiento). Existen diversos autores que a partir de los resultados obtenidos por este tipo de arquitectura sugieren que en el cerebro los módulos neuronales interactúan entre sí inhibitoriamente.

El problema de aprendizaje de la arquitectura ME puede ser tratado como un problema de estimación de parámetros. Un enfoque común es aplicar el método de gradiente descendente para obtener los estimadores de máxima verosimilitud (ML). En [JJ91], [JJB91] y [JJ92], resultados empíricos revelan que, aunque el enfoque del gradiente es exitoso en encontrar valores razonables de los parámetros en problemas particulares, la razón de convergencia no fue significante mayor que el obtenido al utilizar el método de gradiente en arquitecturas feedforward.

Un enfoque alternativo es presentado en [JJ94], donde se introduce el algoritmo de Máxima Expectación (EM) para la arquitectura Mezcla de Expertos que mostró tomar ventaja de la modularidad de la arquitectura. El algoritmo EM es un algoritmo de propósito general para obtener estimadores de máxima verosimilitud en una amplia variedad de situaciones mejor descritas como problemas de *datos incompletos*. Aunque no existan datos perdidos u otra forma de incompletitud en los datos, casi siempre es posible reformular un problema dentro de un marco EM. La formulación actual del algoritmo EM [DLR77], considera una variedad de ejemplos y condiciones especiales para su convergencia.

En esta tesis, se propone utilizar el algoritmo EM para estimar los parámetros del modelo ME de manera de aprovechar la ventaja de la estructura modular de la arquitectura. Se mostrará que cuando se presentan datos atípicos en la muestra (valores aberrantes que se alejan fuertemente del modelo subyacente), denominados outliers en la literatura inglesa, el supuesto distribucional del comportamiento entre los datos de entrada y de salida ya no es completamente válido. Tales datos atípicos son usualmente eliminados utilizando un filtro basado en un umbral para la chi-cuadrada antes de aplicar el algoritmo EM. Sin embargo, encontrar el umbral adecuado es difícil y usualmente será arbitrario. Consecuentemente con ésto, información 'necesaria' podría ser rechazada como datos atípicos. Por ejemplo, en problemas de clasificación,

una clase con pocas muestras sería difícil de identificar y la teoría de filtros podría asumir que estas muestras constituyen outliers estadísticos, eliminando información que podría ser la semilla para una nueva clase.

Como una alternativa a las técnicas habituales de preprocesamiento de los datos, estudios de robustez, en especial en la etapa de entrenamiento de redes feedforward han sido propuestas en varios trabajos (ver [CM94], [AMS01], [AMS02]) donde se ha mostrado la efectividad de utilizar algoritmos que acotan la influencia de los datos atípicos sin llegar a eliminarlos.

Debido a que la arquitectura ME es sensible a la presencia de datos atípicos, el objetivo de esta tesis es construir un algoritmo EM Robusto para la estimación de los parámetros mediante la maximización de la verosimilitud del modelo neuronal ME utilizando M-estimadores, de manera tal que la influencia de los datos atípicos es acotada.

2 Alcance y Contribución de esta Tesis

El objetivo de esta tesis es desarrollar un algoritmo de aprendizaje robusto para el modelo Mezcla de Expertos, que le permita ser insensible a la presencia de datos atípicos y que tome ventaja de la modularidad del modelo (en cuanto a rapidez de convergencia).

En base a este objetivo general, en esta tesis se realizan las siguientes hipótesis:

- Elección del Modelo ME: cuando se tienen tareas de distinta naturaleza que se desean modelar por una única red se produce un problema en el aprendizaje conocido como 'interferencia destructiva'. La elección más adecuada en estos casos es seleccionar el modelo ME donde distintas redes modelan distintas tareas, y por tanto no interfieren entre sí durante el aprendizaje.

- Se obtienen mejoras significativas en el rendimiento obtenido por el modelo ME en el conjunto de prueba al obtener los parámetros mediante un algoritmo insensible a los datos atípicos.

- Se reduce la complejidad del modelo. Cuando el algoritmo no considera un tratamiento para los datos atípicos, es altamente probable que el número de expertos del modelo ME aumente debido a que el modelo trata de modelar los datos atípicos con expertos adicionales. La introducción de robustez permite modelar la real tendencia de los datos independiente del ruido que presenten.

- El modelo ME converge al menos un orden de magnitud más rápido si se utiliza durante el aprendizaje el algoritmo EM en vez de un algoritmo basado en el gradiente, debido a que tomará ventaja de la modularidad del modelo.

- Cuando los conjuntos de datos presentan datos atípicos, el modelo ME con aprendizaje robusto presentará mejoras significativas en el rendimiento.

Esta tesis tiene una contribución especial al mundo científico debido a que el modelo Mezcla de Expertos desde su aparición a comienzos de los años 90, [JJNH91b], ha sido exitosamente utilizado en problemas que involucran diferentes fuentes de información. En este tipo de problemas el modelo ME ha mostrado ser superior a una única red neuronal en cuanto a su rendimiento, su capacidad de generalización y su razón de convergencia. Lamentablemente, cuando los datos son ruidosos, presentando datos atípicos (usualmente los conjuntos de datos reales presentan estas características), el modelo ME disminuye su capacidad de generalización, y disminuye su rapidez de convergencia, siendo necesario usualmente aumentar el número de expertos de la mezcla, y por lo tanto aumenta el número de parámetros a estimar, ver [TSAM02], [TSAM03] y [ATSM03]. Esta tesis presenta una posible solución a este problema, robustificando el algoritmo de aprendizaje, adquiriendo por tanto insensibilidad frente a los datos estadísticos atípicos, sin rechazarlos (como lo haría una teoría de filtros) pero acotando su influencia, permitiendo que el modelo extraiga la información relevante que ellos pudiesen enmascarar.

Es importante mencionar, que durante el desarrollo de esta Tesis se generaron las siguientes publicaciones indexadas en Congresos internacionales: [TSAM03] y [ATSM03]. Un tercer artículo ha sido aceptado para su pronta publicación [AMST04].

3 Mezcla de Expertos

Cuando el algoritmo de gradiente descendente es utilizado para entrenar una red multicapa para realizar diferentes subtareas, generalmente aparecerá un efecto de interferencia que generará un proceso de aprendizaje lento y una pobre capacidad de generalización. Si se conoce a priori que el conjunto de casos de entrenamiento, puede ser naturalmente dividido en

subconjuntos que corresponden a cada una de las distintas subtareas, la interferencia puede ser reducida utilizando un sistema compuesto de distintas redes expertas más una red de agregación, que decide cuáles de los expertos deberían ser usados para cada observación.

Cuando se tiene un espacio de entrada estratificado en diferentes subregiones, mediante el aprendizaje se debe inferir que existirán distintos mapas en cada subregión. Aunque una única red homogénea podría ser aplicada a este problema, se espera que las tareas sean más fáciles de realizar si se asignasen diferentes redes expertas a cada una de las subregiones, y el criterio de asignación de un experto a una subregión fuese realizado por una red extra, denominada red de agregación, que recibe como información durante el aprendizaje, el patrón de entrada y el rendimiento relativo de los expertos para ese patrón.

La arquitectura (ME), [JJ99], está compuesta de K redes expertas, cada una de las cuales resuelve un problema de aproximación de función sobre una región local del espacio de entrada. El conjunto de datos es denominado Υ, que está formado por duplas compuestas de vectores de entrada $\underline{x} \in \mathbb{R}^n$ y vectores de salida $\underline{y} \in \mathbb{R}^m$. En esta sección, utilizaremos la versión extendida del conjunto de datos, es decir su descomposición explícita en vectores de entrada y salida.

A cada una de las redes expertas del modelo ME, denominado ε_j, se le asocia un modelo probabilístico como sigue:

$$P(\underline{y}|\underline{x}, \underline{\theta}_j, \Sigma_j), \qquad j = 1, \ldots, K, \tag{1}$$

donde $\underline{\theta}_j$ es el vector de los parámetros de la red experta j-ésima y Σ_j es la matriz de covarianza de los residuos generados entre la diferencia del dato verdadero y la salida del experto. Por motivos de simplicidad se asume que estas densidades de probabilidad pertenecen a la familia exponencial, restringiendo el análisis del presente trabajo a modelos Gaussianos.

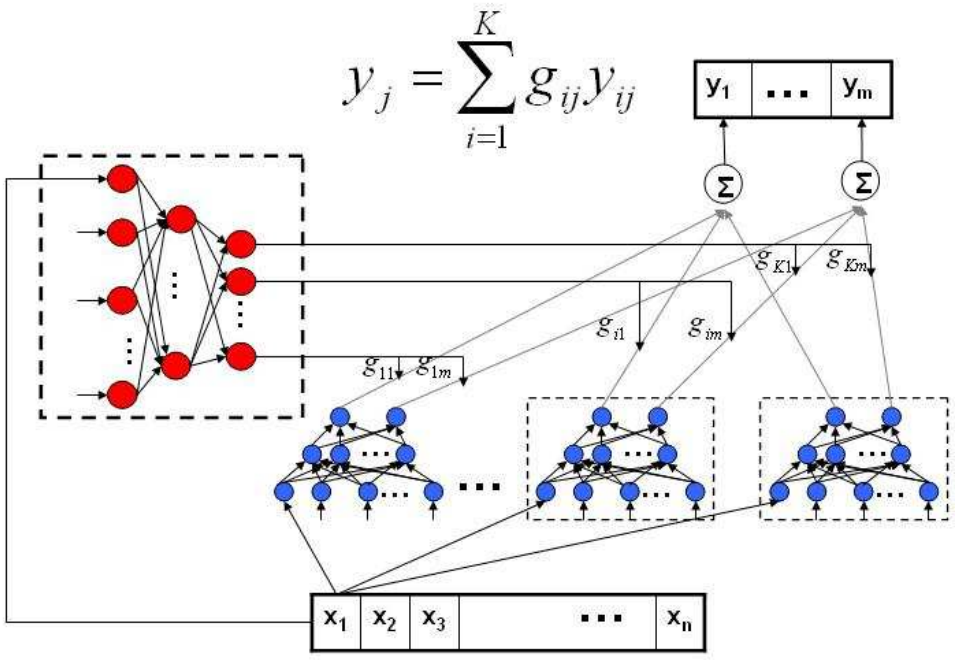

Figure 2: *Arquitectura Mezcla de Expertos(ME)*: La arquitectura ME consiste en un conjunto de redes expertas y una red de agregación. Los expertos compiten por aprender diferentes aspectos del problema y la red de agregación es la mediadora de la competencia.

Considerando la figura 2, cada experto ε_j (red j-ésima) genera como salida un vector de parámetros μ_j, con

$$\underline{\mu}_j = E_P[\underline{y}|\underline{x}, \underline{\theta}_j] = \underline{f}_j(\underline{x}, \underline{\theta}_j), \quad j = 1, \ldots, K, \tag{2}$$

donde $\underline{\mu}_j$ es el parámetro de localización del modelo $P(\underline{y}|\underline{x}, \underline{\theta}_j, \Sigma_j)$(en este caso es simplemente la media). En esta tesis se asumirá que los $\underline{f}_j$ son lineales en los parámetros y que la componente de no linealidad del modelo es entregada por la red de agregación, debido a que cada salida, g_i, probabilidad condicional de los datos y del rendimiento de cada experto.

Cada red experta tiene asociada una matriz de covarianza, Σ_j, no singular, por lo tanto el modelo probabilístico para el experto ε_j, puede ser escrito como $N(f_j(\underline{x},\underline{\theta}_j),\Sigma_j)$, o explícitamente

$$P(\underline{y}|\underline{x},\underline{\theta}_j,\Sigma_j) = \frac{1}{(2\pi)^{\frac{m}{2}}\|\Sigma_j\|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}[\underline{y}-\underline{f}_j(\underline{x},\underline{\theta}_j)]^T\Sigma_j^{-1}[\underline{y}-\underline{f}_j(\underline{x},\underline{\theta}_j)]\right\} \tag{3}$$

Diferentes redes expertas son apropiadas en diferentes regiones del espacio de entrada, por lo tanto la arquitectura determina estocásticamente para una entrada $\underline{x}$, cuál experto o mezcla de expertos es más apropiada para producir la salida deseada. Para este fin, la arquitectura ME también utiliza una red auxiliar conocida como la *red de agregación* (gating) que particiona el espacio de entrada en diferentes regiones correspondientes a diferentes redes o mezcla de redes expertas. Esto se logra asignando un vector de probabilidad $[g_1,g_2,\ldots,g_K]^T$ a cada punto del espacio de entrada. En particular, la red de agregación implementa una función parametrizada $\xi(\underline{x},\underline{\theta}_0)$ y una función normalizadora $\underline{g}(\xi)$. La función ξ es un mapa desde $\mathbb{R}^n$ a $\mathbb{R}^K$, para cada elemento del vector de parámetros $\underline{\theta}_0$, y la función $\underline{g}$ es un mapa de $\mathbb{R}^K$ a $\mathbb{R}^K$. Una forma particular de $\underline{g}$ es la función softmax,

$$g_j = g_j(\underline{x},\underline{\theta}_0) = \frac{e^{\xi_j(\underline{x},\underline{\theta}_0)}}{\sum_{i=1}^K e^{\xi_i(\underline{x},\underline{\theta}_0)}}, \qquad j=1,\ldots,K. \tag{4}$$

Notar que esta definición implica que g_j es no-negativa, $g_j \geq 0$, y $\sum_{i=1}^K g_i = 1$.

Una interpretación probabilística es ver g_j como una superficie discriminante en un problema de clasificación y por lo tanto a la red de agregación como un sistema *clasificador* que mapea una entrada $\underline{x}$ a la probabilidad de que un experto o una mezcla de expertos sea capaz de generar la salida deseada (basado sólo en el conocimiento de $\underline{x}$).

Es importante hacer notar, que aunque se seleccione una estructura lineal para las redes expertas, el espacio es dividido suavemente gracias a la forma de las salidas de la red de agregación, siendo éstas las que entregan la componente de no linealidad al modelo, permitiendo utilizar arquitecturas más simples para las redes expertas con salidas lineales en vez de sigmoidales sin degradar el rendimiento obtenido por el modelo.

Se asume que los datos de entrenamiento son generados de acuerdo al siguiente modelo de probabilidad. Asumiendo que para un $\underline{x}$ dado, un experto ε_j es seleccionado con probabilidad $P(\varepsilon_j|\underline{x},\underline{\theta}_0) = g_j(\underline{x},\underline{\theta}_0)$. Una salida $\underline{y}$ es entonces escogida con probabilidad $P(\underline{y}|\underline{x},\underline{\theta}_j,\Sigma_j)$. Por lo tanto, la probabilidad total de generar $\underline{y}$ dada la entrada $\underline{x}$ es dado por la siguiente densidad de mezcla,

$$P(\underline{y}|\underline{x}) = \sum_{j=1}^K P(\varepsilon_j|\underline{x},\underline{\theta}_0)P(\underline{y}|\underline{x},\underline{\theta}_j,\Sigma_j) = \sum_{j=1}^K g_j(\underline{x},\underline{\theta}_0)P(\underline{y}|\underline{x},\underline{\theta}_j,\Sigma_j) \tag{5}$$

Las *redes expertas* modelan los distintos procesos que generan los datos, y la *red de agregación* modela la decisión de utilizar uno de esos diferentes procesos. Cuando el aprendizaje es extremadamente competitivo, la salida $\underline{y}$ es generada por un único experto.

Se asume que el conjunto de entrenamiento $\Upsilon = \{(\underline{x}^{(t)},\underline{y}^{(t)})\}_{t=1}^N$ es generado de la siguiente manera: dada la elección de la entrada $\underline{x}$, un experto ε_j es escogido con probabilidad $P(\varepsilon_i|\underline{x},\underline{\theta}_0^*)$ (donde el exponente '*' distingue los valores de los parámetros reales del modelo). Dada la elección del experto ε_j y dada la entrada $\underline{x}$, se asume que la salida deseada $\underline{y}$ es generada con probabilidad $P(\underline{y}|\underline{x},\underline{\theta}_j^*,\Sigma_j)$.

4 Algoritmo de Aprendizaje basado en el gradiente

Para desarrollar un algoritmo con el fin de estimar los parámetros de la arquitectura de una mezcla de expertos, se puede utilizar el principio de máxima verosimilitud (ML). Esto significa que se deben escoger los parámetros para los cuáles la probabilidad del conjunto de entrenamiento dado los parámetros (función conocida como verosimilitud) es máxima.

Dado un conjunto de N muestras independientes e idénticamente distribuidas, $\Upsilon = \{(\underline{x}^{(t)},\underline{y}^{(t)})\}_{t=1}^N$ la verosimilitud correspondiente a una mezcla de K-componentes es

$$L(\underline{\Theta},\Upsilon) = P(\{\underline{y}^{(t)}\}_1^N|\{\underline{x}^{(t)}\}_1^N) = \prod_{t=1}^N P(\underline{y}^{(t)}|\underline{x}^{(t)}) = \prod_{t=1}^N\sum_{j=1}^K g_j(\underline{x}^{(t)},\underline{\theta}_0)P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j) \tag{6}$$

El problema de aprendizaje de la arquitectura ME es tratado como un problema de encontrar los estimadores de los parámetros $\underline{\theta}_0$, $\underline{\theta}_j$ y Σ_j que maximicen la función de verosimilitud $L(\underline{\Theta}, \Upsilon)$. Como es común en Estadística, es más conveniente trabajar con el logaritmo de la verosimilitud que con ella misma. Tomando el logaritmo del producto de N densidades lleva a la siguiente ecuación,

$$l(\underline{\Theta}, \Upsilon) = \ln \prod_{t=1}^{N} \sum_{j=1}^{K} g_j(\underline{x}^{(t)}, \underline{\theta}_0) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j) = \sum_{t=1}^{N} \ln \sum_{j=1}^{K} g_j(\underline{x}^{(t)}, \underline{\theta}_0) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j) \tag{7}$$

donde $\underline{\Theta} = [\underline{\theta}_0, \underline{\theta}_1, \ldots, \underline{\theta}_K, \Sigma_1, \Sigma_2, \ldots, \Sigma_K]^T$.

Para obtener los parámetros de las redes expertas y la red de agregación, el logaritmo de la función de la verosimilitud $l(\underline{\Theta}, \Upsilon)$, debe ser maximizada a través de un proceso de aprendizaje del modelo. En este trabajo el gradiente ascendente es aplicado a la ecuación (7).

Como se nombró anteriormente, la red de agregación es considerada lineal, $\underline{\xi} = \theta_0^T \underline{x}$, sus salidas son obtenidas luego de aplicar la función softmax. La distribución condicional es $P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{m}{2}}} \exp\left\{ -\frac{(\underline{y}-\underline{\mu}_i)^T (\underline{y}-\underline{\mu}_i)}{2} \right\}$. Calculando el gradiente de $l = l(\underline{\Theta}, \Upsilon)$ con respecto a $\underline{\xi}_j$ y $\underline{\mu}_j$ se obtiene que

$$\frac{\partial l}{\partial \underline{\mu}_j} = \sum_{t=1}^{N} h_j^{(t)} \frac{\partial}{\partial \underline{\mu}_j} \ln P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j) \tag{8}$$

y

$$\frac{\partial l}{\partial \underline{\xi}_j} = \sum_t (h_j^{(t)} - g_j^{(t)}) \tag{9}$$

donde $h_j^{(t)}$ es definido como $P(\varepsilon_j | \underline{x}^{(t)}, \underline{y}^{(t)})$. Utilizando la regla de Bayes:

$$P(\varepsilon_j | \underline{x}^{(t)}, \underline{y}^{(t)}) = \frac{P(\varepsilon_j | \underline{x}^{(t)}) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \varepsilon_j)}{\sum_i P(\varepsilon_i | \underline{x}^{(t)}) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \varepsilon_i)} \tag{10}$$

Esto sugiere que, $h_j^{(t)}$ está definido como la *probabilidad a posteriori* del experto j-ésimo, condicionado a la entrada $\underline{x}^{(t)}$ y a la salida $\underline{y}^{(t)}$. Similarmente, la probabilidad $g_j^{(t)}$ puede ser interpretada como la *probabilidad a priori* $P(\varepsilon_j | \underline{x}^{(t)})$ del experto j-ésimo dado solamente la entrada $\underline{x}^{(t)}$. Dada estas definiciones, la ecuación (9) tiene una interpretación natural de mover las probabilidades a priori hacia las probabilidades a posteriori.

Un caso especial es una arquitectura en que las redes expertas y la red de agregación son lineales y con densidad de probabilidad Gaussiana con $\Sigma = I$. Por lo tanto la ecuación (8) y (9) nos permiten escribir explícitamentelas la regla de actualización de los parámetros para la red de agregación (11):

$$\Delta \underline{\theta}_0 = \underline{\theta}_0^{k+1} - \underline{\theta}_0^k = \alpha \sum_t (h_j^{(t)} - g_j^{(t)}) \underline{x}^{(t)^T} \tag{11}$$

Para las redes expertas (12) se obtiene que:

$$\Delta \underline{\theta}_j = \alpha \sum_{t=1}^{N} h_j^{(t)} (\underline{y}^{(t)} - \underline{\mu}_j) \underline{x}^{(t)^T} \tag{12}$$

donde α es la razón de aprendizaje. La expresión de la probabilidad a posteriori en el caso Gaussiano viene dada por:

$$h_j^{(t)} = \frac{g_j^{(t)} \exp\{-\frac{1}{2}(\underline{y}^{(t)} - \underline{\mu}_j^{(t)})^T (\underline{y}^{(t)} - \underline{\mu}_j^{(t)})\}}{\sum_i g_i^{(t)} \exp\{-\frac{1}{2}(\underline{y}^{(t)} - \underline{\mu}_i^{(t)})^T (\underline{y}^{(t)} - \underline{\mu}_i^{(t)})\}} \tag{13}$$

Esto es una medida de distancia normalizada que refleja las magnitudes relativas de los residuales $\underline{y}^{(t)} - \underline{\mu}_i^{(t)}$. Si los residuales para el experto ε_j son pequeños relativo a los otros, entonces $h_j^{(t)}$ es grande. Los $h_j^{(t)}$ son positivos y su suma por cada $\underline{x}^{(t)}$ es uno; ésto implica que la responsabilidad es distribuida entre los expertos en una manera competitiva.

5 Algoritmo de Máxima Expectación para el Modelo Mezcla de Expertos

Jacobs et al. en [JJ91] basados en pruebas empíricas realizadas en su investigación, revelaron que aunque el enfoque del gradiente encontraba exitosamente los valores de los parámetros para problemas particulares, la razón de convergencia no era mejor que la obtenida utilizando el enfoque del gradiente en redes multicapas. La razón es que el gradiente no parecía tomar ventaja de la modularidad de la arquitectura.

Un método alternativo al enfoque del gradiente fue propuesto por Jordan y Jacobs en [JJ94], quienes introdujeron el algoritmo de Máxima Expectación (EM) ([DLR77]) para la arquitectura Mezcla de Expertos. Para esta arquitectura el algoritmo EM desacopla el proceso de estimación en una manera que ajusta bien con la estructura modular de la arquitectura.

EM está basado en la idea de resolver una sucesión de problemas simplificados que son obtenidos por aumentar las variables observadas originalmente con un conjunto adicional de variables denominadas *escondidas*. .

Un conjunto de datos observados Υ, es aumentado por un conjunto Υ_{perd}, denominado conjunto de variables *perdidas* o *escondidas*. Considérese el problema de máxima verosimilitud para un conjunto de *datos completos*, $\mathbb{Z} = \{\Upsilon, \Upsilon_{perd}\}$. Las variables perdidas se escogen de tal manera que el logaritmo de la máxima verosimilitud de los *datos completos*, dado por $l_c(\underline{\Theta}, \mathbb{Z}) = \log P(\Upsilon, \Upsilon_{perd}|\underline{\Theta})$, es fácil de maximizar con respecto a $\underline{\Theta}$. El modelo de probabilidad $P(\Upsilon, \Upsilon_{perd}|\underline{\Theta})$ debe ser escogido de tal manera que su distribución marginal a través de Υ, denominado en este contexto como la verosimilitud de los *datos incompletos* es la verosimilitud original

$$P(\Upsilon|\underline{\Theta}) = \int_{\Upsilon_{perd}} P(\Upsilon, \Upsilon_{perd}|\underline{\Theta}) d\Upsilon_{perd} \tag{14}$$

Debido a que la verosimilitud es una función de las variables aleatorias perdidas, no se puede trabajar directamente con la verosimilitud de los *datos completos*. Es necesario promediar Υ_{perd}, es decir maximizar el *valor esperado* del logaritmo de la verosimilitud de los *datos completos* $E_{\Upsilon_{perd}}[\log P(\Upsilon, \Upsilon_{perd}|\underline{\Theta})|\Upsilon, \Theta]$.

El algoritmo EM es un algoritmo iterativo que consta de dos pasos:

- El paso de expectación (E) que calcula el valor esperado condicional del logaritmo de la verosimilitud

$$Q(\underline{\Theta}|\underline{\Theta}^{(k)}) \quad = \int_{\Upsilon_{perd}} P(\Upsilon_{perd}|\Upsilon, \underline{\Theta}^{(k)}) \log P(\mathbb{Z}|\underline{\Theta}) d\Upsilon_{perd} \tag{15}$$

  donde $\underline{\Theta}^{(k)}$ es el valor esperado del vector de parámetros en la iteración k.

- El paso de maximización (M) que calcula

$$\underline{\Theta}^{(k+1)} = \arg\max_{\underline{\Theta}} Q(\underline{\Theta}|\underline{\Theta}^{(k)}) \tag{16}$$

El paso M escoge un valor para el vector de parámetros que aumenta la función Q, es decir el valor esperado del logaritmo de la verosimilitud de los *datos completos*. Dempster et al. [DLR77] muestran que una iteración del algoritmo EM también aumenta el logaritmo de la verosimilitud original l. Es decir,

$$l(\underline{\Theta}^{(k+1)}; \Upsilon) \geq l(\underline{\Theta}^{(k)}; \Upsilon) \tag{17}$$

Eso quiere decir que la función de verosimilitud l crece monóticamente de acuerdo a la secuencias de parámetros estimados generados por el algoritmo EM. La solución del paso M puede ser obtenido analíticamente y en aquellos casos en que no sea posible, será necesario realizar un procedimiento iterativo interno para optimizar Q.

Aunque en la práctica, frecuentemente la solución del paso M existe en una forma cerrada, existen veces en que no es factible encontrar el valor de $\underline{\Theta}$ que maximiza globalmente la función $Q(\underline{\Theta}, \underline{\Theta}^{(k)})$. Para tales situaciones, Dempster et al. [DLR77], definieron un algoritmo EM generalizado (algoritmo GEM) para el cuál el paso M requiere escoger $\underline{\Theta}^{(k+1)}$ tal que:

$$Q(\underline{\Theta}^{(k+1)}, \underline{\Theta}^{(k)}) \geq Q(\underline{\Theta}^{(k)}, \underline{\Theta}^{(k)}) \tag{18}$$

Esto significa escoger un $\underline{\Theta}^{(k+1)}$ que aumenta la función Q sobre su valor en $\underline{\Theta} = \underline{\Theta}^{(k)}$, en vez de maximizarlo sobre todo $\underline{\Theta} \in \Omega$. Como se puede ver en [MP01] (sección 3.3), la condición anterior en $\underline{\Theta}^{(k+1)}$ es suficiente para asegurar que

$$L(\underline{\Theta}^{(k+1)}) \geq L(\underline{\Theta}^{(k)}) \tag{19}$$

En este caso, la verosimilitud $L(\underline{\Theta},\Upsilon)$ no decrece después de una iteración GEM, y así una secuencia GEM de valores de verosimilitud debe converger si existe un límite superior.

Para la arquitectura ME se escogen los datos perdidos como variables aleatorias indicadoras $\Upsilon_{perd} = \{\gamma_{perd_j}^{(t)}, j = 1,\ldots,K,\ t = 1,\ldots,N\}$ con

$$\gamma_{perd_j}^{(t)} = \begin{cases} 1, & \text{si } \underline{y}^{(t)} \text{ es generado por el modelo } j-esimo. \\ 0, & e.t.o.c. \end{cases} \tag{20}$$

y

$$\sum_{j=1}^{K} \gamma_{perd_j}^{(t)} = 1,\ para\ cada\ t \tag{21}$$

Asumiendo que la distribución de los *datos completos* es dada por

$$P(\mathbb{Z}|\underline{\Theta}) = \prod_{t=1}^{N}\prod_{j=1}^{K}[g_j(\underline{x}^{(t)},\underline{\theta}_0)P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)]^{\gamma_{perd_j}^{(t)}} \tag{22}$$

ecuación que satisface (14).

De (15) se obtiene que

$$\begin{array}{ll} Q(\underline{\Theta}|\underline{\Theta}^{(k)}) & = E_{\Upsilon_{perd}}\{\ln P(\mathbb{Z}|\underline{\Theta})|\Upsilon,\underline{\Theta}^{(k)}\} \\ & = \sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(t)\ln g_j(\underline{x}^{(t)},\underline{\theta}_0) + \sum_{t=1}^{N} h_1^{(k)}(t)\ln P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_1,\Sigma_1) \\ & + \ldots + \sum_{t=1}^{N} h_K^{(k)}(t)\ln P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_K,\Sigma_K) \end{array} \tag{23}$$

donde

$$\begin{array}{ll} h_j^{(k)}(t) & = E[\gamma_{perd_j}^{(t)}|\Upsilon,\underline{\Theta}^{(k)}] = P(j|\underline{x}^{(t)},\underline{y}^{(t)}) \\ & = \frac{g_j(\underline{x}^{(t)},\underline{\theta}_0^{(k)})P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_j^{(k)},\Sigma_j^{(k)})}{\sum_{i=1}^{K} g_i(\underline{x}^{(t)},\underline{\theta}_0^{(k)})P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_i^{(k)},\Sigma_i^{(k)})}, \end{array} \tag{24}$$

donde $P(j|\underline{x}^{(t)},\underline{y}^{(t)})$ denota la probabilidad de que el par $\{\underline{x}^{(t)},\underline{y}^{(t)}\}$ haya sido generado por el modelo de probabilidad $j-esimo$. Notar que siempte $h_j^{(k)}(t) > 0$.

La implementación del paso M basándose en las ecuaciones (5), (4) y (23), es obtenida

$$\frac{\partial Q}{\partial \underline{\theta}_0} = \sum_{t=1}^{N}\sum_{j=1}^{K}[h_j^{(k)}(t) - g_j(\underline{x}^{(t)},\underline{\theta}_0)]\frac{\partial \xi_j}{\partial \underline{\theta}_0} \tag{25}$$

$$\frac{\partial Q}{\partial \underline{\theta}_j} = \sum_{t=1}^{N} h_j^{(k)}(t)\frac{\partial \underline{f}_j^T(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)}{\partial \underline{\theta}_j}\Sigma_j^{-1}[\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)], \tag{26}$$

y

$$\frac{\partial Q}{\partial \Sigma_j} = -\tfrac{1}{2}\sum_{t=1}^{N} h_j^{(k)}(t)\Sigma_j^{-1}\{\Sigma_j - [\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)][\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)]^T\}\Sigma_j^{-1} \tag{27}$$

Si $\frac{\partial Q}{\partial \Sigma_j}\big|_{\Sigma_j=\Sigma_j^{(k+1)}} = 0$, se obtiene la actualización para las matrices de covarianza

$$\Sigma_j^{(k+1)} = \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(t)}\sum_{t=1}^{N} h_j^{(k)}(t)[\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)][\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)]^T \tag{28}$$

Asumiendo que el conjunto de entrenamiento Υ, es generado por un modelo de mezcla de expertos, cuando el número de muestra es suficientemente grande (relativo a la dimensión de $\underline{y}$), el espacio dado por los N vectores $[\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)]$ será de dimensión completa con probabilidad 1. Debido a que $h_j^{(k)}(t) > 0$, cuando el número de muestras N, es suficientemente grande las matrices $\Sigma_j^{(k+1)}$ son definidas positivas con probabilidad 1.

Haciendo $\frac{\partial Q}{\partial \underline{\theta}_j}\big|_{\underline{\theta}_j=\underline{\theta}_j^{(k+1)}} = 0$, se obtiene que

$$\sum_{t=1}^{N} h_j^{(k)}(t)\frac{\partial \underline{f}_j^T(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)}{\partial \underline{\theta}_j}(\Sigma_j^{(k)})^{-1}[\underline{y}^{(t)} - \underline{f}_j(\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)] = 0, \tag{29}$$

lo que puede ser resuelto explícitamente dado el supuesto de que las redes expertas son lineales

$$\underline{\theta}_j^{(k+1)} = (\underline{R}_j^{(k)})^{-1}\underline{c}_j^{(k)} \tag{30}$$

donde

$$\underline{c}_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)}(t) X_t (\Sigma_j^{(k)})^{-1}\underline{y}^{(t)}, \tag{31}$$

$$\underline{R}_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)}(t) X_t (\Sigma_j^{(k)})^{-1} X_t^T, \tag{32}$$

Notar que $R_j^{(k)}$ es invertible con probabilidad uno cuando la muestra de tamaño N es suficientemente grande.

Finalmente, consideremos la actualización para θ_0. Jordan y Jacobs observaron que la red de agregación es una forma específica del modelo generalizado, en particular un modelo multinomial *logit*. Estos modelos pueden ser ajustados eficientemente con una variante del método de Newton conocido como *IRLS*, [Tor03].

La actualización de los parámetros de la red de agregación es obtenida como sigue. Denotando el vector gradiente en la iteración k como

$$\underline{e}_g^{(k)} = \sum_{t=1}^{N}\sum_{j=1}^{K} [h_j^{(k)}(t) - g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})] \frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0^{(k)}}, \tag{33}$$

y la matriz Hessiana en la iteración k como

$$\underline{R}_g^{(k)} = \sum_{t=1}^{N}\sum_{j=1}^{K} g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})[1 - g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})] \frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0} \frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0^T}. \tag{34}$$

Entonces la actualización del IRLS generalizado es dado como sigue

$$\underline{\theta}_0^{(k+1)} = \underline{\theta}_0^{(k)} + \gamma_g (\underline{R}_g^{(k)})^{-1}\underline{e}_g^{(k)} \tag{35}$$

donde γ_g es la razón de aprendizaje.
En resumen, la actualización de los parámetros del modelo ME es dado como sigue

1. (El paso E) Calcular $h_j^{(k)}(t)$ mediante la ecuación (24).

2. (El paso M) Calcular $\Sigma_j^{(k+1)}$ mediante la ecuación (28), calcular $\underline{\theta}_0^{(k+1)}$ mediante la ecuación (35), y calcular $\underline{\theta}_j^{(k+1)}$, $j = 1, \ldots, K$ mediante la ecuación (30).

6 Estimación Robusta de Parámetros para Modelo Mezcla de Expertos

La teoría de estimación de los parámetros de un modelo utilizando métodos robustos, fue desarrollada por Huber [Hub64] y [ABH+72], quien la propuso para estimar de manera robusta un parámetro de localización en un contexto no-mixto. En trabajos posteriores fue extendido al caso multivariado por [Mar76]. Campbell [Cam84] derivó los M-estimadores para Modelos de Mezcla de densidades finitas, obteniendo un algoritmo tipo EM, pero con una función de ponderación, la cuál le asignaba a cada pixel una medida de tipicidad.

6.1 M-Estimadores

El proceso de estimación de los parámetros de una función de distribución, tradicionalmente es realizado usando el método de los mínimos cuadrados (LS) o bien, el método de máxima verosimilitud (MLE). La clase específica de funcionales estadísticos que se estudian en esta tesis son los M-estimadores.

Los M-estimadores fueron propuesto por Huber [Hub64] como una generalización del estimador máximo verosímil.

Un *M-estimador* T_N es definido como la solución del problema de minimización de:

$$\sum_{i=1}^{N} \rho(\Upsilon_i, T_N) = \min_{T_N}! \tag{36}$$

donde $\rho(\gamma,\underline{\Theta})$ es una función real derivable en $\underline{\Theta}$. Equivalentemente se puede definir la estimación T_N como la solución de la ecuación de estimación:

$$\sum_{i=1}^{N}\psi(\Upsilon_i,T_N)=0 \tag{37}$$

donde $\psi(\gamma,\underline{\Theta})=\frac{\partial\rho(\gamma,\Theta)}{\partial\underline{\Theta}}$. En particular, la elección de $\rho(\gamma,\underline{\Theta})=-I$ corresponde al estimador MLE.

6.2 Robustificación del Algoritmo de Máxima Expectación

En esta sección, un método robusto de estimación de parámetros para el modelo Mezcla de Expertos utilizando el algoritmo EM es introducido. El objetivo final, es obtener los estimadores ML de los parámetros del modelo ME, considerando los datos atípicos, pues podrían aportar información valiosa y necesaria, pero limitando su influencia.

En esta sección se muestra el proceso de robustificación del algoritmo de máxima expectación para mezcla de expertos denominado REM-ME. El paso de expectación (paso E), dado por la ecuación (15) que calculaba una función $Q(\underline{\Theta}|\underline{\Theta}^{(k)})$ y los $h_j^{(k)}(t)$ para los K expertos es reformulado, robustificando la función $Q(\underline{\Theta}|\underline{\Theta}^{(k)})$, al introducir M-estimadores (propuesto por Huber en [Hub64]), denominándose esta nueva función $RQ(\underline{\Theta}|\underline{\Theta}^{(k)})$. En el paso de maximización del algoritmo REM-ME, se deberá maximizar la función $RQ(\underline{\Theta}|\underline{\Theta}^{(k)})$ para obtener una estimación del vector de parámetros que servirá como base para que en la próxima iteración se realice una nueva estimación de este vector, hasta su convergencia.

Como se puede ver experimentalmente en [TSAM03], cuando los datos están contaminados el rendimiento de este algoritmo es considerablemente reducido. Ésto se debe a los supuestos distribucionales que son realizados. Usualmente los datos atípicos no pueden ser ajustados por el modelo supuesto y por lo tanto éste pierde validez. Desafortunadamente, los datos reales no están libres de datos atípicos. En un modelo ME, bajo resultados empíricos, notamos que cuando existe este tipo de datos, el modelo tiende a tratarlos como nuevas clases, y que por tanto al aumentar el número de expertos, el modelo adquiere la capacidad de modelarlos, asignándole expertos a esos 'casos especiales'. Un primer problema es que la complejidad del modelo aumenta, y por lo tanto su velocidad de convergencia también. Un segundo problema es que la capacidad de modelar los datos atípicos será garantizado sólo durante el entrenamiento, perdiendo generalidad, debido a que no existe garantía acerca del comportamiento distribucional de estos datos en etapas posteriores.

Una solución a este problema es la utilización de métodos robustos que sean capaces de identificar estos datos atípicos y acoten su influencia en la estimación de los parámetros del modelo. El objetivo es no agregar mayor complejidad al modelo adicionando nuevos expertos para modelar un dato en particular, sino identificarlo, e identificar que mezcla de expertos es capaz de modelarlo, al menos aproximadamente. Ésto se logra gracias a la introducción de una función $\rho(\ln(\cdot))$ en cada experto en vez de la utilización del logaritmo natural $\ln(\cdot)$, de la siguiente manera:

$$\begin{aligned} RQ(\underline{\Theta}|\underline{\Theta}^{(k)}) \quad &= \textstyle\sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(t)\ln g_j(\underline{x}^{(t)},\underline{\theta}_0)+ \\ &\quad \textstyle\sum_{j=1}^{K}\left[\sum_{t=1}^{N} h_j^{(k)}(t)\rho\left(\ln P(\underline{y}^{(t)}|\underline{x}^{(t)},\underline{\theta}_j,\Sigma_j)\right)\right] \end{aligned} \tag{38}$$

donde $h_j^{(k)}(t)$ es dado por la ecuación (24). Nótese que, la función $\rho(\ln(\cdot))$ no es aplicado a la red de agregación.

La robustificación es introducida localmente, eso significa que cuando un nuevo patrón es presentado al experto, si el patrón está alejado de la mayoría de los datos que actualmente está modelando, la influencia introducida por este patrón en el paso M será limitada sin importar si el experto tiene o no la más alta probabilidad dada por la ecuación (24). Finalmente el modelo total será robusto por el hecho de ser no sensitivo a las observaciones outliers debido a la contribución de robustez de cada experto.

El problema es cómo escoger la correcta función $\rho(\cdot)$ para cumplir con la tarea de robustificación. En [HRRS86] algunas funciones especiales para los M-estimadores son discutidas. El objetivo es ponderar cada observación de acuerdo a la magnitud de la verosimilitud evaluada en la observación. Muestras con baja verosimilitud tienden a ser tratadas como datos atípicos y su ponderación es baja. En particular para problemas de localización, los datos que están muy alejados deben tener un impacto limitado en el algoritmo de estimación. Existen diferentes funciones que pueden ser utilizadas, y en particular en esta tesis se ha seleccionado la función de Huber dada por

$$\rho_H(z)=\begin{cases} z+\frac{1}{2}\log(2\pi) & \text{si } z\geq\frac{1}{2}(-k^2-\log(2\pi)) \\ -k\{-2z-\log(2\pi)\}^{\frac{1}{2}}-\frac{1}{2}k^2 & \text{e.t.o.c} \end{cases} \tag{39}$$

$$\psi_H(z) = \begin{cases} 1 & \text{si } z \geq \frac{1}{2}(-k^2 - \log(2\pi)) \\ k\{-2z - \log(2\pi)\}^{-\frac{1}{2}} & \text{e.t.o.c} \end{cases} \tag{40}$$

El paso de maximización M calcula:

$$\underline{\Theta}^{(k+1)} = \arg\max_{\underline{\Theta}} RQ(\underline{\Theta}|\underline{\Theta}^{(k)}) \tag{41}$$

El paso M escoge un valor para el parámetro que aumenta la función Q; el valor esperado del logaritmo de la verosimilitud de los *datos completos*.

La implementación del paso M, basada en las ecuaciones (5), (4) y (38), es obtenida para los parámetros de la red de agregación:

$$\frac{\partial RQ}{\partial \underline{\theta}_0} = \sum_{t=1}^{N}\sum_{j=1}^{K}[h_j^{(k)}(t) - g_j(\underline{x}^{(t)}, \underline{\theta}_0)]\frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0} \tag{42}$$

Sea

$$\psi_j^{(k)}(t) = \psi(\ln P(\underline{y}^{(t)}|\underline{x}_{(t)}, \underline{\theta}_j)) \tag{43}$$

Para los parámetros $\underline{\theta}_j$ de las redes expertas se obtiene que:

$$\frac{\partial RQ}{\partial \underline{\theta}_j} = \sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)^{-1}\frac{\partial P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)}{\partial \underline{\theta}_j} \tag{44}$$

y para los parámetros Σ_j de la redes expertas

$$\frac{\partial RQ}{\partial \Sigma_j} = \sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)^{-1}\frac{\partial P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)}{\partial \Sigma_j} \tag{45}$$

donde

$$\frac{\partial P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)}{\partial \underline{\theta}_j} = -2|\Sigma|^{-1/2}\varphi'(\cdot)\frac{\partial f_j(\cdot)}{\partial \underline{\theta}_j}x\Sigma^{-1}(\underline{y}^{(t)} - f_j(\cdot)) \tag{46}$$

y

$$\frac{\partial P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j, \Sigma_j)}{\partial \Sigma_j} = -|\Sigma|^{-1/2}|\Sigma|^{-1}\left[\frac{1}{2}\varphi(\cdot) - \varphi'(\cdot)(y^{(t)} - f_j(\cdot))(\underline{y}^{(t)} - f_j(\cdot))^T\Sigma^{-T}\right] \tag{47}$$

donde $\varphi(\cdot) = \varphi\left\{(\underline{y}^{(t)} - f_j(\cdot))^T\Sigma_j^{-1}(\underline{y}^{(t)} - f_j(\cdot))\right\}$, $\varphi'(z) = \frac{\partial\varphi(z)}{\partial z}$ y $f_j(\cdot) = f_j(\underline{x}^{(t)}, \underline{\theta}_j)$.

Para obtener las ecuaciones de la actualización de los parámetros de la red de agregación y de las redes expertas, el método de Newton puede ser utilizado. Si se considera que la densidad $\varphi(z)$ pertenece a la familia exponencial, es decir $\varphi(z) = (2\pi)^{-m/2}\exp\{-z/2\}$, y específicamente una densidad Gaussiana, y por otro lado se considera que las redes expertas son lineales, la actualización de las matrices de covarianzas es dada por:

$$\Sigma_j^{k+1} = \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)}\sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)[\underline{y}^{(t)} - f_j(\cdot)][\underline{y}^{(t)} - f_j(\cdot)]^T \tag{48}$$

La actualización de los parámetros $\underline{\theta}_j$ de las redes expertas es dado por

$$\underline{\theta}_j^{(k+1)} = (\underline{R}_j^{(k)})^{-1}\underline{c}_j^{(k)} \tag{49}$$

donde

$$\underline{c}_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)X_t(\Sigma_j^{(k)})^{-1}\underline{y}^{(t)}, \tag{50}$$

$$\underline{R}_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)}(t)\psi_j^{(k)}(t)X_t(\Sigma_j^{(k)})^{-1}X_t^T, \tag{51}$$

La actualización de los parámetros de la red de agregación es obtenida mediante el algoritmo IRLS generalizado como sigue:

$$\underline{\theta}_0^{(k+1)} = \underline{\theta}_0^{(k)} + \alpha(\underline{R}_0^{(k)})^{-1}\underline{e}_0^{(k)} \tag{52}$$

donde α_0 es la razón de aprendizaje de la red de agregación y

$$\underline{e}_0^{(k)} = \sum_{t=1}^{N}\sum_{j=1}^{K}[h_j^{(k)}(t) - g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})]\frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0^{(k)}}, \tag{53}$$

y

$$\underline{R}_0^{(k)} = \sum_{t=1}^{N}\sum_{j=1}^{K} g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})[1 - g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)})]\frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0}\frac{\partial \underline{\xi}_j}{\partial \underline{\theta}_0^T}. \tag{54}$$

En resumen, la actualización de los parámetros del modelo ME es el siguiente:

1. (Paso E) Calcular $h_j^{(k)}(t)$, $\psi_j^{(k)}(t)$, para cada experto ($j = 1,\ldots,K$), mediante las ecuaciones (24) y (43) respectivamente.
2. (Paso M) Estimar $\Sigma_j^{(k+1)}$, $\underline{\theta}_j^{(k+1)}$, $j = 1,\ldots,K$ y $\underline{\theta}_0^{(k+1)}$ mediante las ecuaciones (48) (49), (52).

6.3 Robustificación del Algoritmo de Aprendizaje basado en el Gradiente

Detalles de este algoritmo pueden ser revisados en [Tor03].

En particular, si se considera que las redes expertas y la red de agregación son redes lineales que siguen un modelo Gaussianoy si se utiliza la función softmax para las salidas de la red de agregación, dada por la ecuación (4), entonces la actualización de los parámetros de la red está dada por la siguiente expresión:

$$\begin{aligned}
\Delta\underline{\theta}_i &= \alpha\psi(\zeta)g_i\frac{1}{(2\pi)^{m/2}}\exp\left\{-\frac{(\underline{y}^{(t)}-\underline{\mu}_i)^T(\underline{y}^{(t)}-\underline{\mu}_i)}{2}\right\}(\underline{y}^{(t)} - \underline{\mu}_i^{(t)})\underline{x}^{(t)^T} \\
\Delta\underline{\theta}_{0i} &= \alpha g_i\left\{P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_i) - \textstyle\sum_k g_k P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_k)\right\}\underline{x}^{(t)^T}
\end{aligned} \tag{55}$$

donde α es la razón de aprendizaje.

7 Resultados y Comparaciones

El criterio de Prechelt [Pre94] establece que:

> Una evaluación a un algoritmo es denominada aceptable, si ésta utiliza un mínimo de dos problemas reales y compara esos resultados con al menos un método alternativo. Prechelt [Pre94].

Para mostrar el rendimiento que se obtiene al robustificar el algoritmo de aprendizaje basado en el gradiente y el algoritmo EM (REM-ME) se utilizan conjuntos de datos de regresión de dos fuentes: DELVE y PROBEN1.

DELVE (Datos para evaluar aprendizaje mediante experimentos válidos) es una colección de conjuntos de datos de muchas fuentes, un ambiente dentro de los cuáles los datos pueden ser usados para medir el rendimiento de los algoritmos de aprendizaje y un repositorio para los resultados de tales experimentos.

7.1 Experimentos para el Modelo ME en conjunto de datos *Boston* (DELVE)

El conjunto de datos 'Boston Housing Data' creada por David Harrison y Daniel Rubinfeld, está compuesta de 506 muestras, cada una con trece entradas, compuestas de 12 atributos continuos y un atributo binario representando las características de los vecindarios en el área de Boston y una salida, denominada 'atributo clase', representado el valor promedio de una casa en esos vecindarios. En [Tor03] se entrega una descripción y el rango de los atributos de entrada y salida, que son normalizados entre [-1,1] y luego desnormalizados para ser presentado en las tablas de resultados.

Para todos los experimentos que se realizaron en este trabajo, se dividió el conjunto de datos de tamaño N en 50% de los datos para el conjunto de entrenamiento, 25% de los datos para el conjunto de validación y el 25% restante de los datos para el conjunto de prueba. Nótese que el conjunto de datos original está ordenado según el vecindario. Los experimentos realizados fueron:

Debido a que el conjunto de Datos Boston original proviene de diferentes vecindarios de Boston donde las realidades económicas y culturales son distintas, es caracterizado como un problema con diferentes fuentes de datos apto para ser

modelado por un modelo ME. Las muestras tomadas para este problema son datos reales ruidosos. Por otro lado, la presentación de este conjunto de datos está ordenado por vecindario, lo que introduce una nueva dificultad en el sentido de que no todas las clases están igualmente representadas en los subconjuntos de datos (entrenamiento, validación, prueba) pudiendo incluso haber presencia en un subconjunto y ausencia en otro.

La conjetura realizada en este experimento es que la robustez en los algoritmos de aprendizaje permiten obtener mejores resultados en los conjuntos de prueba. El tamaño del conjunto de datos de entrenamiento es de 253 datos, lo cual nos entrega una cota superior para el número de expertos posibles de incluir en la mezcla, en este caso 15 expertos.

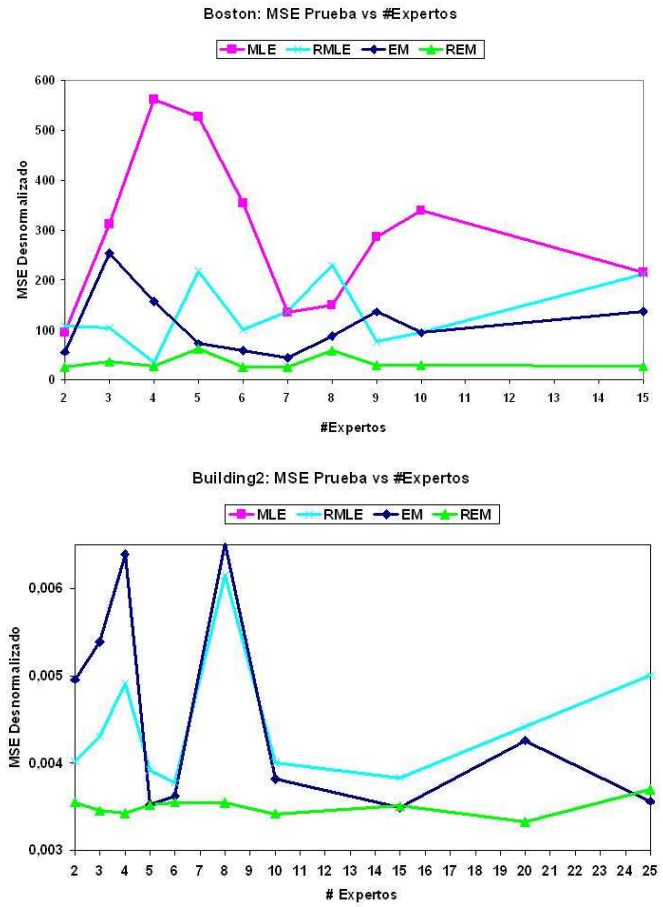

Figure 3: *Resultados*: MSE obtenido durante la fase de pruebas versus el Número de Expertos para el conjunto de datos Boston y el conjunto de Datos Building2 respectivamente. Notar que para el conjunto de datos Building 2, el gráfico es una ampliación de tres de los algoritmos, donde el algoritmo MLE por su bajo performance no aparece.

En la figura 3 se muestran los resultados obtenidos para el modelo de Mezcla de Expertos, utilizando los algoritmos basados en el gradiente, gradiente robusto, el algoritmo EM y REM.

En la Figura 4 se muestran los resultados obtenidos para el conjunto de datos con 2 expertos (ver [Tor03]).

Para este experimento, concluimos que el algoritmo REM presenta mejores resultados, debido a que es insensible a los datos atípicos, acotando la influencia que ellos ejercen en los parámetros del modelo ME. Aunque pareciera ser evidente que el algoritmo RMLE (gradiente robusto) presentase similares resultados, esta hipótesis se hace falsa debido a la incapacidad de este algoritmo de tomar ventaja de la estratificación de los datos de entrada, estratificación que se produce debido a que los datos provienen de distintos vecindarios de Boston, donde las realidades entre éstos no son iguales la mayoría de las veces. En cambio, el algoritmo REM se basa en el algoritmo EM, lo cual le adiciona la facultad de tomar ventaja de la modularidad del modelo, por lo que es más efectivo cuando los datos provienen de distintas fuentes.

Detalles acerca de otra serie de experimentos realizados sobre esta serie, tales como sobre el conjunto de datos ajustados

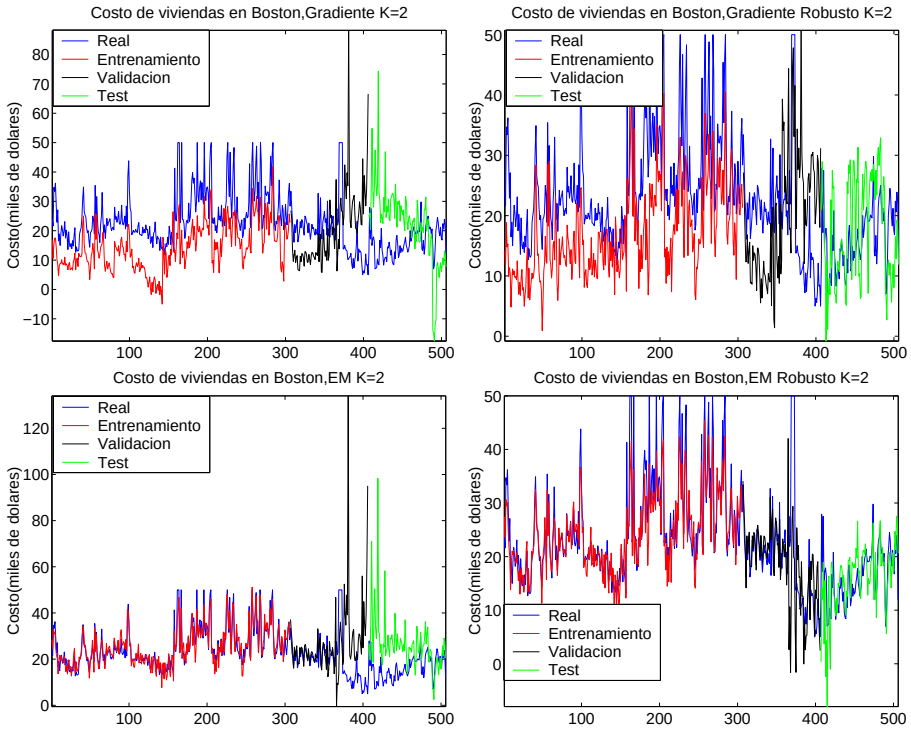

Figure 4: *Mezcla de* 2 *expertos*: Resultados para el conjunto de datos Boston Original utilizando 2 expertos y utilizando 4 algoritmos de aprendizaje diferentes. (arriba-izquierda) Gradiente, (arriba-derecha) Gradiente RObusto. (abajo-izquierda) Algoritmo EM. (abajo-derecha) Algoritmo REM.

o desordenados, donde también es mostrada la superioridad del algoritmo robusto pueden ser encontrados en [Tor03].

7.2 Experimentos para el Modelo ME en conjunto de datos *Building2* (PROBEN1)

Consiste de un problema de consumo de energía en un edificio. Este conjunto de datos *Building2* pertenece a la colección de problemas de aproximación para aprendizaje en Redes Neuronales, *PROBEN1* [Pre94]. Se trata de predecir el consumo de energía eléctrica, agua caliente y agua fría, basado en información acerca del día de la semana, hora del día, temperatura externa, humedad del aire externa, radiación solar y velocidad del viento. Por lo tanto el vector de entrada consiste de 14 atributos y el vector de salida es de dimensión 3. La muestra proviene de observaciones realizadas durante seis meses (desde Septiembre a Febrero). Acerca de los datos, observaciones de diversos investigadores, tales como [Pre94] observan que los requerimientos de agua caliente tienen una fuerte correlación con la temperatura externa [Tor03].

El tamaño del conjunto de datos, N es 4208 patrones, de los cuales 2104 patrones fueron utilizados para el conjunto de entrenamiento, 1052 patrones para el conjunto de validación y los 1052 datos restantes para el conjunto de prueba.

En la figura 3 se muestran los resultados obtenidos para el modelo de Mezcla de Expertos, utilizando los algoritmos basados en el gradiente, gradiente robusto, el algoritmo EM y REM.

De este experimento se puede inferir que robustificar el algoritmo de aprendizaje, algoritmos RMLE y REM, la complejidad es del modelo ME decrece en cuanto al número de expertos, y por tanto a la cantidad de parámetros a estimar del modelo. Esto es claro en la figura 3 donde se observa que para 3 o 10 expertos por ejemplo en el caso del algoritmo REM, se obtiene un error relativamente similar.

Según el test de Pretchel debemos comparar nuestros resultados con al menos un algoritmo alternativo. Para ésto. hemos entrenado una red de tres capas completamente conectada. El número de neuronas de la red en la capa escondida es escogido empíricamente, basándose en el rendimiento obtenido en el conjunto de prueba. Se utilizan distintas variantes de algoritmos de aprendizaje y optimizaciones para lograr un modelo de una única red de tres capas para compararlo con los resultados obtenidos por el modelo ME con algoritmos de aprendizaje MLE, RMLE, EM y REM.

Para el entrenamiento de la red multicapa se utilizan tres métodos alternativos para el aprendizaje:Algoritmo de aprendizaje backpropagation descendiente (GD), Algoritmo de aprendizaje backpropagation descendiente con momento (GDM) y Algoritmo de aprendizaje backpropagation basado en el método Levenberg-Marquardt(LM).

El resumen de los resultados pueden ser apreciados en la tabla 1 y 2.

Algoritmo	**Entrenamiento**	**Validacion**	**Prueba**
LM	2,260368	89,390206	50,9428
GDM	44,313272	84,339018	35,758352
MLE	44,945971	203,195314	134,599954
RMLE	52,781237	111,285037	34,943334
EM	9,208113	83,03982	44,594861
REM	12,152561	74,165716	25,197388

Table 1: *Boston Original*: Resumen de los resultados obtenidos sobre el conjunto de Datos Boston Original

Algoritmo	**Entrenamiento**	**Validacion**	**Prueba**
GD	0,007632	0,007555	0,007616
GDM	0,0076	0,0078	0,008
MLE	0,008324	0,008414	0,0086
RMLE	0,003832	0,004323	0,004
EM	0,003380	0,003704	0,003489
REM	0,003225	0,003574	0,003329

Table 2: *Building2*: Resumen de los resultados obtenidos sobre el conjunto de Datos Building2

En la Figura 5 se muestran los resultados obtenidos para el conjunto de datos con 33 expertos (ver [Tor03]).

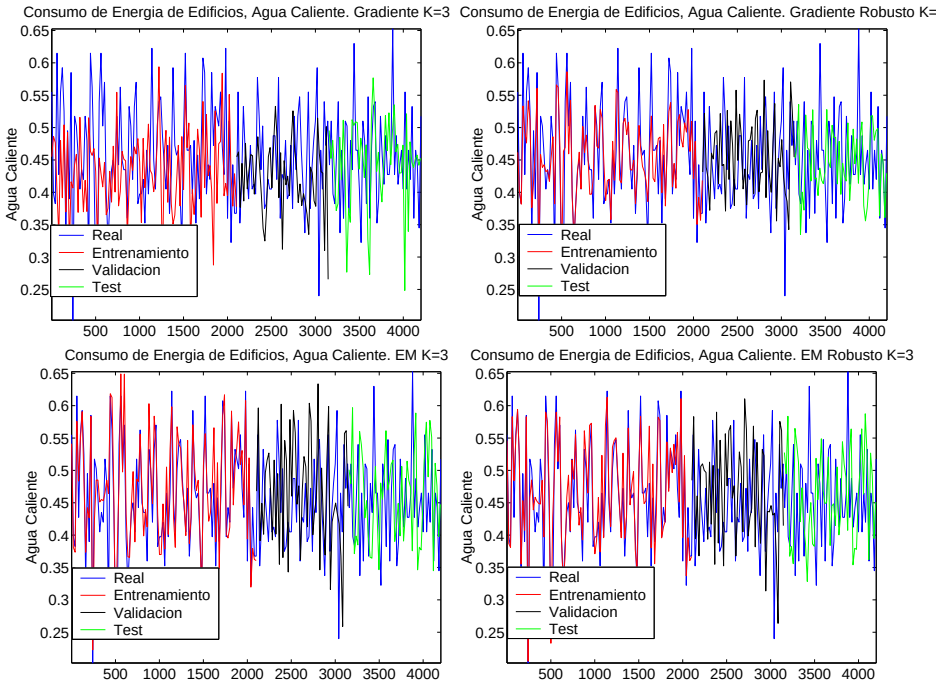

Figure 5: *Mezcla de* 3 *expertos*: Resultados para la predicción del consumo de agua caliente para el conjunto de datos Building 2 utilizando 3 expertos y utilizando 4 algoritmos de aprendizaje diferentes. (arriba-izquierda) Gradiente, (arriba-derecha) Gradiente RObusto. (abajo-izquierda) Algoritmo EM. (abajo-derecha) Algoritmo REM.

8 Conclusiones

Al utilizar el modelo Mezcla de Expertos con el algoritmo de aprendizaje robustificado en problemas de regresión (donde los datos presentaban datos atípicos) se lograron mejoras sinificativas según el test de Prechelt. Los conjuntos de datos reales que se utilizaron estaban altamente contaminados y al compararlo con las versiones no robustas, tanto el gradiente robusto y el REM mostraron ser superiores cuando aplicar robustez se justificaba, es decir cuando los datos contenían datos atípicos. La mejora que presenta este algoritmo es significativa según el test de Prechelt.

Basado en cada una de la hipótesis realizadas al comienzo de esta tesis, se hará un análisis de cada una.

Elección del Modelo ME: Basándonos en el conjunto de datos Boston original, donde los datos acerca del costo promedio de una casa dependía fuertemente de cuál vecindario se estaba interesado, lo cual podía ser visto como distintas fuentes de datos, se mostró que debido a la estratificación del espacio un modelo ME era el más adecuado, y en particular la utilización del algoritmo EM que saca ventaja de la modularidad del modelo obtenía un error cuadrático medio menor al presentado por otras técnicas alternativas.

Mejoras significativas en los conjuntos de prueba por utilizar algoritmos robustos durante el aprendizaje: esta hipótesis es verdadera en todos los casos de prueba siempre y cuando los datos presenten datos atípicos, es decir donde la robustez se justifica.

Reducción del número de expertos del modelo ME al utilizar algoritmos robustos durante el aprendizaje: esta hipótesis es verdadera. Nótese que cuando se utiliza algoritmos robustos de aprendizaje, el número de expertos necesarios para alcanzar igual precisión que las versiones de los algoritmos no robustos, es mucho menor, debido a que la robustez permite detectar aquellos datos que son atípicos y que por lo tanto no reflejan una real tendencia de los datos, entonces el modelo ME, no los trata como una nueva clase y por tanto no los modela de manera especial asignándoles un experto o una mezcla de expertos nueva sino que los agrupa a una clase de datos similar, que es donde realmente corresponde.

El modelo ME converge al menos un orden de magnitud más rápido si se utiliza durante el aprendizaje el algoritmo EM en vez de un algoritmo basado en el gradiente: esta hipótesis fue confirmada debido a que el algoritmo EM saca ventaja de la modularidad del modelo ME, y por lo tanto la validez de esta hipótesis es también para datos que provienen de distintas fuentes.

Cuando los conjuntos de datos presentan datos atípicos, el modelo ME con aprendizaje robusto presentará mejoras significativas en el rendimiento: siempre que los conjuntos de datos presenten datos atípicos o contaminados, las versiones robustas de los algoritmos de aprendizaje son superiores no sólo en las etapas de entrenamiento y validación sino en la de prueba, siendo esta más importante debido a que corresponde a un conjunto de datos antes no visto por el sistema.

Gracias a los experimentos realizados en esta tesis, podemos concluir que, el rendimiento del modelo Mezcla de Expertos presentado en [JJNH91b], puede ser mejorado si se utilizan algoritmos robustos de aprendizaje. En particular si se utiliza el algoritmo REM, se obtiene un modelo ME, de rápida convergencia, insensible a los datos atípicos en el sentido que extrae la información relevante de éstos pero acotando su impacto, de alta capacidad de generalización durante la fase de prueba o funcionamiento del sistema, sin aumentar la complejidad de la arquitectura, pues puede trabajar con el número mínimo de expertos necesarios sin degradar su rendimiento. Si se utiliza el algoritmo robusto basado en el gradiente, referido en esta tesis como RMLE, se obtienen las mismas bondades que para el algoritmo REM, a excepión de la rápidez de convergencia debido a que los algoritmos basados en el gradiente, tanto robustos como no robustos, no sacan ventaja de la modularidad del modelo de mezcla.

References

[ABH+72] D. Andrews, P. Bickel, F. Hampel, P-Huber, W. Rogers, and J. Tukey. Robust estimate of location: Survey and advances. Technical report, Princeton University Press, Princeton, N.J., 1972.

[AMS01] H. Allende, C. Moraga, and R. Salas. Neural model identification using local robustness analysis. *Lecture Notes in Computer Science. Fuzzy Days 2001*, 2206:162–173, Nov 2001.

[AMS02] H. Allende, C. Moraga, and R. Salas. Robust estimator for the learning process in neural networks applied in time series. *Lecture Notes in Computer Science, ICANN 2002*, 2415:1080–1086, Aug 2002.

[AMST04] H. Allende, C. Moraga, R. Salas, and R. Torres. Modular Neural Network applied to non-stationary Times Series. *Accepted in Advances in Soft Computing*, 2004.

[ATSM03] H. Allende, R. Torres, R. Salas, and C. Moraga. Robust learning algorithm for the mixture of experts. *Lectures Notes in Computer Science. IBPRIA2003*, 2652:19–27, Jun 2003.

[Cam84] N. A. Campbell. Mixture models and atypical values. *Math. Geol.*, pages 465–477, 1984.

[CM94] J.T. Connor and R.D. Martin. Recurrent neural networks and robust time series prediction. *IEEE Transactions of Neural Networks*, 2(5):240–253, 1994.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. 39:1–38, June 1977.

[HRRS86] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W. A. Stahel. *Robust Statistics, The approach based on Influence Functions*. Wiley Series in probability and mathematical statistics, 1986.

[Hub64] P. J. Huber. Robust estimation of a location parameter. *Ann. Math. Statist.*, 16, 1964.

[JJ91] R. A. Jacobs and M. I. Jordan. A modular connectionist architecture for learning piecewise control strategies. *Proceedings of the American Control Conference*, 2.:1597–1602, 1991.

[JJ92] M. I. Jordan and R. A. Jacobs. Hierarchies of adaptive experts. pages 985–992. 1992.

[JJ94] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[JJ99] M. Jordan and R. Jacobs. *Modular and hierarchical learning systems, The Handbook of Brain Theory and Neural Networks, Cambridge, MA*, volume 1. MIT Press, 1999.

[JJB91] R. A. Jacobs, M. I. Jordan, and A. G. Barto. Task decomposition through competition in a modular connectionist architecture - the What and Where vision tasks. *Cognitive Science*, 15(2):219–250, 1991.

[JJNH91a] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[JJNH91b] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[Mar76] R. A. Maronna. Robust M-estimators of multivariate location and scatter. *Ann. Statist.*, 4, 1976.

[MP01] G. McLachlan and T. David Peel. *Finite Mixture Models*, volume 1. Wiley Series in Probability and Statistics, 2001.

[Pre94] L. Prechelt. PROBEN1 – a set of benchmarks and benchmarking rules for neural training algorithms. *Technical Report 21/94, Fakultat fur Informatik, Universitat Karlsruhe, D-76128 Karlsruhe, Germany*, 1994.

[Tor03] R. Torres. Algoritmo Robusto de Aprendizaje para el Modelo Mezcla de Expertos. *Tesis de Magíster en Ciencias de la Ingeniería Informática, Universidad Técnica Federico Santa María*, Noviembre, 2003.

[TSAM02] R. Torres, R. Salas, H. Allende, and C. Moraga. Estimador robusto en modelos de mezcla de expertos locales. *CLATSEV*, Nov 2002.

[TSAM03] R. Torres, R. Salas, H. Allende, and C. Moraga. Robust expectation maximization learning algorithm for mixture of experts. *Lectures Notes in Computer Science. IWANN2003*, 2686:238–245, Jun 2003.