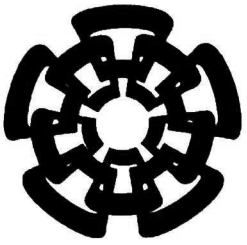

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Evaluación de la Fragmentación en el Aprendizaje de Árboles de Decisión Basada en el Análisis de las Particiones en Agrupaciones de Datos

Tesis que presenta:

Roberto Valerio Molina

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Mario Angel Siller González Pico
Dr. Ricardo Vilalta López

Guadalajara, Jalisco, Septiembre de 2009.

Evaluación de la Fragmentación en el Aprendizaje de Árboles de Decisión Basada en el Análisis de las Particiones en Agrupaciones de Datos

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Roberto Valerio Molina
Ingeniero en Sistemas Computacionales
Instituto Tecnológico y de Estudios Superiores de Monterrey
2002-2007

Becario de CONACYT, expediente no. 213032

Directores de Tesis
Dr. Mario Angel Siller González Pico
Dr. Ricardo Vilalta López

CINVESTAV del IPN Unidad Guadalajara, Septiembre de 2009.

Agradecimientos

Quisiera agradecer a mis padres por su amor y su apoyo. Aquellas dos personas tan especiales para mi, por ser quienes a lo largo de toda mi vida, han estado ahí. Gracias a ustedes he logrado ser quien soy.

También quiero agradecer a mi asesor el Dr. Ricardo Vilalta quien me guió en mi investigación y me ayudo a resolver los problemas encontrados en este proceso. Sin su ayuda no hubiera sido posible realizar este trabajo, llevar a cabo la investigación y por lo tanto obtener el grado.

Quisiera agradecer especialmente al Dr. Mario Angel Siller Gonzalez Pico y al Dr. Felix Ramos Corchado, no solamente por su apoyo y consejo durante mi estancia en el CINVESTAV, sino por hacerme sentir su aprecio y preocuparse de manera constante en el desarrollo de mis actividades realizadas en este centro de investigación.

Al Dr. Andrés Méndez, quien a pesar del corto tiempo de conocerlo me brindo consejo y ayudo a tener una mejor visión acerca de mi futuro.

A Melissa por su gran apoyo en estos días, gracias por brindarme siempre una sonrisa y compartir tu felicidad conmigo.

A toda la comunidad del CINVESTAV, porque cuando se cae una pieza por mas pequeña e insignificante que parezca; toda la maquinaria deja de funcionar.

Al CONACYT por otorgar los recursos necesarios para realizar todas las actividades que se realizaron durante el desarrollo de esta tesis.

Y a Dios, quien a cuidado de mi a lo largo de toda mi vida.

Además quiero mencionar a todas las personas que de alguna manera estuvieron inmersas en el proceso de desarrollar esta tesis:

- Mi Familia: Rosi, Alhelí, Jorge, Tano, Jorge Alejandro y Natalia Alhelí
- Mis Amigos: Nora, Metales, Victorio, Alma Verónica, Dann, Ernesto, Mario y Andrea.
- Mis compañeros de investigación

- Mis profesores

Gracias a todos ustedes.

Prefacio

Los árboles de decisión constituyen una solución adecuada al problema de la clasificación, debido a su habilidad de dividir el espacio de entrada (variables) en regiones con una distribución de clases uniforme en los eventos. El árbol de decisión además provee un modelo fácil de interpretar, a diferencia de otros métodos como las redes neuronales. Sin embargo este método cuenta con una limitación inherente, que es la disminución progresiva del soporte estadístico del clasificador final debido a que las agrupaciones de eventos de la misma clase son fragmentados en cada partición, un problema conocido como el problema de la fragmentación. En esta tesis se describe una métrica para medir el grado de fragmentación causado por el árbol de decisión en cada una de las agrupaciones de eventos. Estas agrupaciones se encuentran mediante la descomposición de los datos utilizando la técnica de Spectral Clustering. Donde cada agrupación es analizada en términos de el número y tipo de particiones inducidas por el proceso de inducción de los arboles de decisión. Nuestro dominio de aplicación recae en la búsqueda de la partícula sub-atómica single top quark, un problema difícil debido a la similitud entre esta y otras partículas como W+jets y $t\bar{t}$, además de las señales de baja energía y un pequeño numero de jets. Al final del proceso la métrica genera una serie de estadísticas describiendo el grado de errores en la clasificación atribuido al problema de la fragmentación.

Preface

Decision tree learning constitutes a suitable approach for classification due to its ability to partition the input (variable) space into regions of class-uniform events, while providing a structure amenable to interpretation (as opposed to other methods such as neural networks). But an inherent limitation of decision tree learning is the progressive lessening of the statistical support of the final classifier as clusters of single-class events are split on every partition, a problem known as the fragmentation problem. We describe a metric that measures the degree of fragmentation caused by a decision tree learner on every event cluster. Clusters are found through a decomposition of the data using a technique known as Spectral Clustering. Each cluster is analyzed in terms of the number and type of partitions induced by the decision tree. Our domain of application lies on the search for single top quark production, a challenging problem due to large backgrounds (similar to W+jets and $t\bar{t}$ events), low energetic signals, and low number of jets. At the end of the process the metric will produce a series of statistics describing the degree of classification error attributed to the fragmentation problem.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

En este capitulo se presenta una breve introducción a esta tesis. Primero se presentan los antecedentes y la motivación por la cual se aborda el problema de la fragmentación. Después se presenta una breve introducción al problema. Finalmente se presenta la propuesta que se propone para medir de manera efectiva esta fragmentación.

1.1. Antecedentes y Motivacion

El aprendizaje maquina o reconocimiento de patrones intenta encontrar similitudes entre diferentes entidades pertenecientes a una población que conforman la base de datos a examinar sobre la cual se está trabajando. Estas similitudes nos ayudan a definir grupos de entidades llamadas clases, cada uno de los elementos que pertenecen a una clase comparten algunas o muchas características con todos los demás elementos que forman la clase.

La extracción de los atributos o las características comunes que definen la pertenencia a una clase es el objetivo principal de los algoritmos de clasificación. Estos algoritmos no trabajan por si solos, necesitan ser entrenados para obtener un comportamiento correcto. Para realizar el entrenamiento es necesario un conjunto de entrenamiento, el cual es una base de datos especial. Esta base de datos tiene un atributo extra que no existe en la base de datos que deseamos clasificar, este atributo extra es el atributo clase, el cual determina a cual clase pertenecen los elementos. Con esta información el algoritmo de clasificación puede agrupar los elementos que comparten la misma clase y al mismo tiempo proveer le al usuario información acerca de cómo limitar las clases para procesar datos sin clasificar.

Debido a la existencia de múltiples algoritmos de clasificación, tenemos la necesidad de escoger uno con el cual trabajar. Tomar esta decisión es uno de los grandes problemas en aprendizaje por computadora, problema conocido como el problema de selección del algoritmo[11]. Esto quiere decir que aun en la actualidad; no se tiene idea de cómo determinar cuál es el

mejor algoritmo para realizar la clasificación de la base de datos con la que estamos trabajando.

El otro problema con el que nos enfrentamos es el hecho de que al seleccionar un algoritmo en particular existen diversos factores que determinan la precisión de la clasificación del método seleccionado. Para obtener la mejor clasificación es necesario realizar diversas pruebas entre varios métodos y sus variantes para verificar realmente cual es el mejor algoritmo para clasificar de mejor manera los datos deseados.

Sin embargo a pesar de todas las modificaciones y nuevos algoritmos propuestos solamente se ha llegado a tener una ilusión de progreso[5]. Los árboles de decisión son un claro ejemplo de estas falsas ilusiones de progreso, uno de los mas viejos algoritmos de clasificación y sin embargo aun no se ha podido resolver el problema de la fragmentación. Se han generado una gran cantidad de contribuciones para tratar de resolver el problema, sin embargo ninguna ha tenido éxito.

1.2. Definición del problema

En este trabajo se examinara uno de los más famosos algoritmos en reconocimiento de patrones. El algoritmo de aprendizaje de árboles de decisión, este algoritmo define de manera iterativa un corte en el espacio de entrada, hasta que la mayoría de los elementos en las regiones generadas pertenecen a una misma clase. Dicho corte tiene como función el separar en diferentes regiones a elementos que pertenecen a clases diferentes y agrupar en la misma región a elementos que pertenecen a la misma clase. La decisión de donde realizar dicho corte del espacio de entrada e controlada por las funciones de impureza que nos permiten obtener diferentes modelos de los mismos datos dependiendo de la función de impureza utilizada.

Debido al criterio de divide y vencerás utilizado en los árboles de decisión para general el modelo, existe una disminución de la importancia estadística de los datos con cada corte, debido a que se disminuye la cantidad de datos en las regiones con cada nuevo corte. Este problema, mejor conocido como el problema de la fragmentación, conduce a un modelo de salida que genera errores de clasificación cuando se aplica a los datos no vistos. En esta tesis se propone un método para medir la cantidad de fragmentación ejercida por el modelo del árbol de decisión sobre las características del espacio de entrada.

1.3. Propuesta

La fragmentación es medible a través del método propuesta en esta tesis, el cual consiste en: encontrar las agrupaciones de los datos de entrada. Una vez que se cuenta con las

agrupaciones, se generan los hiper-rectángulos o regiones en las cuales se divide el espacio de entrada. Estas regiones se obtienen del modelo del árbol de decisión. Finalmente se traslapan las agrupaciones sobre los hiper-rectángulos para obtener el número de fragmentos en los cuales las agrupaciones son divididas por el modelo. Con esta información se obtienen las estadísticas concernientes a como se fragmentan los datos y la importancia que se le da a cada fragmento.

1.3.1. Hipótesis

La métrica existente para medir la fragmentación, al estar basada en el numero de nodos finales del árbol, no genera información útil para identificar los puntos críticos del modelo del árbol de decisión donde se genera un mayor grado de fragmentación.

La propuesta se basa en el análisis de los datos y el modelo con el cual se espera determinar estadísticas tales como la cantidad de particiones y la evaluación de los fragmentos, con dichas estadísticas se desea identificar los puntos críticos para efectos de mejora.

1.3.2. Objetivos

Llegar a desarrollar una métrica de la fragmentación existente en un modelo de árboles de decisión. La cual no solamente nos brinde una cifra para compararla con otros modelos, sino que también nos brinde información útil para encontrar las partes del árbol donde se están generando los errores de clasificación.

La métrica propuesta debe de ser congruente con la métrica actual, es decir debe de poder determinar cual de los modelos presenta una mayor o menor fragmentación, y al mismo tiempo ser congruente. Es decir si la métrica actual de un modelo define una mayor fragmentación que otro modelo, la métrica propuesta debe de determinar un comportamiento similar.

Capítulo 2

Marco teórico y estado del arte

En este capitulo se presenta el marco teórico el cual consiste en una introducción al tema. Donde se presenta toda la información que el lector necesita para comprender el tema sin necesidad de ser un experto del mismo. Se presenta la información referente al reconocimiento de patrones, clasificación, los árboles de decisión y el problema de la fragmentación.

También se presenta el estado del arte, donde se presenta un resumen de los trabajos relacionados con el problema de la fragmentación. Cada uno de estos trabajos presenta una manera de resolver el problema de la fragmentación. Sin embargo ninguno de estos trabajos presenta un enfoque donde se caracterice o se trate de medir el problema. Por lo tanto a partir de estos trabajos podemos concluir que hace falta una métrica que ayude a identificar los puntos críticos en los árboles de decisión, para poder realizar una disminución efectiva de la fragmentación en un modelo de árboles de decisión.

2.1. Marco teórico

2.1.1. Reconocimiento de patrones

El reconocimiento de patrones es una sub-área de la inteligencia artificial, la cual toma como entrada información y realiza una acción dependiendo de la categoría del patrón. Esta es una actividad que los humanos realizamos de manera inconsciente día a día, por ejemplo cuando buscamos las llaves de nuestra casa en la bolsa de nuestro pantalón. A través de nuestro sentido del tacto, obtenemos la información de los diferentes objetos que se encuentra en nuestra bolsa, (p.e. Monedas, llaves del carro, celular, llaves de la casa, chicles, etc.) hasta que encontramos las llaves correctas y las sacamos del bolsillo.

Cuando estamos utilizando el reconocimiento de patrones para diferenciar entre varios elementos que pertenecen a clasificaciones diferentes, debemos de tomar en cuenta las carac-

terísticas que diferencian a un tipo de elementos de otro. El conjunto de características que definen a cada una de las clases, es un modelo. El objetivo principal del reconocimiento de patrones es el encontrar este modelo para generar una correcta clasificación de los elementos y por ende tomar la mejor decisión.

Para poder generar el modelo es necesario tener una serie de elementos de los cuales conocemos su clasificación y de esta manera podemos extraer las características de los elementos que pertenecen a una misma clase. Este conjunto de elementos es nuestro conjunto de entrenamiento. Este conjunto de datos es primordial para la clasificación, debido a que el modelo se genera a partir de estos elementos, si este conjunto de datos no es realmente representativo del problema que queremos abordar el modelo generado tendrá errores.

El modelo genera separaciones entre clases, esta separación genera los borde de decisión. Los bordes de decisión delimitan el espacio de variables que describen las características de los elementos. Dependiendo de como se genero el modelo son las características de los bordes de decisión. Si el modelo es demasiado especifico el borde de decisión sera muy complicado, si el modelo es demasiado generalizado el borde especifico sera muy simple. Además un borde general tiene mas holgura para clasificar elementos que no han sido tomados en consideración en la generación del modelo, mientras que un modelo demasiado especifico puede generar errores en la clasificación.

Aprendizaje

El aprendizaje hace referencia a un conjunto de técnicas o algoritmos que reducen el error en un conjunto de entrenamiento, un claro ejemplo de este tipo de algoritmos es del tipo de gradiente descendiente en el cual se van alterando los parámetros de clasificación con el objetivo de reducir el error. Existen tres tipos principales de aprendizaje

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje por refuerzo

En el aprendizaje supervisado se cuenta con un identificador el cual aporta la categoría a la cual pertenece cada elemento que forma el conjunto de entrenamiento. A partir de estas categorías se aprende el modelo de clasificación que caracteriza a la base de datos de entrenamiento. Un ejemplo clásico de este tipo de aprendizaje son los arboles de decisión.

En el aprendizaje no supervisado no existen las etiquetas de a que clase pertenece cada elemento. Aquí se examina la estructura de los datos, su cercanía y similitudes encontradas entre ellos, a partir de esta información se generan grupos de elementos que comparten las

mismas características y por lo tanto forman una clase. Un ejemplo de este tipo de algoritmos es el agrupamiento espectral.

Por ultimo el aprendizaje por refuerzo es un claro ejemplo de como aprendemos los humanos. Primero se genera el modelo con el conjunto de entrenamiento y se prueba este modelo. Después de probar el modelo se genera un reporte acerca de los errores que cometió este modelo. A partir de este reporte se generan modificaciones en el modelo y se vuelve a probar y a comparar su precisión. Este proceso es repetido hasta que el error desaparece o llega a un mínimo aceptable. Este tipo de aprendizaje es utilizado para mejorar los algoritmos realizados con aprendizaje supervisado y no supervisado.

2.1.2. Clasificación

En la clasificación buscamos encontrar las similitudes existentes entre elementos que pertenecen a la misma clase y las diferencias entre elementos de clases diferentes. Con estas similitudes y diferencias se genera un modelo de clasificación que realiza esta separación de manera automática.

La separación automática de una señal de la medición cuando no existe dicha señal puede verse como un problema de clasificación. En el que se asume como entrada un espacio de entrenamiento $S = \{(\mathbf{x}, y)\}$, donde cada evento $\mathbf{x} = (a_1, a_2, \cdots, a_n)$ es un vector de variables o atributos (por ejemplo, energía, momentum, masa invariante, etc.). El vector $\mathbf{x}$ es un punto en el espacio de entrada $\mathcal{X}$, y le asignamos la etiqueta de una clase y del espacio de salida $\mathcal{Y} = \{0, 1\}$, con $y = 0$ representando la ausencia de señal y $y = 1$ representando la señal.

El resultado del clasificador es una función f que mapea el espacio de entrada al espacio de salida, $f : \mathcal{X} \rightarrow \mathcal{Y}$. La función f puede ser usada para predecir la clase de eventos que no habían sido vistos con anterioridad. Nuestro interés principal es la habilidad de que f pueda predecir correctamente la clase de eventos fuera de S.

En la clasificación se buscan hipótesis que no solo sean consistentes con S, sino que también generalicen mas allá de este conjunto.

Problemas en la clasificación.

Actualmente existen múltiples algoritmos que nos ayudan a resolver el problema de la clasificación, además todos los días se generan nuevas modificaciones a los algoritmos o surgen nuevas propuestas de solución, sin embargo la mejoría en el comportamiento de estas nuevas versiones es muy pequeña al ser comparados con sus antecesores. Esto nos provee una ilusión de progreso justo como lo presenta Hand[5].

Los clasificadores pueden ser demasiado buenos para ciertos tipos de datos, pero tener un

comportamiento pésimo con otros tipos de datos de datos. Esto hace que las comparaciones de efectividad con otros métodos pierdan la credibilidad, debido a que se aparenta tener una ganancia en el rendimiento del algoritmo pero puede ser que esto no sea del todo cierto, debido a que al hacer las comparaciones con otros tipos de datos podemos llegar a obtener ganancias nulas o negativas. Por lo tanto existe la necesidad de crear una forma de determinar cuáles clasificadores presentan el mismo comportamiento con los mismos tipos de bases de datos y de esta manera poder obtener una comparación justa. Lo cual nos lleva al problema de selección del algoritmo presentado por Rice[11], el problema es que no existe una manera de determinar cuál es el mejor algoritmo para ser usado de acuerdo con los datos que se desea procesar, lo cual nos imposibilita realizar dicha categorización de clasificadores con comportamientos similares sobre todos los tipos de datos.

Además contamos también con el problema de sobre entrenamiento, el cual ocurre cuando el algoritmo caracteriza de manera exacta los datos de entrenamiento. Cuando estos modelos se usan para clasificar bases de datos con eventos diferentes a los eventos con los que fue entrenado genera errores de clasificación. Esto es porque el modelo es tan específico que no existe una generalización los conjuntos de datos que nos interesa clasificar.

La dimensionalidad de los datos es otro de los grandes problemas en reconocimiento de patrones, porque es muy difícil evaluar espacios de entrada de grandes dimensiones, debido a que existe demasiada información para ser tomada en cuenta al ejecutar el algoritmo. La dimensionalidad está determinada por el número de características que representan una instancia de los datos con los que se está trabajando. Para lidiar con este problema existe la opción de seleccionar solo los atributos relevantes, estos son los atributos que realmente ayudan a hacer la diferencia entre las diferentes clases; para resolver este problema se utilizan algoritmos de selección de atributos, manifolds y eigen vectores entre otros.

Estos problemas han sido atacados a lo largo de los años, sin embargo el proceso logrado ha sido relativamente poco, por lo que se necesita seguir trabajando para poder llegar a su solución.

2.1.3. Árboles de decisión

Los algoritmos de aprendizaje de árboles de decisión son uno de los métodos no paramétricos más populares utilizados en clasificación, este método de manera iterativa secciona el espacio de entrada hasta que cada región aproximadamente contiene una distribución de clases uniforme.

Los árboles de decisión están basados en la estructura de datos árboles donde cada nodo del árbol hace referencia a una frontera de decisión, la cual divide los datos en conjuntos y cada nodo hoja está asociado a una etiqueta de clase. La clasificación se realiza al seguir el camino de la raíz del árbol a la hoja correspondiente. Al elemento que se está clasificando se

le asocia la etiqueta de clase correspondiente al nodo hoja encontrado por el camino.

Un árbol de decisión es un clasificador que utiliza de forma iterativa la estrategia de divide y vencerás para formar una frontera de decisión sobre el espacio de entrada y generar un modelo de clasificación con dicha frontera.

Procediendo de arriba hacia abajo, la raíz del árbol se forma al seleccionar una variable [1] A la cual divide el conjunto de entrenamiento en subconjuntos mutuamente excusivos $S_0, S_1, \cdots, S_m$, donde cada S_i contiene todos los eventos que comparten el mismo valor para A. Comúnmente A es seleccionada a través de una medida de impureza (p.e., entropía, gini, Laplace, χ^2), la cual genera particiones paralelas a los ejes sobre el espacio de entrada. Existen otras técnicas menos comunes buscan combinaciones de atributos en cada nodo para producir particiones no paralelas a los ejes.

La misma metodología es aplicada de manera recursiva a cada subconjunto S_i para construir respectivamente cada uno los sub-árboles hijos. Un subconjunto S_i representa una hoja si la mayoria de sus ejemplos pertenecen a la misma clase o S_i es demasiado pequeño (e.c., si $|S_i| < \epsilon$,donde ϵ es definida por el usuario); la clase mayoritaria en S_i es asociada entonces con esa hoja del árbol.

Un ejemplo $\mathbf{x}$ es clasificado, iniciando desde la raíz del árbol, al seguir de manera iterativa la rama que concuerda con el valor de la variable del nodo. Al final del camino, la clase a la que pertenece la hoja es la clase asignada a $\mathbf{x}$.

Para construir un árbol de decisión debemos de tomar en cuenta la elección de las características a ser usadas en cada nodo interno además de la selección de la regla de decisión que va a ser aplicada a cada nodo. Esta elección es basada sobre todo en la calidad del corte que genera, esta medida está asociada a las funciones de impureza $\phi(p)$ lo cual nos da una idea de que tanto nos ayuda esta división en hacer el corte correcto para llegar a una clasificación correcta. Las funciones de impureza más populares son gini y entropía[7].

gini function $\phi(p) = \sum_j p_j (1 - p_j)$

entropía function $\phi(p) = -\sum_j j p_j \log p_j$

Donde p_j representa que tan bueno o que tan malo es el corte para el subconjunto j que lo está creando. El comportamiento del árbol disminuye conforme el valor de la función de impureza se eleva y viceversa, esto es, el comportamiento del árbol mejora conforme el resultado de la función de impureza disminuye.

El hecho de que cada uno de los nodos del árbol tenga una frontera de decisión nos ayuda

[1] Utilizamos letras mayúsculas para las variables (p.e., A) y letras minúsculas para los valores de las mismas (p.e., a).

a construir barreras de decisión bastante complejas, pero al mismo tiempo debemos de ser bastante cuidadosos al utilizar este poder porque podemos llegar a sobre entrenarlo. El sobre entrenamiento pasa cuando hacemos que el árbol clasifique de manera perfecta la base de entrenamiento pero cuando se intenta validar el árbol con la base de pruebas ocurren muchos errores de clasificación. Para resolver este problema existe la poda del modelo construido. La poda puede ser vista como cortar algunas ramas para hacer el modelo un árbol más generalizado y de esta manera hacerlo consistente con la base de verificación.

Construcción de los árboles de decisión

Existen varias maneras de construir un árbol de decisión, estas aproximaciones tienes algunas cosas en común pero son usados para crear diferentes tipos de árboles de acuerdo a su estructura la cual es definida por el método [12].

- De las hojas a la raíz
- De la raíz a las hojas
- Hibrido
- Crecer y podar

En la estrategia de las hojas a la raíz las clases de definen usando alguna métrica de distancia. La base de datos de entrenamiento se trata de ser clasificada en una forma de agrupamientos, donde elementos que están cercanos se agrupan juntos; es decir pertenecen a la misma clase la cual es muy diferente a la clase a la que pertenecen los elementos lejanos los cuales también son agrupados juntos para formar una clase diferente.

El esquema de la raíz a las hojas es el más común de todos debido a su forma de construcción, en cada nodo se realiza una división del conjunto de los datos en diferentes subconjuntos. Al definir el corte basándonos en la selección de la característica y una regla de división que actúa directamente sobre los elementos que pertenecen al subconjunto seleccionado para definir el corte. Cuando un subconjunto solamente tiene elementos que pertenecen a la misma clase o el número de elementos que pertenecen a la misma clase es pequeño comparado con aquellos que pertenecen a la misma clase se le puede dar una etiqueta al subconjunto, dicha etiqueta lo define como un nodo final o nodo hoja que representa a una clase determinada por la clase mayoritaria.

Un método híbrido es aquel que utiliza tanto la estrategia de las hojas a la raíz como la de de la raíz a las hojas para realizar la construcción del árbol, esto se puede dar de dos maneras: utilizar el esquema de la raíz a las hojas para definir los cortes iníciales del árbol y utilizar el método de las hojas a la raíz para generar las hojas del árbol o viceversa, es decir,

utilizar la estrategia de las hojas a la raíz para generar los cortes iníciales y el esquema de la raíz a las hojas para generas las hojas del árbol.

Al utilizar cualquiera de las formas de construcción anteriores podemos caer en el problema de sobre entrenamiento. El método de crecer y podar construye en una primera etapa un árbol que clasifica de manera perfecta el conjunto de datos de entrenamiento y eventualmente en una segunda etapa utiliza un conjunto de datos de validación para realizar la poda de las ramas del árbol. La idea principal es podar o cortar las ramas que sobre especifican el comportamiento del árbol y hacerlas más generales para hacer un modelo que pueda predecir el comportamiento de elementos aun no vistos. Al realizar este procedimiento nos aseguramos que el árbol resultante tendrá una mayor precisión al momento de realizar la clasificación[15].

2.1.4. El problema de la fragmentación

Los árboles de decisión construidos de la raíz a las hojas tienen una limitación, también conocida como el problema de la fragmentación, en el cual la continua partición de los datos de entrenamiento realizada en todos los nodos del árbol reduce el número de ejemplos o eventos en los nodos de más bajo nivel lo que genera una pérdida de soporte estadístico.

Una manera de ver el problema de la fragmentación es la siguiente. Si las señales que conforman las clases pueden ser vistas como una mezcla de componentes de un modelo probabilístico (p.e. una mezcla de gaussianas), cada intento del árbol de separar una componente del modelo de otra clase terminará produciendo fronteras de decisión que realizan el corte a través de componentes de otra señal. Demasiados cortes en el árbol terminan dividiendo los componentes de la señal en exceso, disminuyendo su soporte estadístico. Bajo ese escenario es común experimentar múltiples errores en la clasificación [13, 8, 9, 5, 6, 10, 4].

Un algoritmo de un árbol de decisión asume una representación donde cada clase es la disyunción de varios subconceptos, también conocida como forma normal disyuntiva o representación DNF Cada rama de la raíz del árbol a un nodo terminal o hoja representa un sub-concepto (o disyunción). Como ejemplo asumimos un espacio de variables compuesto por dos variables A_1 y A_2 y una distribución de eventos como se muestra en la Fig. 2.1-izquierda. Un posible árbol de decisión correspondiente a las particiones mostradas sobre el espacio de variables bidimensional mostrado en la Fig. 1-derecha.

En este ejemplo la señal de clase ($y = 1$) se divide en tres disyuntos, o áreas rectangulares, además la clase que representa la ausencia de señal ($y = 0$) contiene solamente un disyunto. Sobresaliendo que las distribuciones originales de los eventos de la señal se dividen en dos agrupaciones sin embargo el primer intento realizado por el árbol de decisión de aislar el agrupamiento de la esquina superior izquierda del agrupamiento que representa la ausencia de señal resulta en cortar el segundo agrupamiento (ubicado en la parte inferior izquierda) en dos partes.

Los algoritmos de decisión realizan una partición o refinamiento continuo sobre el espacio de variables; cada rama crece hasta que un nodo terminal u hoja define una región con una sola clase. Una limitación inherente en este método es que, mientras se busca por una región que muestra uniformidad en la clase, cada corte realizado en los datos de entrenamiento puede separar o alejar ejemplos en beneficio de diferentes disyuntos. Esta situación no solo requiere encontrar bastantes aproximaciones en las regiones dispersas del espacio de variables, sino que también reduce el soporte o la credibilidad en la evidencia de cada disyunto en forma individual, lo que complica eventualmente su identificación. Este problema es mejor conocido como el problema de la fragmentación. Como un ejemplo del mismo, en la Fig. 2-derecha se muestra una configuración de eventos similares a los de la Fig. 1 con la diferencia de que la división del agrupamiento ubicado en la parte inferior izquierda deja algunos eventos correspondientes a la señal en la región donde la ausencia de la misma resulta ser dominante. Fig. 2-derecha muestra el árbol correspondiente donde una porción fragmentada del agrupamiento inferior está siendo clasificada de manera incorrecta.

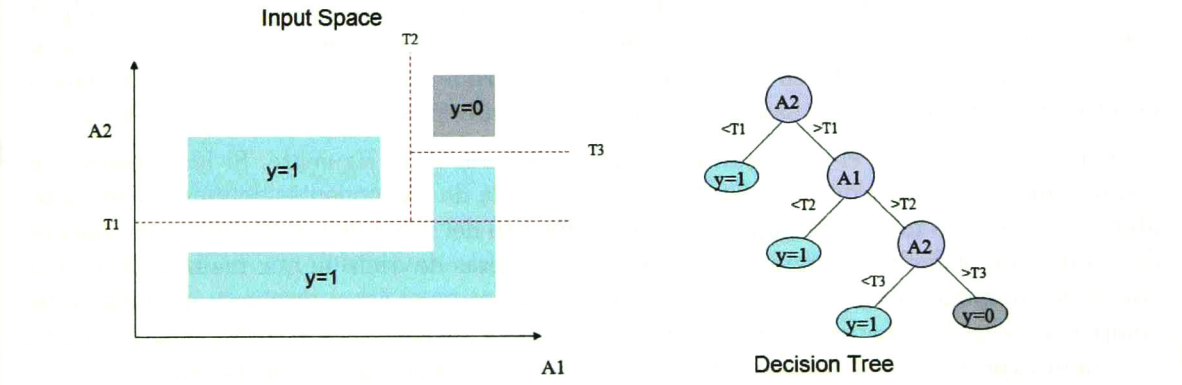

Figura 2.1: Izquierda. Un espacio bidimensional de variables con 2 clases (señal o positiva, $y = 1$, y la ausencia de la misma o negativa $y = 0$). Derecha. Árbol de decisión correspondiente.

El problema de la fragmentación se origina por dos razones principales. La primera es el requerimiento de que cada región cubierta por disyuntos diferentes (cada camino desde la raíz a las hojas) sea mutuamente excluyente de todos los demás disyuntos. Al excluir el traslape de estas regiones, forzamos que algunos eventos se separen de su agrupación más cercana. La segunda son las particiones burdas impuestas por el árbol sobre el espacio de variables; un árbol de decisión genera particiones paralelas a los ejes lo que disminuye la flexibilidad de tener polinomios de alto orden o inclusive particiones lineares sin límites en la inclinación del hiper-plano. Estas fronteras rígidas requieren demasiados pasos antes de que el algoritmo sea capaz de delimitar una región con clase uniforme; una consecuencia directa es la fragmentación indeseada de las agrupaciones de eventos pertenecientes a la misma clase.

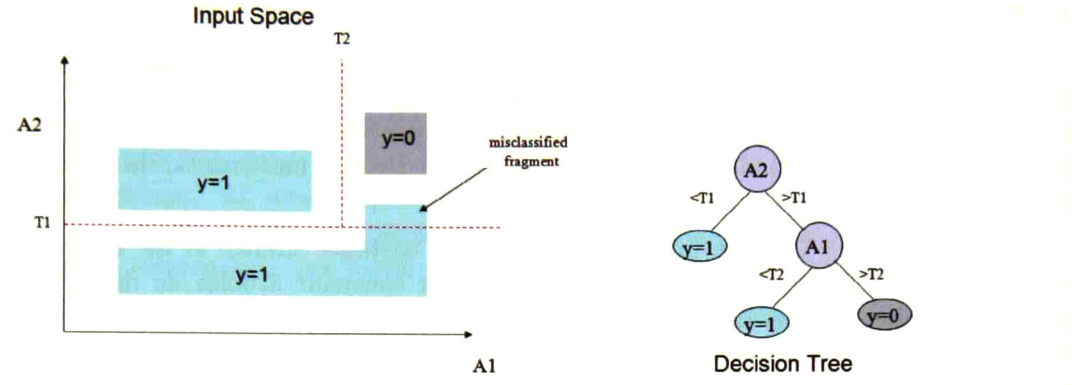

Figura 2.2: Izquierda. Un espacio bidimensional de variables con 2 clases (señal o positiva, $y = 1$, y la ausencia de la misma o negativa $y = 0$). Derecha. Árbol de decisión correspondiente con errores en la clasificación debido al problema de la fragmentación.

2.2. Estado del arte

Se presentan las principales propuestas que abordan el problema de la fragmentación, dichas propuestas buscan reducir la fragmentación en los arboles de decisión, la mayoría opta por utilizar múltiples variables en cada nodo. Sin embargo existen algunas propuestas que resuelven este problema generando modelos alternos a los de los árboles de decisión que llegar a ser muy interesantes.

2.2.1. Principales propuestas que abordan el problema de la fragmentación.

Este problema ha sido atacado por múltiples investigadores pero ninguno de ellos ha sido capaz de resolverlo del todo. Lo cual soporta el trabajo presentado por Hand en el cual critica la falsa Ilusión de progreso en la literatura de clasificación de patrones debido a pequeños incrementos en la exactitud entre los modelos pasados y los nuevos modelos propuestos[5], En los siguientes párrafos se presentan los trabajos más prometedores que han tratado de resolver el problema de la Fragmentación.

Li y Wong propusieron los patrones emergentes. Estos patrones se encontraban mediante reglas globales que afectaban a todo el conjunto de datos logrando reunir a la mayoría de los elementos de una misma clase pero sin cubrir algún elemento de algunas de las clases diferentes. Esto Les permite generar barreras de decisión que separan de manera casi perfecta

las diferentes clases entre si[8].

Vilalta, Blix y Rendell presentan el análisis de datos global, esta técnica toma en consideración todos los datos de entrenamiento para generar árboles de decisión más confiables, además utilizan la combinación de atributos para superar el problema de la fragmentación al disminuir la cantidad de particiones y al mismo tiempo incrementar la exactitud del modelo[13].

Por su parte Ho y Scoot presentan un trabajo bastante similar al de Vilalta, Blix y Rendell donde comparan 8 diferentes métodos para construir árboles de decisión. En su trabajo describen formas de realizar la agrupación de pares de atributos con la intención de reducir el tamaño del árbol y por consecuente la fragmentación. Al final remarcan la importancia de tomar en cuenta los elementos de forma global en cada nodo para mejorar el comportamiento del árbol[6].

Liu y Sctiono desarrollaron los árboles de decisión multi variables, los cuales al igual que los dos métodos anteriores trata de resolver el problema al realizar la combinación de característica a través de la mejor transformación de características T y después realizar la búsqueda de la mejor división sobre la nueva característica multi variable provista por T, este método también es acompañado con la poda de árboles. Sin embargo a pesar de todo se sigue presentando la fragmentación en esta nueva característica[10].

Siguiendo la misma ruta de investigación posteriormente Liu, Hu and Hsu presentan una solución bastante interesante al problema debido a que ellos no modifican el algoritmo del árbol de decisión. Presentan un algoritmo de post procesamiento en el cual toman como entrada la salida del algoritmo del árbol de decisión. Este algoritmo realiza la búsqueda de nodos cercanos que comparten la misma etiqueta de clase y tienen reglas bastantes similares. Al compartir similitudes en las reglas son unificados en un nodo general con una regla unificada que abarca las reglas particulares de cada uno de los nodos originales, si esta regla general abarca a otros nodos originales que tienen una etiqueta de clase diferente a la del nodo general se generan nodos de excepción para derivar estas excepciones de la regla general[9]. Al final del algoritmo se genera un nuevo árbol con nodos generales y de excepción como los nodos hojas. Esta es una de las técnicas mas prometedoras para acabar con el problema de la fragmentación debido a que los nodos generales y de excepción se adaptan de una manera más natural a la forma de los agrupamientos del los datos, y brindan bordes de decisión más flexibles que los árboles tradicionales.

Por otra parte DeLisle y Dixon proponen una nueva solución al proceso de inducción de los árboles de decisión al utilizar programación evolutiva. Ellos tratan a los árboles de Decisión como un problema más de optimización y lo resuelven mediante el uso de algoritmos genéticos. Las operaciones genéticas de selección, mutación y cruza son aplicadas directamente a las estructuras de los árboles de decisión. Como resultado logran un incremento del 5 al 10 por ciento de mejoría en el numero de clasificaciones correctas en comparación con los árboles

con los cuales se inicia el proceso evolutivo; también reportan haber obtenido una reducción en la complejidad del árbol resultante lo cual puede ser traducido en una disminución en la fragmentación[4].

Una de las características comunes de estos trabajos es que presentan soluciones que pretenden resolver el problema de la fragmentación sin embargo ninguna de estas propuestas resuelve el problema. En ningún método se propone una medida de fragmentación o algo que busque entender cómo es que la fragmentación genera errores de clasificación. Por lo tanto nosotros proponemos una forma de medir la fragmentación ejercida sobre un espacio de entrada debido al proceso de inducción propio de los árboles de decisión.

Capítulo 3

Propuesta

Se presenta una nueva forma para medir la fragmentación ejercida por un modelo de árboles de decisión. Para llevar a cabo esta métrica es necesario tener el modelo del árbol y los datos con los cuales fue generado. Con esta información se realiza un análisis, de como el modelo del árbol divide el espacio de entrada, generando así los fragmentos causantes del problema. Este análisis es detallado paso a paso, en el se brinda información sobre el grado de fragmentación existente en todos los niveles del árbol, es decir desde la distribución de los elementos en los fragmentos hasta una medición de la fragmentación total en todo el árbol de decisión.

3.1. Evaluación de la fragmentación en el aprendizaje de árboles de decisión

Actualmente la forma de medir la fragmentación tiene algunas deficiencias, las cuales surgen a partir de que se miden características propias del modelo, sin tomar en cuenta la estructura interna de los datos. Es decir solo se toma en cuenta el numero de nodos que contiene el árbol de decisión, en lugar de medir la fragmentación en si. Esto genera que la medición obtenida sea errónea y de por tal manera los esfuerzos por reducir la fragmentación no sean tan adecuados.

Ante esta problemática proponemos una forma de medir la fragmentación ejercida por el modelo de un árbol de decisión sobre el espacio de entrada utilizado en su construcción. Con la intención de no solo obtener una métrica de la fragmentación, sino también obtener mas información respecto a como se encuentran fragmentados los datos, con la finalidad de detectar las partes del modelo que necesitan una mejora para obtener una mayor precisión en la clasificación.

La idea principal es descomponer cada clase en sus agrupaciones inherentes, donde cada agrupación representa un sub-concepto el cual puede sufrir múltiples particiones en las diferentes hojas del árbol. Cada agrupación es analizada de forma independiente y como resultado produce una serie de estadísticas que evalúan cuantitativamente el grado de fragmentación ejercido por el árbol en cada agrupación de cada una de las clases.

3.1.1. Notacion preliminar

En el capitulo dos se presento la notación general utilizada en la literatura de reconocimiento de patrones para representar a los arboles de decisión. En esta sección volveremos a repasar los conceptos básicos, generalizando de manera ligera la notación previamente vista:

Definimos el espacio de entrada $\mathcal{X}$ de la misma manera que lo hicimos previamente, y el espacio de salida $\mathcal{Y} = \{y_1, y_2, \cdots, y_k\}$ compuesto por k clases diferentes.

Cada clase y_j sera descompuesta en agrupaciones $\{c_j^i\}$, donde n_j es el número de agrupaciones en y_j.

3.1.2. Medición de fragmentos

Para poder realizar la medición de la fragmentación generamos un evaluador de fragmentos, el cual es un sistema que toma como entrada el modelo de un árbol de decisión y la base de datos que fue utilizada para generar dicho modelo, es decir el conjunto de entrenamiento. Estos datos deben de ser procesados, para cumplir nuestro objetivo:

1. Dividir los datos en conjuntos de elementos con la misma etiqueta de clase.
2. Ejecutar un algoritmo de Clustering sobre los datos que cuentan con la misma etiqueta de clase.
3. Separar las agrupaciones obtenidas por el método de Clustering.

Clustering es un método que se encarga de examinar los elementos que conforman la base de datos que estamos examinando para detectar las agrupaciones de elementos que se encuentran de manera interna en la base de datos. Estas agrupaciones son las que nos sirven a nosotros para detectar la fragmentación en los arboles de decisión. Debido a que un árbol de decisión ideal es aquel que tiene la capacidad de modelar de manera perfecta cada una de las agrupaciones de los datos de entrenamiento en un nodo final del árbol. La fragmentación se da cuando una agrupación esta asociada a dos o mas nodos finales del modelo del árbol de decisión.

El evaluador de fragmentos genera como salida una serie de estadísticos donde se muestran los fragmentos y sus porcentajes en las cuales son fragmentadas las agrupaciones de los datos de entrenamiento. Después con estos porcentajes se calcula el grado de fragmentación de cada agrupación. Con estas cifras se puede ahora medir la fragmentación de cada una de las clases. Y por ultimo se calcula el total de la fragmentación existente en el árbol. Dicho sistema se divide en 4 partes:

1. Extraer las reglas del árbol.
2. Determinar las particiones sobre el espacio de entrada.
3. Obtener los fragmentos de las agrupaciones.
4. Evaluar los fragmentos.

Extraer las reglas del árbol

Para extraer las reglas del árbol se toma como entrada el modelo del árbol de decisión generado por un algoritmo de aprendizaje de árboles de decisión. El modelo es recorrido desde la raíz hasta cada una de las hojas para definir las reglas que genera dicho modelo en sus caminos desde la raíz hasta las hojas. El recorrido inicia en la raíz y genera el inicio de dos reglas. Conforme se va recorriendo el arbol, cada nuevo nodo de decisión genera dos nuevas reglas. Dichas Reglas se forman con la regla que se había generado por su nodo padre mas cada una de las restricciones adicionales que genera el nuevo nodo, cuando finalmente se llega a un nodo hoja se deja de crecer la búsqueda en el nodo y se anexa la regla generada a la lista de reglas del árbol. Estas reglas nos ayudan a definir las particiones a realizar en las agrupaciones y de esta manera es posible encontrar los fragmentos que existen en el modelo.

Determinar las particiones sobre el espacio de entrada

Con las reglas que genera cada nodo hoja, podemos generar las fronteras de decisión existentes en cada hoja del árbol. Con dichas fronteras uno puede descomponer el espacio de entrada en hiper-rectángulos. Estos hiper-rectángulos definen donde se realizan las divisiones, con esta información podemos dividir las agrupaciones existentes en el espacio de entrada y de esta manera podemos encontrar los fragmentos específicos del modelo.

Para realizar dicho proceso es necesario tomar cada una de las agrupaciones y verificar cuales de los elementos que la conforman pertenecen a cada una de las reglas. Para agilizar este proceso se puede interrumpir la verificación de la agrupación una vez que todos los elementos que la conforman se encuentren asignados a alguna regla del árbol.

Obtener los fragmentos de las agrupaciones

Las agrupaciones de los datos de entrenamiento también se toman como nuestros datos de entrada, con dichas agrupaciones y los cortes obtenidos en el paso anterior podemos dividir cada una de las agrupaciones en el número de fragmentos generados por el modelo de entrada. Los fragmentos son conformados por cada uno de los elementos pertenecientes a una agrupación que caen en la misma regla. Cuando los separamos de la agrupación estos forman un Fragmento. Una vez que las agrupaciones son fragmentadas podemos definir el porcentaje perteneciente a cada fragmento. Es decir del total del numero de elementos de la agrupación cual es el porcentaje correspondiente a los elementos que conforman el fragmento. Esta es la información mas importante, porque gracias a ella podemos generar estadísticas que nos ayuden a determinar que tan bueno o que tan malo es el modelo de clasificación generado por el algoritmo de árboles de decisión.

Evaluar los fragmentos

Con los fragmentos generados en el paso anterior, se obtienen los porcentajes de elementos correspondientes a la agrupación en la que están dichos fragmentos. Con esta información se generan la medida de fragmentación. Primero se calcula la cantidad de fragmentación existente en cada agrupación al evaluar que tan importante es el porcentaje de concentración de los elementos contenidos en cada uno de los fragmentos que la componen. Posteriormente calculamos la fragmentación existente en cada clase al promediar la medida de la fragmentación existente en cada una de las agrupaciones que conforma la clase. Como ultimo paso se promedia la medida de fragmentación de cada clase, y de esta manera se obtiene la fragmentación contenida en todo el árbol.

3.1.3. Arquitectura

La Fig. 3.1 muestra una representación esquemática de nuestro sistema. Como primer paso tomamos el árbol de decisión como entrada y extraemos los disyuntos o las reglas de los caminos desde el nodo raíz hasta cada uno de los nodos hojas. Los disyuntos son agrupados en su clase correspondiente. Cada disyunto divide el espacio de variables en un hiper-cubo donde la meta es llegar a tener dicha región con una distribución uniforme de la misma clase. En este punto somos capaces de determinar la posición de cada disyunto o regla del árbol sobre el espacio de variables; lo cual nos brinda una clara idea de donde están siendo impuestos los cortes por el árbol de decisión.

El segundo paso toma como entrada la descomposición de cada clase en agrupaciones. Para esto aplicamos un algoritmo de agrupamiento sobre todos los eventos que conforman una misma clase utilizando el algoritmo de agrupamiento espectral [3]; este algoritmo ha

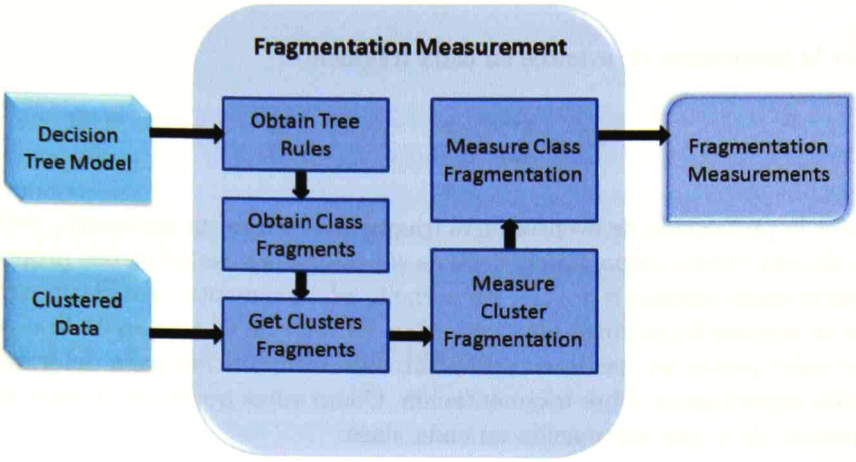

Figura 3.1: Arquitectura de la evaluación de la fragmentación en los árboles de decisión.

mostrado un excelente comportamiento en comparación con otras técnicas de agrupamiento (p.e., EM or K-Means) y debido a que utiliza la técnica de Sprectrum tiene la importantísima propiedad de ser capaz de reducir la dimensionalidad de los datos y por lo tanto tener un mejor comportamiento con datos que cuentan con una dimensionalidad grande.

Una vez que se identifican los agrupamientos, determinamos como es que cada uno de estos agrupamientos fue fragmentado. En esencia traslapamos los agrupamientos con cada una de las regiones definidas por cada disyunto del árbol de decisión. Nuestra meta es capturar como los eventos de cada agrupamiento se fragmentan al caer en las diferentes regiones de los nodos hojas y generan los fragmentos inducidos por el árbol de decisión.

En el siguiente paso extraemos las características relacionadas con el grado de fragmentación de cada agrupación con base en la distribución de eventos en sus fragmentos. Comparamos dichas agrupaciones *vs* híper rectángulos para obtener el numero de fragmentos que comprende cada agrupación. Para cada agrupación c_j^i obtenemos el numero de fragmentos en los cuales se divide el agrupamiento. Después promediamos para obtener un estimado del número de fragmentos por agrupación en cada clase:

$$\Phi_j = \frac{1}{n_j} \sum_i \phi_j^i \tag{3.1}$$

Este estadístico mostrado anteriormente provee información de en promedio que tanto se encuentra dividido un grupo, pero no habla nada acerca de la proporción de eventos existentes en cada fragmento. Esto es importante debido a que la presencia de pequeños fragmentos incrementa la probabilidad de errores en la clasificación. Introducimos una métrica para cada agrupamiento, ψ_j^i, la cual da mayor peso a pequeños fragmentos, la medida de los fragmentos

está dada por la proporción de eventos en cada fragmento:

$$\psi_j^i = \sum_l e^{-\frac{\alpha_l}{\sigma}} \qquad 1 \leq l \leq \phi_j^i \tag{3.2}$$

Donde α_l es la proporción de eventos en el fragmento l y σ es un parámetro definido por el usuario para decidir cuanta importancia debe de ser asignada a las pequeñas proporciones (en nuestros experimentos usamos $\sigma = 0{,}1$). En esencia, ψ_j^i incrementa cuando un agrupamiento está dividido en muchos fragmentos muy pequeños. El valor de ψ_j^i esta acotado superiormente por ϕ_j^i; dicho valor puede ser usado con propósitos de comparación para determinar cuándo una agrupación experimenta sobre fragmentación. Como antes podemos obtener un estimado del valor esperado de ψ por agrupación en cada clase:

$$\Psi_j = \frac{1}{n_j} \sum_i \psi_j^i \tag{3.3}$$

Estas estadísticas forman parte de la salida de la evaluación de la fragmentación en los árboles de decisión. Dicha salida es un instrumento útil en evaluar la calidad de un árbol de decisión como una función del grado de fragmentación impuesto sobre todas agrupaciones de las clases. La estrategia de realizar la evaluación de lo particular a lo general nos permite evaluar el grado de fragmentación primero a nivel agrupación y después al nivel de clase.

Al utilizar esta medida, podemos comprender de mejor manera el comportamiento del algoritmo de los árboles de decisión. Podemos observar como es que se realizar la fragmentación etapa por etapa. Al tener un conocimiento mas profundo del tema podemos idear mejores técnicas para combatir la fragmentación y generar algoritmos que mejoren la precisión en la clasificación e incrementar el uso del los árboles de decisión en aplicaciones del mundo real por su precisión y facilidad de interpretación del modelo generado.

Capítulo 4

Trabajo Experimental

En este capitulo se presenta una breve descripción sobre el ambiente en el cual fueron desarrollados los experimentos. Se describen los objetivos que se buscan satisfacer mediante estos experimentos. es decir, demostrar que la métrica propuesta mide efectivamente la fragmentación y hacer una comparación de la métrica propuesta con la métrica actual. Al final se presentan los resultados de dichos experimentos.

4.1. Experimentos

Los experimentos que se presentan a continuación buscan demostrar que la medida propuesta en el capitulo anterior efectivamente mide la fragmentación y posteriormente se realizara una comparación con el método clásico que se utiliza actualmente en la literatura de reconocimiento de patrones. Para establecer una comparación entre ambas métricas y determinar la utilidad de la métrica presentada en esta tesis.

Para realizar los experimentos es necesario tener un algoritmo de aprendizaje de árboles de decisión con el cual generar los modelos a los cuales se les va a evaluar la fragmentación. Para este fin utilizaremos el software WEKA Waikato Environment for Knowledge Analisys[14].

Este software contiene múltiples algoritmos de clasificación, entre ellos se encuentran distintas variantes de los árboles de decisión como:

- J48 una implementación del algoritmo C4.5
- Best-First Decision Tree
- RandomTree
- Naive Bayes Tree

- Functional Trees
- REPTree una implementación de Fast Decision Tree Learner
- Logistic Model Trees

Escogeremos el algoritmo de BestFirst debido a que, es uno de los que nos permite realizar arboles decisión con las métricas de gini e information gain. Estos nos ayudara a generar arboles diferentes con la misma base de datos y de esta forma tener una comparación acerca de cuál de los dos modelos es aquel que ejerce mas fragmentación sobre los mismos datos.

Las agrupaciones las realizaremos con el algoritmo de spectral clustering[3]. Este algoritmo realiza el agrupamiento basándose en el concepto de spectrum, lo cual le permite disminuir la dimensionalidad de los datos y realizar un agrupamiento más cercano al real. Esta es la característica principal por la cual se escoge este algoritmo de agrupamiento.

Para generar el agrupamiento utilizaremos la implementación de spectral clustering integrada en el paquete "The Spider" Dicho paquete es una librería de objetos de Matlab el cual ayuda a realizar de manera razonable operaciones de reconocimiento de patrones sobre grandes bases de datos.

Las salidas de ambos algoritmos serán las entradas de nuestros experimentos. Utilizaremos la información proporcionada por ambos algoritmos para determinar los fragmentos existentes en el modelo generado por el algoritmo Best-First Decision Tree sobre las agrupaciones determinadas por el spectral clustering.

Nuestros experimentos se harán sobre la búsqueda de la partícula subatómica single top quark, el cual es un problema desafiante en la física de partículas debido a la gran cantidad de partículas subatómicas diferentes que existen con las mismas características[2, 1].

En esencia, las señales digitales producidas por un detector de partículas son guardadas después de cada colisión para identificar ciertas partículas, y de esta manera poder generar teorías mas acertadas acerca de como se comportan las partículas. Cada colisión o evento es caracterizado por variables relevantes para separar la señal deseada; en este caso single top quark, de otras señales muy similares. Las variables se dividen en 3 grandes grupos:

- Cinemática del objeto individual
- Cinemática del evento global
- Correlaciones angulares

Ejemplos de estas variables incluyen momentum transversal, masa invariante de todos los objetos, separación angular entre los dos jets lideres, entre otras. El problema es inherentemente desafiante debido a que la partícula single top quark es cinemáticamente y

topológicamente muy similar a otros eventos como los eventos W+jets y $t\bar{t}$ Nuestros datos fueron proporcionados por el Dr. Gordon Watts afiliado al departamento de física de la Universidad de Washington, estos datos son generados a partir de un simulador de eventos tipo monte carlo basado en el detector DØ existente en Fermilab.

La razón por la cual se escogió esta base de datos como nuestro caso de estudio es debido a la naturaleza propia de los datos. Las similitudes encontradas entre las diferentes partículas hace que sean muy difícil diferenciar las partículas entre si. Lo cual genera una gran cantidad de errores en la clasificación. Estos errores al ser traducidos a un árbol de decisión, se dan por efecto de la fragmentación. La gran similitud de las partículas genera traslapes de las agrupaciones de las distintas clases, este fenómeno además de generar errores en la clasificación es determinante para asegurar la existencia de fragmentación en un árbol de decisión. Esta ultima característica es la que hace de la base de datos del single top quark una base de datos ideal para verificar nuestra aportación.

La base de datos del single top quark contiene una gran cantidad de elementos y las clases no cuentan con el mismo número de elementos por lo que se tomaron diez diferentes muestras aleatorias del 0.5 % del total de elementos que conforman cada clase. Las muestras se realizaron de esta manera con la finalidad de mantener la distribución inicial de los datos. Es decir darle más importancia a las clases que cuentan con un mayor número de elementos, y por lo tanto una menor importancia a las que cuentan con una menor cantidad de elementos.

Para cada una de las diez bases de datos se realizo un modelo de clasificación con el algoritmo de Best-First Decision Tree utilizando como medida de impureza gini y posteriormente se realizo un segundo modelo de clasificación con el mismo algoritmo pero utilizando information gain como medida de impureza. Estos dos modelos generan arboles de decisión distintos y por lo tanto los fragmentos generados en los datos también son diferentes.

Para poder realizar el proceso de clustering o agrupamiento, cada una de las bases de datos es descompuesta en diferentes bases más pequeñas que únicamente contienen elementos pertenecientes a la misma clase. Estas nuevas bases de datos son la entrada para el algoritmo de agrupamiento. Se va a realizar el agrupamiento de todas las clases que conforman el espacio de entrada de manera individual para obtener la cantidad de agrupamientos existentes en cada clase. El agrupamiento se realiza con el algoritmo de Spectral Clustering o Agrupamiento Spectral, utilizando la versión de este algoritmo que esta en la librería "The Spider" de MatLab.

Una vez obtenidos ambos modelos de clasificación y las agrupaciones contenidas en cada clase, esta información se utiliza para obtener las reglas del árbol. Con dichas reglas se obtienen los hiper-rectángulos que dividen el espacio de entrada en el modelo. Y a partir de los hiper-rectángulos se obtienen los fragmentos en los cuales se dividen cada uno de los agrupamientos que conforman las distintas clases de una base de datos.

Cada fragmento contiene cierto número de elementos pertenecientes a una agrupación,

con este numero de elementos se obtiene el porcentaje del total de elementos que conforman al agrupamiento. Cuando se obtiene este porcentaje para todos y cada uno de los fragmentos existentes en todas las agrupaciones de todas las clases; se utilizan los porcentajes para medir la fragmentación ejercida a nivel agrupación al utilizar la ecuación 3.2. Para realizar estos experimentos se utiliza un valor de 0.01 en σ en la ecuación 3.2.

De la misma forma al tener la medida de fragmentación correspondiente a cada agrupación, podemos calcular la fragmentación existente en cada clase mediante la ecuación 3.3. Al tener la medida de fragmentación correspondiente a cada clase se termina el proceso y se presentan las estadísticas generadas al utilizar esta métrica de fragmentación.

4.2. Resultados

En una primera instancia se presentan los resultados que la métrica propuesta lanza con el objeto de demostrar que efectivamente mide la fragmentación existente en un modelo de árboles de decisión. Primero se presentan los valores iniciales que se necesitan para realizar este proceso, es decir la obtención de los porcentajes del total de elementos pertenecientes a la agrupación que contiene cada fragmento. En un segundo termino se presenta la métrica de la fragmentación a nivel agrupación es decir los valores de Φ_j y Ψ_j. Posteriormente se presentan los valores de Φ_j y Ψ_j en cada clase y el total de la fragmentación existente en el árbol. Finalmente se muestran los valores de Φ_j y Ψ_j que toman cada una de las clases y el árbol sobre el promedio de las 10 bases de datos que se utilizaron para realizar dichos experimentos.

A continuación se muestran los resultados de los experimentos. En la Tabla 4.1 se muestra un ejemplo de la distribución de los elementos sobre los fragmentos en los cuales se dividen las diferentes agrupaciones. En la primera columna se muestra el número de fragmento, mientras que de la columna 2 a la 7 se muestran los porcentajes del total de elementos en la agrupación que contiene dicho fragmento. Aquí se puede observan como cada fragmento contiene porciones diferentes de la agrupación en su interior.

Fragmento	$Cluster_0$	$Cluster_1$	$Cluster_2$	$Cluster_3$	$Cluster_4$	$Cluster_5$	$Cluster_6$
1	0.86	0.87	0.9	0.91	0.86	0.93	0.89
2	0.09	0.01	0.06	0.08	0.12	0.06	0.07
3	0.05	0.12	0.04	0.01	0.02	0.01	0.04

Tabla 4.1: Distribución de los elementos en los fragmentos en los cuales se dividen las diferentes agrupaciones.

Una vez que tenemos estas proporciones esta información es utilizada para obtener los valores que toman tanto Φ_j y Ψ_j para cada agrupación. Como se puede observar en la Tabla 4.2

se muestra la Fragmentación ejercida sobre un conjunto de agrupaciones que pertenecen a la misma clase. En la primera columna se muestran los agrupamientos, mientras que en la segunda y tercera columna se muestran los valores de Φ_j y Ψ_j correspondientes a cada agrupación.

Cluster	Φ_j	Ψ_j
0	1.01	3
1	1.21	3
2	1.22	3
3	1.35	3
4	1.12	3
5	1.45	3
6	1.17	3

Tabla 4.2: Valores de Φ_j y Ψ_j correspondientes a la fragmentación ejercida en cada una de las agrupaciones

Por último se obtiene la fragmentación a nivel clase. En la Tabla 4.3 se puede observar la fragmentación ejercida por el árbol de decisión sobre cada una de las clases que determina. En la primer columna se muestra la clase a la que pertenecen los valores de Φ_j y Ψ_j. En las columnas dos y tres se pueden observar los valores de Φ_j y Ψ_j que toman dichas clases.

Clase	Φ_j	Ψ_j
0td	9.52	13.15
0tl	17.91	21.67
0wb	13.47	16.9
0wc	4.41	7.17
0wl	0.07	1.33
1s	1.35	2.86
1t	0.07	2
Tree	6.69	9.3

Tabla 4.3: Valores de Φ_j y Ψ_j correspondientes para cada una de las clases de un árbol

La Tabla 4.4 muestra el promedio y la desviación estándar de la fragmentación ejercida a nivel clase en las diez bases de datos que conforman nuestro conjunto de pruebas. En la primera columna se muestran las clases (los valores 0wb, 0wc y 0wl representan la partícula W+jets, los valores 0td y 0tl representa la señal $t\bar{t}$, mientras que los valores 1s y 1t representan a la partícula single top quark.) La segunda y la tercera columna muestran los valores de Φ_j y Ψ_j cuando se utilizo la métrica gini en la construcción de los árboles de decisión; la cuarta y quinta columna son equivalentes a la segunda y tercera columna pero para árboles que utilizan information gain como su medida de impureza.

Clase	Φ_j (Gini)	Ψ_j (Gini)	Φ_j (Info. Gain)	Ψ_j (Info. Gain)
0td	21.4 (25.50)	11.12 (6.28)	26.4 (11.53)	17.61 (4.13)
0tl	20.85 (12.18)	15.04 (7.99)	25.09 (12.53)	17.86 (6.87)
0wb	16.89 (2.96)	11.7 (1.80)	25.93 (13.50)	14.93 (4.86)
0wc	5.61 (2.93)	2.52 (0.83)	7.54 (0.89)	3.33 (0.15)
0wl	1.47 (0.03)	0.03 (0.00)	1.32 (0.00)	0.16 (0.00)
1s	0.98 (0.02)	2.79 (0.04)	3.46 (0.11)	1.3 (0.04)
1t	2 (0.00)	0.07 (0.00)	2 (0.00)	0.47 (0.00)

Tabla 4.4: Valores estimados para Φ_j y Ψ_j.Los valores mostrados son para árboles de decisión que utilizan gini e information gain como métricas de impureza. Entre paréntesis se muestran los valores que toma la desviación estándar.

En esta segunda parte se muestra un comparativo de la métrica clásica de fragmentación *vs* la métrica propuesta en esta tesis. El comparativo busca definir que tan buena o mala es la métrica propuesta en este trabajo. Para comparar estas métricas utilizaremos la precisión de los modelos generados para clasificar elementos como nuestra base, debido a que cada nuevo algoritmo busca incrementar esta precisión.

Para poder realizar la comparación entre la métrica tradicional y la métrica propuesta presentamos las Tablas 4.5 y 4.6 donde en las columnas presentamos la medida de fragmentación tradicional, seguido de los valores de la métrica que estamos proponiendo Φ_j y Ψ_j y finalmente la precisión en la clasificación que presenta dicho modelos del árbol de decisión. Esta información se muestra para cada una de las diez bases de datos generadas sobre las cuales se realizaron los experimentos, las cuales representan cada una de las filas de la tabla, además de una fila adicional que representa el promedio de las diez bases. Cabe destacar que la métrica tradicional toma en consideración únicamente una medida a nivel árbol de decisión y no una medición especifica para cada una de las clases sobre las cuales se aplica el modelo.

En la Tabla 4.5 se muestran las medidas de fragmentación que toma el modelo del árbol de decisión cuando se utiliza la función de impureza gini y en la Tabla 4.6 se muestran los valores cuando la función de impureza information gain. Con estos dos diferentes tipos de árboles se puede verificar si efectivamente la función de impureza que cuenta con una menor cantidad de fragmentación es aquella que tiene una mayor precisión en la clasificación. Además de verificar que la métrica propuesta se comporte de manera similar a la actual o presente un mejor comportamiento.

Base de datos	Métrica tradicional	Φ_j	Ψ_j	Precisión
1	107	6.69	9.3	81.7
2	79	7.03	11.79	81.27
3	52	4.62	6.81	80.68
4	106	7.06	10.09	80.51
5	118	6.76	10.19	79.32
6	82	7.68	11.26	80.31
7	69	8.35	12.23	80.39
8	94	7.4	10.37	80.12
9	88	6.56	10.08	79.93
10	82	7.86	10.96	80.96
Promedio	87.7	7	10.31	80.52 (0.46)

Tabla 4.5: Comparación de la métrica tradicional *vs* la métrica propuesta en esta tesis. Los modelos que se están comparado fueron realizados utilizando la función de impureza gini.

Base de datos	Métrica tradicional	Φ_j	Ψ_j	Precisión
1	121	7.66	10.61	80.66
2	110	10.67	14.47	80.5
3	147	9.53	13.19	81.35
4	127	10.31	13.97	79.82
5	152	8.89	12.33	79.99
6	92	8.85	12.45	80.03
7	131	10.05	13.9	79.04
8	98	10.18	13.9	79.62
9	83	9.66	13.45	79.59
10	115	9.52	12.81	80.78
Promedio	117.6	9.52	13.11	80.14 (0.46)

Tabla 4.6: Comparación de la métrica tradicional *vs* la métrica propuesta en esta tesis. Los modelos que se están comparado fueron realizados utilizando la función de impureza information gain.

Capítulo 5

Conclusiones y Trabajo Futuro

Finalmente se presentan las conclusiones a las cuales se llegaron mediante el trabajo experimental. La mas importante de ellas es que se demostró que la métrica propuesta efectivamente mide la fragmentación, además de que brinda una mayor información acerca del estado de la fragmentación en el modelo a diferencia de la métrica tradicional la cual no aporta ninguna información extra. Además se presentan varias maneras de solucionar el problema de la fragmentación mediante el uso de esta métrica.

5.1. Conclusiones

Como se puede observar en las Tablas 4.1, 4.2, 4.3 y 4.4 podemos concluir que efectivamente la métrica que nosotros presentamos cumple con el primer objetivo que nos planteamos en los experimentos, es decir, demostrar que efectivamente nuestra propuesta es capaz de medir la fragmentación.

Por otro lado en las Tablas 4.5 y 4.6 vemos que la información que proveen ambas métricas concuerda. Es decir cuando se generan los árboles de decisión utilizando gini como la función de impureza ambas métricas generan valores bajos de fragmentación, para la métrica tradicional se obtiene en promedio un valor de 87.7 mientras que Φ y Ψ tienen un valor de 7 y 10.31 respectivamente. Por el otro lado cuando los modelos son generados utilizando como función de impureza information gain se obtienen valores altos de fragmentación, para la métrica tradicional se obtiene en promedio un valor de 117.6 y Φ y Ψ toman los valores de 9.52 y 13.11 respectivamente.

Algo que cabe destacar es que gini cuenta con una mayor precisión en la clasificación (80.52) en comparación con information gain (80.14), por lo que podemos concluir que la métrica propuesta de fragmentación esta asociada con el desempeño del algoritmo, mientras esta métrica obtenga un menor grado de fragmentación en el modelo, este tendrá una mayor

precisión, en comparación con otros modelos que cuenten con mayor fragmentación.

Además de ser congruente con la métrica tradicional, la métrica propuesta provee una mayor cantidad de información que puede ser utilizada para analizar el comportamiento de los arboles de decisión y desarrollar mecanismos basados en esta nueva métrica para solucionar de una manera mas efectiva el problema de la fragmentación, el cual tiene un alto impacto en el desempeño de los árboles de decisión.

A continuación presentamos otras conclusiones a las que se llegaron al observar de manera detenida la información proveida por las Tablas 4.1, 4.2, 4.3, 4.4, 4.5 y 4.6. La información contenida en la Tabla 4.1 es de vital importancia debido a que es en este punto donde se pueden identificar los fragmentos que son causantes de los errores en la clasificación. El localizar estos puntos nos puede ayudar a realizar modificaciones en el modelo del árbol, para incrementar la precisión del modelo al reducir la fragmentación existente en el mismo.

Por su parte la información que nos brinda la Tabla 4.2 nos da una imagen general acerca de como se encuentra distribuida la fragmentación sobre todas las agrupaciones de la clase, de esta forma es fácil determinar cual de las agrupaciones que conforman la clase es la que se encuentra mas fragmentada y por lo tanto sobre la que nos tenemos que enfocar en disminuir su fragmentación.

Mientras que la Tabla 4.3 Nos brinda información de manera general acerca de la fragmentación existente en cada una de las clases que conforman a árbol de decisión. Esta información puede ser utilizada para tomar decisiones estratégicas acerca de cuales son las clases que están mas fragmentadas y en base a esta información y nuestras clases mas importantes podemos determinar cual de las clases es en la que nos debemos de concentrar para reducir su fragmentación y por ende incrementar la precisión de la clasificación.

La medida de fragmentación propuesta es útil para determinar el grado de fragmentación que se ejerce sobre los datos con los cuales se construyo el árbol. A partir de la tabla 4.4 se pueden obtener conclusiones acerca de la calidad de los árboles de decisión cuando estos son generados sobre diferentes parámetros. En nuestro dominio de aplicación podemos observar que al utilizar gini obtenemos en promedio menos fragmentos por agrupación en todas las clases (comparar Φ_j en la segunda y cuarta columna).

Por otra parte la estadística Ψ_j revela más información acerca de la naturaleza de las particiones realizadas en el árbol de decisión. Se puede observar que information gain produce pequeños fragmentos. Esto se observa en la clase 1t donde ambas métricas producen el mismo valor en $\Phi_j = 2$ pero los árboles de decisión que utilizan information gain producen un valor más elevado para Ψ_j, lo cual señala la existencia de pequeños fragmentos. El mismo efecto se puede observar en casi todas las clases con la excepción de la clase 1s donde se observa el efecto opuesto. Es decir los árboles de decisión construidos con gini tienden a producir pequeños fragmentos.

Este análisis nos ayuda a realizar estimaciones en la predicción, de acuerdo con la clase en la que estamos interesados, uno puede decidir con mas información que tipo de árbol de decisión utilizar al favorecer a aquellos árboles que muestren pequeños valores para Φ_j (definiendo un pequeño número de fragmentos) y grandes valores para Ψ_j (representando a fragmentos que no son muy pequeños).

5.2. Trabajo futuro

A pesar de que las estadísticas generadas nos ayudan a evaluar que tipo de árbol reduce la fragmentación en el problema a resolver. Esto todavía no elimina la fragmentación en el árbol. Para resolver este problema podemos trabajar bajo dos perspectivas:

- Post-procesamiento.
- Una nueva métrica de impureza.
- Mejorar la técnica del árbol de generalidades y excepciones[9].
- Generar el algoritmo de árboles de decisión iterativos.

La técnica de post-procesamiento consistiría en utilizar el modelo del árbol de decisión, las agrupaciones de los datos con los cuales fue construido dicho modelo y las estadísticas generadas para modificar el modelo del árbol de decisión. Esta modificación tendría como finalidad definir unas nuevas particiones las cuales reduzcan el nivel de fragmentación existente en el modelo.

Estas estadísticas se podrían utilizar para generar una nueva métrica de impureza a ser utilizada en el proceso de inducción de los árboles de decisión. Dicha métrica evaluaría en base a las estadísticas el corte que genere la menor cantidad de fragmentación en el modelo resultante. Este control estaría basado en la estadística de Ψ_j la cual determina si el tamaño de los fragmentos es demasiado pequeño para causar errores de clasificación en el modelo.

También se podrían utilizar estas estadísticas para mejorar el trabajo realizado por Liu, Hu y Hsu, quienes presentan los árboles de generalidades y excepciones[9]. Con las estadísticas y el conocimiento de las agrupaciones podemos decidir de manera más acertada cuales son los nodos del árbol que deben de ser colapsado en uno solo para generar una generalidad mas adecuada.

Para generar un árbol de decisión de manera iterativa, es necesario ejecutar varias veces el algoritmo de aprendizaje de un árbol de decisión. La idea consiste en aprender el árbol y evaluar la fragmentación existente en dicho árbol, con base en esta información se detectan

los errores que se tienen en dicho árbol y se separan los datos involucrados en el error. Con estos nuevos datos se vuelve a ejecutar el algoritmo de aprendizaje de árboles de decisión, y el árbol resultante se inserta en el árbol generado en la iteración anterior. Dicho proceso continua hasta que ya no existe fragmentación o el árbol resultante es el mismo que en la iteración anterior.

Posteriormente estaremos trabajando en la generación de un árbol de decisión de manera sucesiva debido a que de esta forma tratamos de reducir la fragmentación existente en el modelo en cada iteración hasta llegar al modelo en el cual la naturaleza propia de los datos y el método de clasificación hagan imposible la reducción de la fragmentación.

Bibliografía

[1] et. al. Abazov. Evidence for production of single top quarks and first direct measurement of v_{tb}. *Physical Review Letters*, 98(18), May 2007.

[2] et. al. Abazov. Multivariate searches for single top quark production with the d0 detector. *prd*, 75(9), May 2007.

[3] Francis R. Bach and Michael I. Jordan. Learning spectral clustering. *Advances in Neural Information Processing Systems 16 NIPS 2003*, 2004.

[4] R. K. DeLisle and S. L. Dixon. Induction of decision trees via evolutionary programming. *Journal of Chemical Information and Modeling*, 44(3):862, 2004.

[5] D. J. Hand. Classifier technology and the illusion of progress. *Statistical Science*, 21(1):1, 2006.

[6] K.M.Ho and P.D. Scott. Overcoming fragmentation in decision trees through attribute value grouping. *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, page 337, 1998.

[7] Breiman L.1. Technical note: Some properties of splitting criteria. *Machine Learning*, pages 41–47, 1996.

[8] J. Li and L. Wong. Solving the fragmentation problem of decision trees by discovering boundary emerging patterns. *Proceedings of the 2002 IEEE International Conference on Data Mining ICDM*, page 653, 2002.

[9] B. Liu, M. Hu, and W. Hsu. Intuitive representation of decision trees using general rules and exceptions. *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, page 615, 2000.

[10] Huan Liu and Rudy Setiono. Feature transformation and multivariate decision tree induction. *Proceedings of the First International Conference on Discovery Science (DS'98)*, page 279, 1998.

[11] John R. Rice. The algorithm selection problem. *Advances in Computers*, pages 65–118, 1976.

[12] S. Rasoul Safavian and David A. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 660–674, 1991.

[13] R. Vilalta, G. Blix, and L. Rendell. Global data analysis and the fragmentation problem in decision tree induction. *Proceedings of the 9th European Conference on Machine Learning ECML*, page 312, 1997.

[14] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

[15] H. Zhao and A.P. Sinha. An efficient algorithm for generating generalized decision forests. *Systems, Man and Cybernetics, Part A, IEEE Transactions*, pages 754– 762, 2005.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Evaluación de la Fragmentación en el Aprendizaje de Árboles de Decisión Basada en el Análisis de las Particiones en Agrupaciones de Datos

del (la) C.

Roberto VALERIO MOLINA

el día 18 de Septiembre de 2009.

Dr. Juan Manuel Ramírez Arredondo
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Andrés Méndez Vásquez
Investigador CINVESTAV 2A
CINVESTAV