METODOS PARA OBTENER CONOCIMIENTO UTILIZANDO REDES BAYESIANAS Y PROCESOS DE APRENDIZAJE CON ALGORITMOS EVOLUTIVOS.

Departamento de Lenguajes y
Sistemas Informáticos.

Memoria de Investigación presentada por
D. Francisco Roche Beltrán
para superar la fase de investigación
del programa de doctorado.

Tutor: Dr. D. José Cristóbal Riquelme Santos.

Sevilla, Septiembre 2002

INDICE.-

1.- INTRODUCCIÓN.-

Con la revolución digital capturar información es fácil y almacenarla es extremadamente barato. Almacenamos datos porque pensamos que son un activo valioso por sí mismos. Para los científicos, los datos representan observaciones cuidadosamente recogidas de algún fenómeno en estudio. En los negocios, los datos guardan informaciones sobre mercados, competidores y clientes. En procesos industriales recogen valores sobre el funcionamiento de determinados procesos.

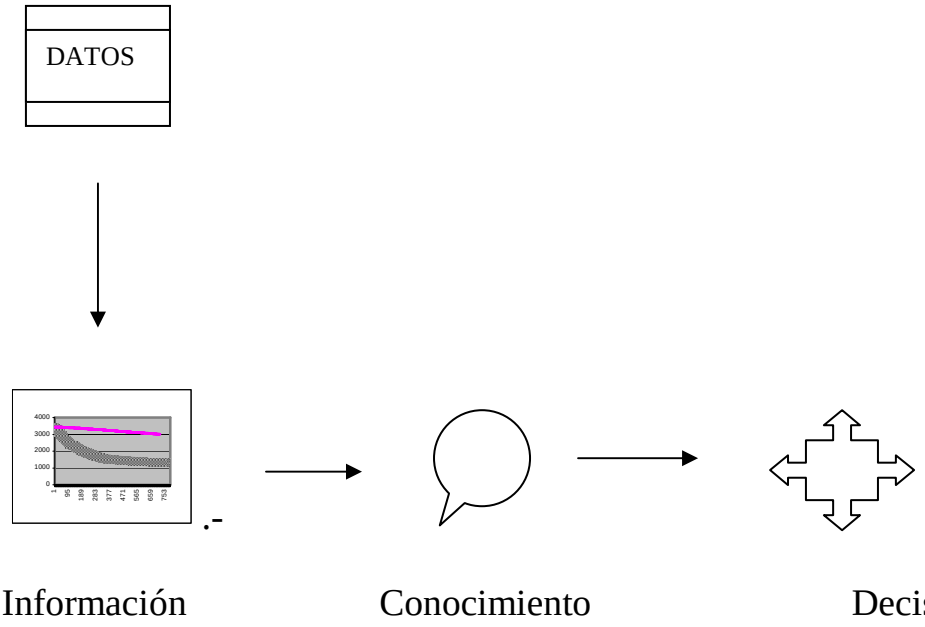
Sin embargo, en general, los datos en bruto raramente son provechosos. Su verdadero valor radica en la posibilidad de extraer información útil para la toma de decisiones o la exploración y comprensión de los fenómenos que dieron lugar a los datos. Tradicionalmente, el análisis de estos datos ha sido efectuado mediante técnicas estadísticas. No obstante, el incremento en la cantidad de datos y en el número de parámetros hace necesaria la aparición de nuevas metodologías y herramientas para un tratamiento automático de los registros depositados en bases de datos.

Estas técnicas se engloban bajo la etiqueta de machine learning, simultáneamente surge el nombre más comercial de minería de datos (data mining).

La automatización de los procesos de aprendizaje por un área de investigación de la inteligencia artificial se conoce como machine learning (ML) o aprendizaje automático. En aprendizaje supervisado, las técnicas de ML buscan descripciones para las clases ya definidas por el usuario, y en aprendizaje no supervisado se construye un resumen del fichero de entrenamiento como un conjunto de las clases descubiertas junto con su descripción. Esta búsqueda de descripciones se realiza usando estrategias de búsqueda iterativa en el conjunto de todas las descripciones posibles. Este proceso consiste en la formulación de una hipótesis inicial que se verifica mediante alguna función de calidad. Esta función, normalmente basada en técnicas estadísticas, calcula la corrección de la hipótesis respecto del conjunto de entrenamiento. Entonces, la hipótesis puede ser aceptada, rechazada o mejorada hasta que se encuentre una hipótesis correcta. La mejora de las hipótesis durante este proceso puede ser llevada a cabo mediante la generalización de condiciones o la adición o sustracción de condiciones sobre atributos.

Un sistema de ML usa un pequeño conjunto de ejemplos de laboratorio cuidadosamente seleccionados y, algunas veces, tiene la habilidad de interactuar con el entorno con el fin de conseguir nuevos ejemplos para investigar el comportamiento bajo condiciones particulares. Un sistema de ML tiene tres componentes básicos: una representación o modelo del conocimiento aprendido, una función que mida la calidad de ese aprendizaje y un algoritmo de búsqueda para dado un modelo y una función de calidad encontrar la mejor instanciación posible.

VISION GENERAL



Data Mining (DM) es la búsqueda de relaciones y patrones globales que existen en grandes bases de datos pero que se encuentran "ocultas" entre grandes cantidades de datos. Estas relaciones representan un conocimiento valioso sobre la base de datos y los objetos de ésta y, si la base de datos es un espejo fiel, del mundo real registrado por la base de datos.

Uno de los principales problemas del DM es que el número de posibles relaciones es muy grande, así que la búsqueda de las correctas por validación de cada una de ellas es computacionalmente prohibitivo. Así que se necesitarán estrategias de búsqueda inteligente que son tomadas del área de machine learning o aprendizaje automático.

En general, las tareas de un proceso de DM pueden ser clasificadas en dos categorías: descriptivas y predictivas. Las primeras describen el conjunto de datos de una manera resumida y concisa y presentan propiedades generales e interesantes de los datos. Por el contrario, las tareas predictivas construyen uno o varios modelos que realizan inferencia sobre el conjunto de entrenamiento para intentar predecir el comportamiento de nuevos datos.

Un sistema de DM pueden llevar a cabo una o más de las siguientes tareas:

1. Descripción de clases. Mediante esta tarea se proporciona un conciso y sucinto resumen de una colección de datos o caracterización y la posibilidad de distinguirlos de otros o discriminación. Un ejemplo simple es obtener la media y la desviación típica de cada parámetro para cada clase. Un ejemplo más sofisticado lo constituyen las técnicas de visualización en múltiples dimensiones.

2. Asociación o descubrimiento de relaciones o correlaciones entre un conjunto de datos. Estas normalmente se expresan en forma de regla mostrando condiciones que relacionan valores de los atributos y que ocurren frecuentemente entre los datos. Una regla de asociación tiene la forma $X \rightarrow Y$ que debe ser interpretada como "los datos que satisfacen X probablemente satisfacen Y".

3. Clasificación o análisis de un conjunto de entrenamiento con clase conocida y construye un modelo para cada clase. Un árbol de decisión o un conjunto de reglas de clasificación se genera mediante un proceso de clasificación, que puede usarse para una mejor comprensión de cada clase o para la clasificación de futuros datos.

4. Predicción o regresión. Esta tarea proporciona posibles valores para datos desconocidos o ausentes o la distribución de valores de ciertos atributos en un conjunto de objetos, incluyendo la posibilidad de encontrar los atributos relevantes o interesantes para determinados casos.

5. Clustering o identificación de subconjuntos de objetos que tienen datos similares entre sí. Un buen método de clustering es aquel que consigue una baja similaridad inter-cluster y una alta similaridad intra-cluster.

6. Análisis de series temporales: búsqueda de regularidades, secuencias o subsecuencias similares, periodicidad y tendencias en datos que dependen del tiempo.

Un proceso de DM consiste básicamente en ajustar modelos y/o determinar patrones a partir de unos datos. Todo algoritmo de DM tiene los mismos tres componentes que un proceso de ML:

- El **modelo** con dos factores relevantes: la función que se desee desempeñe (v. gr. clasificación, clustering, etc.) y la forma (función lineal, conjunto de reglas,...)
- El criterio de preferencia o **función de bondad** para ajustar el modelo a los datos.
- El **algoritmo de búsqueda** para ajustar los parámetros de un modelo particular a unos datos y una función de ajuste.

2.- PLANTEAMIENTO Y RELEVANCIA DEL PROBLEMA.-

En la figura 1 se presenta un esquema del sistema. Básicamente se trata de un sistema de flujo de gases que contiene azufre en mayor o menor medida. Sometiendo a estos gases a los procesos adecuados puede aprovecharse ese azufre para producir ácido sulfúrico en lugar de dejarlo escapar a la atmósfera, de esta forma se contribuye a proteger el medio ambiente a la vez que se dispone de un producto comercializable.

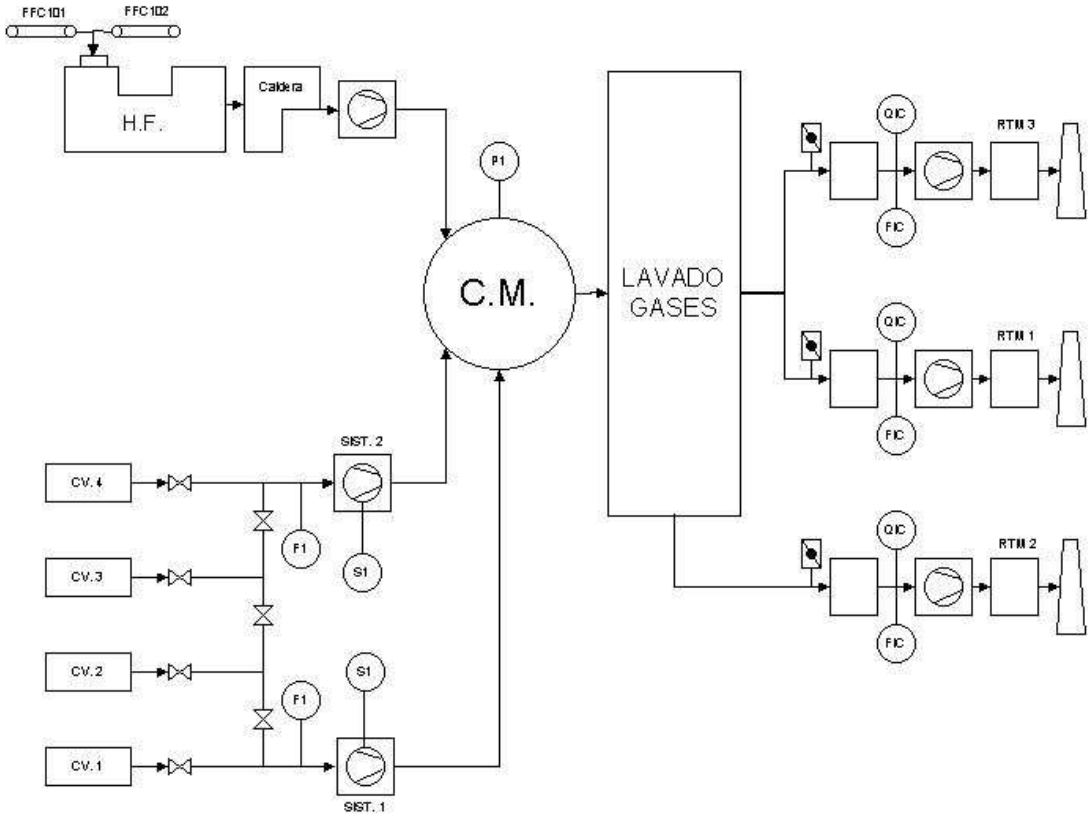


Figura 1: Sistema de producción de ácido sulfúrico

En el sistema existen productores y consumidores de gases. El proceso consiste en canalizar adecuadamente los gases que generan los productores hacia los consumidores de manera que no haya excedente de producción a la vez que se atienden las necesidades de los consumidores. Los gases producidos son conducidos a la Cámara de Mezcla (CM) a partir de la cual se distribuyen a los distintos consumidores según sus necesidades.

Como productores tenemos el Horno Flash (HF) en donde se produce el cobre y se genera SO_2 y los Convertidores (CV1, CV2, CV3 y CV4). Como consumidores están las tres plantas de la Fábrica de Ácido SO_4H_2 (RTM1, RTM2 y RTM3). Cada uno de los productores y consumidores constituye un subsistema completo y muy complejo dentro de la empresa.

Los gases se conducen de un punto a otro a través de soplantes, de forma que se dice que el extremo hacia el cual la soplante actúa tiene presión, mientras que el extremo contrario tiene tiro. Es de vital importancia que en la cámara de mezcla siempre haya tiro, ya que un aumento de presión puede producir que los gases escapen a la atmósfera, lo cual hay que evitar en todo momento.

Debido a la complejidad del sistema y al gran número de parámetros y situaciones que se dan continuamente hay veces que el sistema no es estable durante un tiempo determinado. Durante este tiempo es posible que haya un aumento de presión en la cámara de mezcla. Aunque en la actualidad se toman todas las medidas oportunas para evitar que esto ocurra, sería deseable prever o conocer las circunstancias que provocan que se produzca alguna inestabilidad en el tiro de la cámara de mezcla.

Se ha intentado resolver el problema, pero lo único que han realizado realmente es la colocación de gran cantidad de sensores en la instalación y el almacenamiento de los datos que se generan, sin haber hecho ningún estudio de estos datos posteriormente.

Como resultado del funcionamiento del sistema se dispone de una inmensa base de datos que representa el estado del sistema a lo largo del tiempo. Sería deseable saber si a través de un análisis de estos datos puede llegarse a alguna conclusión.

El estudio propuesto, por tanto, trata de obtener información acerca del comportamiento del sistema para averiguar las circunstancias que provocan situaciones de funcionamiento incorrecto del sistema.

Metodología y plan de trabajo

Dado que nos encontramos ante un problema de adquisición de conocimiento en bases de datos (KDD), las acciones a realizar son las propias de todo proceso de esta índole.

1. Entendimiento del dominio de la aplicación: Incluyendo el conocimiento a priori relevante y los objetivos de la aplicación. En esta fase hay que determinar los objetivos, es decir, definir lo que se desea obtener y saber con qué datos se cuenta para la consecución de dichos objetivos. Además hay que estudiar las limitaciones derivadas de estos datos dependiendo de su calidad, su número, etc.

2. Creación del conjunto de datos de entrenamiento: Seleccionando el subconjunto de variables o ejemplos sobre los que se realizará el descubrimiento. En base a los datos con los que se cuenta es necesario seleccionar parte de los mismos con el fin de optimizar las tareas del proceso de aprendizaje.

3. Preprocesado de los datos: La calidad de los datos influye en la calidad de los resultados. Hay varios factores que influyen en la calidad de los datos, como son los datos erróneos, los datos ausentes, los outliers y otros que hay que corregir: Hay que eliminar el ruido dado por los posibles outliers, decidir sobre el tratamiento que se da a los datos ausentes, normalizar los datos, etc. Una vez finalizada esta fase se

contaría con un conjunto de datos totalmente preparados para aplicarles las técnicas de aprendizaje.

4. Transformación y reducción de los datos: Incluyendo la búsqueda de parámetros útiles para representar los datos dependiendo del objetivo, reducción del número de variables mediante transformación, etc.

5. Elección del método o algoritmo: Dependiendo de las particularidades del proyecto una vez llegado a esta fase, hay que determinar el método de proceso que va a emplearse (clasificación, regresión, clustering, etc.). Esto llevará a la decisión de utilizar un algoritmo de data mining u otro.

6. Proceso de data mining: El proceso de data mining, propiamente dicho, consiste en la búsqueda de patrones y relaciones que existan en los datos. Este proceso es el que permite extraer información oculta de la nube de datos con la que se contaba. El conocimiento extraído viene representado mediante árboles de decisión, reglas de asociación, etc

7. Interpretación del conocimiento extraído: Finalizado el proceso de data mining hay que revisar la información obtenida y evaluarla para seleccionar la información que sea de interés y estudiar, en base a los resultados, un posible regreso a alguno de los puntos anteriores. La evaluación del aprendizaje puede realizarse mediante técnicas de visualización o definiendo medidas de interés (de tipo estadístico, en función de su sencillez, etc.)

8. Uso del conocimiento descubierto: El conocimiento extraído en las fases anteriores se incorporará al sistema para verificar su utilidad y tomar decisiones en base a los resultados. Dicho conocimiento deberá documentarse, revisarse periódicamente, compararlo con el conocimiento anterior, etc.

3.- ASPECTOS RESUELTOS Y POR RESOLVER.-

El proceso de producción del cobre requiere de unos sistemas muy complejos que hacen necesario el acoplamiento de varios subsistemas, cada uno de los cuales se encarga de una fase del proceso productivo. Los subprocesos realizados por estos subsistemas liberan una serie de materiales que, en lugar de constituir material de desecho, pueden servir como subproducto.

En varias fases del proceso productivo se generan gases con contenido en azufre, el cual se aprovecha para producir ácido sulfúrico.

Los objetivos que se persiguen son:

- Identificar los parámetros que influyen en el comportamiento del sistema.
- Conocer el grado de influencia en el sistema de cada uno de los parámetros.
- Saber qué respuesta va a tener el sistema ante actuaciones sobre los parámetros.
- Conocer las causas de los periodos de inestabilidad del sistema.
- Detectar posibles perturbaciones.
- Detectar actuaciones innecesarias o perjudiciales en el control del sistema, así como descubrir nuevas acciones no consideradas.
- Obtener reglas de funcionamiento que permitan realizar un posterior modelo del sistema lo más exacto posible.

Con la consecución de los objetivos la empresa obtendrá:

- Una optimización del funcionamiento del sistema de producción de ácido sulfúrico.
- Un mayor conocimiento del sistema que sirva de ayuda en una posterior toma de decisiones.
- Una reducción de costes de producción.
- Una mayor protección del medio ambiente.

4.- COMPARATIVA DE PROPUESTAS.-

Las principales formas de representar el conocimiento en machine learning son las siguientes:

4.1 Árboles de decisión.-

La representación de un árbol donde los nodos son atributos discretos o condiciones sobre atributos continuos, las ramas son los posibles valores de un atributo discreto o verdadero y falso en el caso de condiciones; por último, las hojas son las clases. La herramienta más popular que genera árboles de decisión es el C4.5[Quinlan 93]. También es de reseñar la herramienta CART[Breiman 84].

4.2 Sistema basado en reglas.-

Una regla es una expresión de la forma:

Si A entonces B
En donde A es un aserto y B puede ser una acción o bien otro aserto.

Por ejemplo:

1. Si la bomba falla entonces la presión es baja.
   Aserto aserto

2. Si la bomba falla entonces chequear el nivel de aceite.
   Aserto acción

3. Si hay fallo de potencia entonces la bomba falla.
   Aserto aserto

Un sistema basado en reglas es una librería de reglas.

Estas reglas reflejan esencialmente las relaciones dentro del dominio del problema, más bien reflejan el camino para razonar sobre el dominio.

Cuando tenemos después información concreta del dominio, esto se aplica a las reglas y te lleva a conclusiones apuntando a acciones determinadas. Esto se llama INFERENCIA.

Por ejemplo en el caso anterior del dominio sabemos ahora que hay fallo de potencia.

Por la regla 3 nos dice que la bomba falla, y aplicando la regla 1 nos dice que la presión será baja, además la regla 2 nos RECOMIENDA que chequeemos el nivel de aceite (acción).

Las reglas también se pueden usar en dirección opuesta.

Por ejemplo supongamos que sabemos actualmente que la presión es baja, por la regla 1 esto puede ser debido a que la bomba falla y por la regla 3 te dice que puede ser debido a un fallo de potencia, y la regla 2 te recomienda chequear el nivel de aceite.

Un inconveniente que tienen consiste cuando los atributos son continuos, es preciso en este caso realizar una discretización previa.

Normalmente las conexiones reflejadas mediante reglas no son absolutamente ciertas, por lo tanto están sujetas a cierta incertidumbre. En estos casos una medida de incertidumbre hay que añadir a las reglas, tanto a las premisas como a las conclusiones.

Si A (con certidumbre x) entonces B (con certidumbre f(x))

Hay muchos esquemas para tratar la incertidumbre en un sistema basado en reglas, la más común es la lógica fuzzy.

En estos esquemas la incertidumbre se trata LOCAMENTE, es decir se añade la incertidumbre a las reglas.

Por ejemplo:

Si C (con certidumbre x) entonces B (con certidumbre g(x))

Supongamos que la evidencia actual te indica que se produce A con certidumbre a y se produce C con certidumbre c ¿Cuál es el grado de certidumbre de B?

Para representar una regla se puede usar una formulación CNF, o conjunción de cláusulas que son disyunciones de condiciones sobre los atributos, v. gr.:

color $\in$ {rojo, verde} $\wedge$ forma $\in$ {círculo} $\wedge$ altura ≤ 13

También se pueden formular reglas mediante lógica fuzzy (borrosa) [Bezdek 81], [Sugeno 93] o utilizando el concepto de conjuntos rough (aproximado) [Pawlak 91]. Las herramientas más conocidas que producen reglas son la familia de algoritmos AQ [Michalski 87] o los sistemas GIL [Janikow 93] y GABIL [DeJong 93] basados en algoritmos genéticos. También utilizan una formulación de antecedente $\Rightarrow$ consecuente denominadas reglas de asociación [Agrawal 93].

4.3 Listas de decisión.-

La lista de decisión [Rivest 87] es una representación del conocimiento de la forma:

(d_1, C_1), (d_2, C_2), ... , (d_n, C_n)

en donde cada d_i es una descripción elemental y cada C_j es una clase.

La clase de un objeto será C_j cuando d_j sea la primera descripción cubierta o satisfecha por el objeto. Otra forma de representar una lista de decisión es una regla de la forma:

Si d_1 entonces C_1 sino si d_2 ... sino si d_n entonces C_n

El sistema CN2 [Clark 89] es una de las herramientas más conocidas que utilizan listas de decisión. COGITO [Riquelme 98] utiliza también esta representación del conocimiento realizando la búsqueda mediante un algoritmo genético.

4.4 Redes Neuronales.-

Las redes neuronales [McCulloch 43] son una representación mediante un grafo del sistema nervioso de los seres vivos. El grafo se organiza en una serie de capas o leyes, formadas cada una por un conjunto de nodos que se relacionan con la capa anterior y posterior recibiendo unos valores numéricos o impulsos ponderados. La primera capa toma los datos del fichero de aprendizaje y la última capa es denominada de salida, cuyos nodos adquieren (mediante un proceso de aprendizaje) distintos valores para patrones distintos. En aprendizaje supervisado el modelo más clásico es el perceptrón multicapa [Rosenblatt 58] y en no supervisado las redes autoorganizadas [Kohonen 82].

Todos los nodos de un nivel están conectados a todos los nodos del nivel superior y del nivel inferior.

Ahora bien, necesita un entrenamiento costoso ya que se deben de encontrar los pesos que tiene cada nodo en el siguiente nivel, partiendo de los ejemplos de entrenamiento en donde los valores de entrada y salida son conocidos.

De sobra es conocido que NO obtiene reglas, y por lo tanto conocimiento sobre la base de datos, sí clasifica pero no sabemos CÓMO.

Se puede decir que es un buen sistema, si sólo necesito clasificar, pero no obtener reglas.

Si las relaciones son con cierto grado de certidumbre, la red puede dar la hipótesis más probable, dado un conjunto de síntomas, en diagnosis médica, sin embargo no podremos obtener el grado de incertidumbre de la conclusión, y no seremos capaces de obtener cual es la hipótesis siguiente más probable.

4.5 Aprendizaje basado en ejemplos.-

Los sistemas de aprendizaje basados en ejemplos representan el conocimiento mediante ejemplos representativos, basándose en similitudes entre los datos. Los algoritmos más extendidos son los clasificadores basados en los vecinos más cercanos [Dasarathy 91]. Habitualmente estas técnicas realizan el aprendizaje a partir de una selección de los ejemplos que mejor representan a los conceptos existentes en la base de datos. Es el concepto de editing de los datos o selección de prototipos.

4.6 Redes bayesianas.-

Esta técnica se encuentra desarrollada en detalle en el apartado número 5 de esta memoria de investigación especialmente en el punto 5.5, describiendo a continuación un resumen de la misma.

La idea esencial consiste en aprovechar las relaciones de dependencia (y por tanto también las de independencia) existentes entre las variables de un problema antes de especificar y calcular con los valores numéricos de las probabilidades [Pearl 86].

Estas relaciones se representan a través de modelos gráficos, habitualmente grafos acíclicos dirigidos [Cooper 92] y [Heckermann 95].

Formalmente se define como red bayesiana una tripleta (N,D,P) en donde

- N es un conjunto de variables del dominio.
- D es una DAG (Grafo acíclico dirigido) cuyos nodos están etiquetados con los elementos de N y los arcos dirigidos indican relación de influencia y en algunos casos relación causal.
- P es una distribución joint sobre N.

D reúne la información de que toda variable i es independiente de sus no descendientes dados sus padres (Padres(i)).

Esto permite expresar que si una instancia está formada por los atributos y_1, y_2, ... y_n:

$$P(y_1,...y_n) = \prod_{i=1}^{n} P(y_i \mid Padres(y_i))$$

Esto se conoce como chain-rule.

El número de modelos (redes bayesianas) diferentes que son posibles, se eleva de manera considerable en función del número n de variables a considerar:

$$2^{\frac{n * (n-1)}{2}}$$

Ya que el número de modelos es realmente grande cuando las variables son muchas, se impone un método de búsqueda para la elección del modelo más probable.

4.7 Conclusiones.-

El modelo va a tener como **función principal** el estudio de la relación de dependencia de parámetros.

Ahora bien la **forma** adoptada es la red bayesiana, los motivos que influyen en esta decisión son:

- El grafo generado por las redes bayesianas representan de **forma visual** la relación de dependencia existente entre las variables del dominio del problema.

- Las redes bayesianas tratan la **incertidumbre de forma global,** por el contrario tanto las reglas de asociación como las redes neuronales la tratan de forma local.

- Las redes bayesianas se pueden aplicar a **variables no cuantificables**, detalle con el que no puede trabajar la técnica de los vecinos, por ejemplo.

- Las redes neuronales se utilizan para clasificar, pero **no dan reglas** de cómo obtienen esta información.

- Un primer inconveniente se centra en que las redes bayesianas trabajan con **variables discretas**, pero esta misma situación se da con las reglas de asociación.

- Un segundo inconveniente resulta del gran **coste computacional** al tener que calcular las funciones de distribución de todas las variables, pero lo mismo le ocurre a las redes neuronales cuando tienen que calcular los distintos pesos en cada nodo.

5.- TECNICAS BAYESIANAS.-

5.1 Introducción.-

En los últimos años los sistemas expertos probabilistas han alcanzado un alto grado de desarrollo. Hasta los 80 se había dado por supuesto que la probabilidad requería mucha información y unos cálculos demasiado complejos para poder resolver problemas reales en los que interviniesen un gran número de variables.

Sin embargo esto cambió a partir de una serie de trabajos entre los que destacan los de [Pearl 1986], [Pearl 1988], [Jensen 1996] y [Lauritzen 1988].

La idea esencial fue la de aprovechar las relaciones de dependencia (y por tanto también las de independencia) existentes entre las variables de un problema antes de especificar y calcular con los valores numéricos de las probabilidades.

Estas relaciones se representan a través de modelos gráficos, habitualmente grafos acíclicos dirigidos [Whiltaker 1990].

La determinación de las relaciones existentes entre las variables se manifestó desde el principio como una cuestión fundamental. En muchas ocasiones el problema está bien estructurado y el experto sabe determinar directamente un modelo gráfico. Sin embargo, es más habitual que no se conozcan, al menos en forma total, las relaciones de influencia entre los elementos que intervienen.

El objetivo de los algoritmos de aprendizaje consiste en determinar un modelo gráfico a partir de un conjunto de datos en bruto u observaciones realizadas sobre el comportamiento del sistema, referencias importantes son [Cooper 1992] y [Heckerman 1995]. Existe en la actualidad una gran diversidad de enfoques y métodos de resolución.

Otro detalle importante consiste en que constituyen una forma de trabajo coherente y efectiva para sistemas de soporte a la decisión que deben de funcionar con conocimiento no seguro (incertidumbre).

Si bien un problema que tienen, consiste en que tenemos que discretizar las variables continuas.

Veamos la siguiente información:

P(enfermo tenga gripe | fiebre>38.5, dolor de cabeza = SI) = 0.75

En una red bayesiana esto quiere decir que si conocemos el valor de los padres de “enfermo tiene gripe” que son “fiebre” y “dolor de cabeza” podemos determinar si el enfermo tiene gripe o no, con un grado de incertidumbre, además el resto de valores de variables no influyen en el valor de si tiene gripe o no.

Todas las variables que aparecen son variables del dominio que tratamos y no como en las redes neuronales que los nodos no tienen representación en el dominio.

La información anterior podría verse como una regla de decisión:

Si (fiebre>38.5 y dolor de cabeza = Si) entonces enfermo tiene gripe (con grado de certidumbre 0.75)

Por lo tanto tienen mucha similitud con las reglas de decisión.

En un sistema basado en reglas se intenta modelizar el camino de razonamiento de los expertos mientras que en una red bayesiana se intenta modelizar las dependencias existentes en el dominio en sí mismo.

En un sistema basado en redes neuronales es imposible introducir conocimiento a priori en los pesos de los nodos, mientras que utilizando redes bayesianas sí se puede realizar introduciendo conocimiento del experto, con dependencias señaladas entre variables. Respecto al coste computacional señalar que las redes bayesianas tienen que calcular muchas funciones de distribución de probabilidad, pero no más que las redes neuronales que deben de calcular los pesos de los nodos. Además en las redes neuronales la dirección de la inferencia queda totalmente delimitada, cosa que en las redes bayesianas esto es mucho más flexible.

Un punto a favor de las redes bayesianas consiste en que se pueden aplicar cuando los atributos tienen valores no cuantificables, más bien no debe de existir información numérica ni de orden.

Por ejemplo si consideramos la variable aleatoria color de un coche, y etiquetamos
blanco..0, negro..1, azul..2

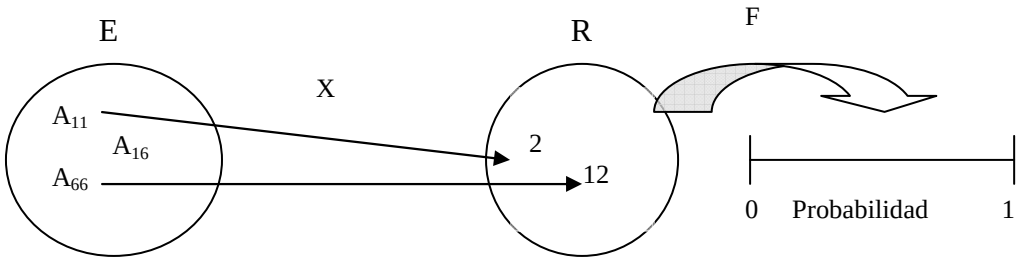
La distancia que hay del blanco al negro es 1, y la distancia que hay del blanco al azul es 2, pero esta distancia hubiera sido diferente desde el momento en que asignáramos las etiquetas numéricas diferentes al blanco, negro y azul, entonces aplicar técnicas como las de los vecinos no tendría sentido.

5.2 Conceptos básicos.-

5.2.1 Variables aleatorias discretas / continuas.-

Si lanzamos dos dados sobre una mesa, el espacio muestral E estará constituido por las 36 parejas:

E = { (1,1), (1,2), (1,3) ... (6,5), (6,6) }



$X : E \rightarrow R$

X es una aplicación que hace corresponder a cada suceso A_{ij} la suma (i + j) de los puntos aparecidos.

$$A_{11} \xrightarrow{X} 2 \quad (R)$$

X(E) = { 2,3,4 ... 12}. Llamándose a la función X variable aleatoria, siendo las inversas de las imágenes en R sucesos en E.

A cada elemento X(E) le asociamos su probabilidad.

Tenemos así definida una nueva función F de la siguiente forma:

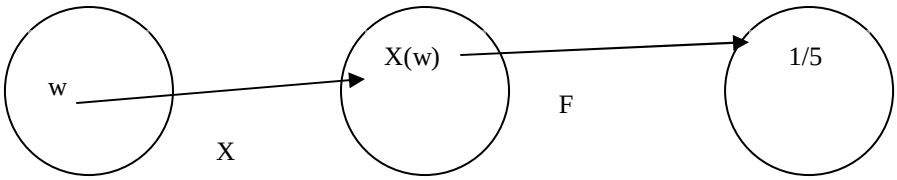
F(2) = P(X=2) = 1 / 36.
F(3) = P(X=3) = 2 / 36.
...
F(12) = P(X=12) = 1 / 36.

Una función definida de esta forma se denomina función de distribución, es una función suprayectiva poseyendo las siguientes propiedades:

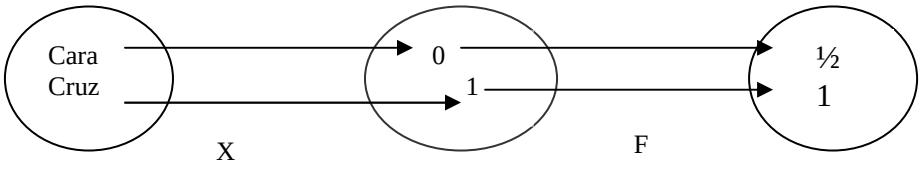
1. $F(X_i) >= 0, \quad \forall X_i.$
2. $\Sigma F(X_i) = 1.$

- Es una función que aplica R en R.
- No decreciente.
- Continua a la derecha. Es decir, existe $\lim_{x \to a \text{ derecha}} F(X) = F(a)$.
- $F(+\infty) = 1.$
- $F(-\infty) = 0.$

Luego F(x) = P{ w : X(w) <= x} siendo la función de distribución de la variable aleatoria X.

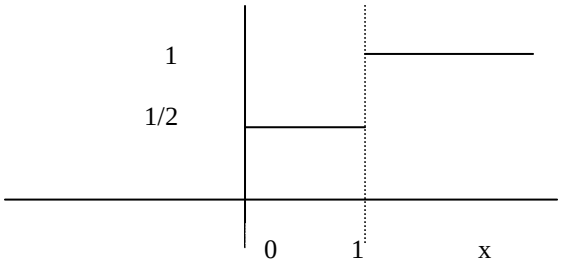


Por ejemplo si w_1 es sacar cara y w_2 es sacar cruz:



$X(w_1) = 0.$
$X(w_2) = 1.$

Luego F(0) = P{ sacar cara} = ½.
F(1) = P{ sacar cara o cruz} = 1.



Una variable aleatoria se dice que es discreta si existe un conjunto numerable E (es decir existe una biyección entre los números naturales y elementos x del conjunto), tal que

$E \subset R \mid P(x \in E) = 1.$

Los puntos que tienen masa de probabilidad se denominan puntos de salto o puntos de incremento de la función de distribución. Por ejemplo en el dibujo de esta página solo x = 0 y x = 1 tienen masa de probabilidad.

La función de distribución (F) de la variable aleatoria discreta será:

$F(x) = P\{w \mid -\infty < X(w) <= x\}$, es decir la suma de las probabilidades de todos aquellos puntos hasta llegar al x, incluido.

Una variable aleatoria se dice que es continua si la función de distribución F(x) es absolutamente continua, es decir si existe una función no negativa f(x) $\forall x \in R$ que:

$$F(x) = \int_{-\infty}^{x} f(t)\,dt$$

La función f(x) recibe el nombre de función de densidad de probabilidad.

Luego f(x) >=0 ya que es no negativa y su integral desde -∞ a +∞ vale 1.

En una variable aleatoria continua la probabilidad está definida para intervalos de puntos y para puntos concretos vale cero.

5.2.2 Regla de la multiplicación, teorema de la probabilidad total y sucesos independientes.-

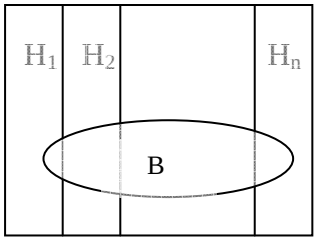
Regla de la multiplicación.-

Sean A_1, A_2 ... A_n sucesos cualesquiera, se cumple que

$P(A_1 \cap A_2 \cap A_3 ... \cap A_n) = P(A_1) * P(A_2|A_1) * ... * P(A_n|A_{n-1}, A_{n-2}..A_1)$

Teorema de la probabilidad total.-

Sean los sucesos H_1, H_2 ... H_j sucesos disjuntos y sea el suceso B que cumple $B = (B \cap H_1) \cup (B \cap H_2)$ ... representado en la figura.



Entonces se cumple que:

$$P(B) = \sum [\ P(H_j) * P(B \mid H_j)]\ \text{ para todo j}$$

Sucesos independientes.-

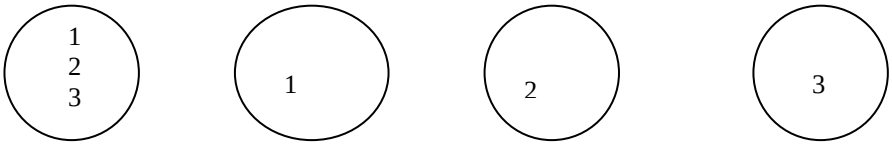
Dos sucesos A y B se dice que son independientes si y sólo si:

$$P(A \cap B) = P(A) * P(B).$$

Para el caso de n sucesos A_1, A_2 ... A_n se debería cumplir:

$$P(A_1 \cap A_2 \cap A_3 ... \cap A_n) = P(A_1) * P(A_2) * ... * P(A_n)$$

Por ejemplo, tengamos cuatro bolas en una bolsa con las siguientes etiquetas.



Se extrae una bola de las cuatro, sea el suceso E_1 en la bola aparece un 1, el suceso E_2 en la bola aparece un 2 y el suceso E_3 en la bola aparece un 3.

$P(E_1) = ½.$
$P(E_2) = ½.$

$P(E_1 \cap E_2) = ¼.$ $\qquad$ $P(E_1) * P(E_2) = ¼$

Luego el suceso E_1 y el suceso E_2 son independientes.

Sin embargo los sucesos E_1 , E_2 y E_3 no son independientes, ya que:

$P(E_1 \cap E_2 \cap E_3) = ¼$
$P(E_1) * P(E_2) * P(E_3) = 1/8.$

5.2.3 Teorema de Bayes.-

Descubierto por Thomas Bayes en 1761.

Consideremos el siguiente experimento: supongamos dos bolsas A y B; la primera que contiene dos bolas blancas y dos negras, y la segunda, dos blancas y una negra.

Si llamamos X al suceso obtener bola blanca e Y al obtener bola negra, y hacemos una serie de pruebas que constan cada una de dos partes: 1º sorteo de bolsas; 2º extracción al azar de una bola de la bolsa que corresponda.

Podemos obtener por ejemplo la siguiente serie:
Ba, Ba, Nb, Na, Na, Bb, Ba, Na, Nb
Expresando la letra mayúscula el color de la bola y la minúscula si ha sido de la primera bolsa(a) o de la segunda bolsa(b).

La frecuencia (fr) será:
fr(X) = 4/9; fr(A|X) =3/4; fr(A)=6/9; fr(X|A)=3/6;

Vemos en este ejemplo que se cumple la siguiente relación:

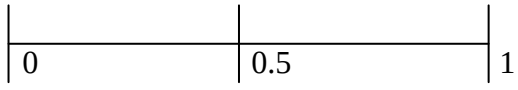
$P(X \cap A) = P(X) * P(A \mid X) = P(A) * P(X \mid A) = 1/3,$

Luego:

P(A|X) = P(A) *P(X|A) / P(X), que es esencialmente la fórmula de Bayes, siempre que está definida la P(X) y sea distinta de cero.

El teorema de Bayes gobierna el proceso de inferencia lógica, determinando el grado de confianza que debemos tener, con varias posibles conclusiones, basadas en el cuerpo de la evidencia disponible. Esto es exactamente el proceso de razonamiento predictivo.

Escala de credibilidad



P(A |E) = 0, quiere decir que A debe de ser falso si parto de que se cumple E.
P(A |E) = 1, quiere decir que A debe de ser cierto si parto de que se cumple E.
P(A |E) = 0.5, quiere decir que E no influye en A.

Cada ejemplo de entrenamiento puede incrementar/decrementar la probabilidad estimada de que una hipótesis sea correcta, por lo tanto supone una aproximación muy flexible de aprendizaje.

El conocimiento a priori puede ser combinado con los datos observados para determinar la probabilidad final de cada hipótesis. La probabilidad a priori se determina asignando una a cada hipótesis inicialmente y considerando la distribución de probabilidad en los datos observados para cada hipótesis posible.

Un inconveniente que tiene este tipo de razonamiento es que requiere conocimiento inicial de muchas probabilidades, cuando no se conocen es preciso estimarlas, así como el posible coste computacional.

5.2.4 Hipótesis MAP e hipótesis ML.-

En aprendizaje automático estamos interesados en determinar la mejor hipótesis de un espacio H, dando los datos observados D.

La mejor hipótesis es aquélla que sea la más probable, teniendo en cuenta los datos D y cualquier conocimiento inicial de las probabilidades a priori de varias hipótesis dentro de H.

El teorema de Bayes proporciona un método para calcular la probabilidad de una hipótesis basada en su probabilidad a priori, la probabilidad de observar esos datos D dada la hipótesis, y los datos observados por sí mismos.

Sea h la hipótesis a considerar y D los datos observados.

P(h $\cap$ D) = P(h) * P(D|h).
P(h $\cap$ D) = P(D) * P(h|D).

Es decir: P(h) * P(D|h) = P(D) * P(h|D).

Por lo tanto

$$\underbrace{P(h \mid D)}_{\text{a posteriori}} = \underbrace{P(h)}_{\text{a priori}} * \underbrace{\frac{P(D \mid h)}{P(D)}}_{\text{Factor de corrección}}$$

P(h) es la probabilidad a priori de que se cumpla la hipótesis h, es decir el conocimiento que tengamos de que la hipótesis h es correcta.

P(h|D) es la probabilidad a posteriori de que se cumpla la hipótesis h una vez conocidos los datos D, reflejando la influencia que tienen los datos D observados a diferencia con la probabilidad a priori en la que no se tiene en cuenta.

P(D|h) es la probabilidad de que los datos D sean observados en un mundo en el que la hipótesis h es correcta.

Por lo tanto P(h|D) aumenta si se incrementa P(h) y P(D|h) y se decrementa en el caso de que aumente P(D).

Una vez que disponemos de una fórmula que nos da la probabilidad de una hipótesis, estamos interesados en obtener aquella hipótesis más probable (maximum a posteriori MAP) observados los datos D.

$h_{MAP} = \text{argmax}_h \; P(h|D) = \text{argmax}_h \; [P(h) * P(D|h) / P(D)]$

Ya que P(D) es la misma en todas las hipótesis, en la obtención del máximo podemos ahorrarnos el cálculo de este valor, quedando:

$$h_{MAP} = \text{argmax}_h \; P(h) * P(D|h)$$

h_{MAP} es la hipótesis más probable, dados los datos observados, P(h|D).

En muchos casos podemos asumir que cada hipótesis en H es equiprobable a priori, $P(h_i) = P(h_j)$ para todo h_i y h_j en el espacio H, por lo tanto en la fórmula anterior podemos suprimir el término P(h) que es idéntico en todas las hipótesis. A la fórmula que nos queda se le denomina función de máxima verosimilitud (maximum likelihood, ML), es decir la hipótesis que maximiza la probabilidad de obtener los datos D partiendo de que se cumple esa hipótesis, P(D|h).

$$h_{ML} = \text{argmax}_h \; P(D|h)$$

Por lo tanto en el maximum likelihood no influyen las probabilidades a priori.

LIKELIHOOD (verosimilitud).- Probabilidad de obtener los "datos observados" a partir de un modelo.

$$\mathbf{P(D|M)}$$

MAXIMUM LIKELIHOOD.- Aquel modelo que obtiene un máximo de probabilidad de obtener los datos observados.

$$\mathbf{\text{argmax}_m \; P(D|M)}$$

En algunas ocasiones, más que obtener h_{ML} se obtiene el logaritmo neperiano de h_{ML}, ya que maximizar el logaritmo de una función maximiza el valor de la propia

función, todos los productos pasan a ser sumas y los exponentes desaparecen, limitando el caso de desbordamientos en las operaciones de cálculo y lo hacen más tratable matemáticamente.

$$h_{ML} = \operatorname{argmax}_h \; Ln\ P(D|h)$$

Es preciso observar que hay que evaluar todas las posibles hipótesis existentes en el espacio H.

Consideremos un caso:

Posibles hipótesis

- el paciente tiene una forma de cáncer.
- el paciente no tiene esa forma de cáncer.

Los resultados de una posible prueba dan + o bien - .

El conocimiento a priori que tenemos nos dice que 8 de cada mil personas tiene esa forma de cáncer, 98 de cada 100 personas que tienen esa forma de cáncer que se someten a la prueba dan + y 97 de cada 100 personas que no tienen esa forma de cáncer que se someten a esa prueba dan -.

P(cáncer) = 0.008. P(no cáncer)= 0.992.
P(+ | cáncer) = 0.98. P(- | cáncer) = 0.02.
P(+ | no cáncer) = 0.03. P(- | cáncer) = 0.97.

Supongamos que tenemos un nuevo paciente y en el test da valor +, calculamos $h_{MAP.}$

Hipótesis a) tiene cáncer.-
P(cáncer) * P(+| cáncer) = 0.98 * 0.008 = 0.0078

Hipótesis b) no tiene cáncer.-
P(no cáncer) * P(+ | no cáncer)= 0.03 * 0.992= 0.0298

Luego en este caso la hipótesis MAP es NO TIENE CÁNCER.

La probabilidad real se puede saber ya que la suma de las dos probabilidades debe de dar 1, ya que son las dos hipótesis posibles, por lo tanto **NO tiene cáncer con una probabilidad del 79%.**

$$\left.\begin{array}{l} 0.0298 \rightarrow 0.0376 \\ x \rightarrow 1 \end{array}\right\} \quad x = 0.0298 / 0.0376 = 0.79$$

Podemos observar que aunque la probabilidad a posteriori de tener cáncer es muy superior a la probabilidad a priori antes del test la conclusión sigue siendo que la hipótesis más probable es que no tenga cáncer. Los resultados de la aplicación de la

inferencia bayesiana dependen de los valores de las probabilidades a priori, que deben de estar disponibles para poder aplicar este método, además con este método las hipótesis no son completamente aceptadas o rechazadas, si no que son mas o menos probables, según los datos observados.

Si las hipótesis de partida se consideran distribuciones gaussianas, obtener el Maximum Likelihood (mas bien su logaritmo neperiano) es lo mismo que minimizar el error cuadrático medio, que es el exponente de las probabilidades cuando son distribuciones normales.

Esta conclusión puede ser diferente si las distribuciones de partida no son gaussianas.

Hay que hacer notar que la hipótesis ML no tiene por qué ser la misma hipótesis MAP, sólo coincidirá cuando las probabilidades de las hipótesis a priori sean idénticas.

5.3 Clasificadores bayesianos.-

5.3.1 Clasificador óptimo bayesiano.-

Supongamos que tenemos tres hipótesis h_1, h_2 y h_3 y las posibles clasificaciones son + y – de una nueva instancia.

Veamos un caso, sea la probabilidad a posteriori:
$P(h_1 | D) = 0.4$
$P(h_2 | D) = 0.3$
$P(h_3 | D) = 0.3$

$h_{MAP} = \text{argmax}_h P(h | D)$, por lo tanto la hipótesis MAP es la hipótesis h_1, la más probable (0.4) teniendo en cuenta los datos observados.

Hasta ahora hemos estudiado la búsqueda de la hipótesis más probable(MAP) dados los datos observados (de entrenamiento), pero nos vamos a centrar en la pregunta de la clasificación más probable de una nueva instancia dados los datos de entrenamiento.

Supongamos una nueva instancia, que se clasifica según las hipótesis h_1 como + y con h_2 y h_3 como -

$P(- | h_1) = 0$ $\quad$ $P(+ | h_1) = 1$
$P(- | h_2) = 1$ $\quad$ $P(+ | h_2) = 0$
$P(- | h_3) = 1$ $\quad$ $P(+ | h_3) = 0$

Así en el ejemplo anterior h_1 es la hipótesis MAP y por lo tanto la clasificación más probable sería +, ya que $P(+ | h_1) = 1$.

Ahora bien si se consideran todas las hipótesis, ponderadas por sus probabilidades a posteriori, de acuerdo con el teorema de la probabilidad total:

$$P(v_j \mid D) = \sum_{h_i \in H} P(v_j \mid h_i) * P(h_i \mid D)$$

Siendo v_j una posible clasificación, y h_i una de las hipótesis dentro del conjunto H de todas las hipótesis posibles.

La clasificación óptima para una nueva instancia será aquella v_j que cumple:

$$\text{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) * P(h_i \mid D)$$

Aplicado al ejemplo anterior:

v_j es + o -, que es la posible clasificación.

H_i son las posibles hipótesis, h_1, h_2 y h_3.

$$\sum_{h_i \in H} P(+ \mid h_i) * P(h_i \mid D) = 1*0.4 + 0*0.3 + 0*0.3 = 0.4$$

$$\sum_{h_i \in H} P(- \mid h_i) * P(h_i \mid D) = 0*0.4 + 1*0.3 + 1*0.3 = 0.6$$

Por lo tanto la clasificación óptima es -, justamente la contraria de la MAP que era +.

Cualquier sistema que clasifique nuevas instancias de acuerdo con la siguiente fórmula recibe el nombre de clasificador óptimo bayesiano:

$$\text{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) * P(h_i \mid D)$$

Este método maximiza la probabilidad de que una nueva instancia sea clasificada correctamente, dados los datos observados D, el espacio completo de hipótesis H ($h_i \in H$) y las probabilidades a priori de las posibles clasificaciones teniendo en cuenta las hipótesis ($P(v_j \mid h_i)$).

5.3.2 Clasificador naive bayesiano (nb).-

Supongamos que queremos clasificar una nueva instancia formada por varios atributos $a_1, a_2, .. a_n$. Sea v_j una de las clasificaciones dentro del conjunto V de las distintas clasificaciones posibles.

Sabemos que:

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}} \; P(v_j) * P(D \mid v_j)$$

En nuestro caso, $v_{MAP} = \underset{v_j \in V}{\text{argmax}} \; P(v_j) * P(a_1, a_2, \ldots a_n \mid v_j)$

Dados los datos de entrenamiento se recorren todos estos datos y se recuenta la clasificación de cada uno de ellos, obteniendo $P(v_j)$ para cada clasificación posible.

El problema surge cuando queremos obtener $P(a_1, a_2, \ldots a_n \mid v_j)$ ya que el número de posibles combinaciones diferentes de valores para cada atributo con todos los demás es muy grande.

Por la regla de la multiplicación:

$P(a_1, a_2, \ldots a_n \mid v_j) = P(a_n \mid v_j) * P(a_1, a_2, \ldots a_{n-1} \mid a_n, v_j)$.

Ahora bien si suponemos que los atributos son independientes condicionalmente conocido v_j, sabemos que $P(a_{n-1} \mid a_n) = P(a_{n-1})$ por lo tanto y generalizando, nos queda la fórmula:

$P(a_1, a_2, \ldots a_n \mid v_j) = P(a_1 \mid v_j) * P(a_2 \mid v_j) * \ldots P(a_n \mid v_j)$.

Quedando la fórmula inicial:

$v_{MAP} = \underset{v_j \in V}{\text{argmax}} \; P(v_j) * P(a_1, a_2, \ldots a_n \mid v_j)$.

De la siguiente manera:

$$v_{NB} = \underset{v_j \in V}{\text{argmax}} \; P(v_j) * \prod_i P(a_i \mid v_j)$$

Las letras NB se utilizan para expresar que se ha utilizado un clasificador Naive Bayesiano, avisando de la premisa fundamental de que los atributos son independientes condicionalmente.

Es decir, deben de estimarse los valores $P(v_j)$ y $P(a_i \mid v_j)$ partiendo de los datos observados.

En el algoritmo naive de clasificación bayesiano no se hace por lo tanto ninguna búsqueda en el espacio de posibles hipótesis, sino más bien en el recuento de los datos observados.

Un problema surge cuando la $P(a_i \mid v_j)$ es cero para algún atributo, en cuyo caso fuerza a que la multiplicación sea cero y por lo tanto dará siempre cero para esa hipótesis, para evitar este problema se introduce la denominada probabilidad m-estimada, cuya fórmula es la siguiente:

$P(a_i \mid v_j) = (N_c + 1) / (N + K)$ siendo

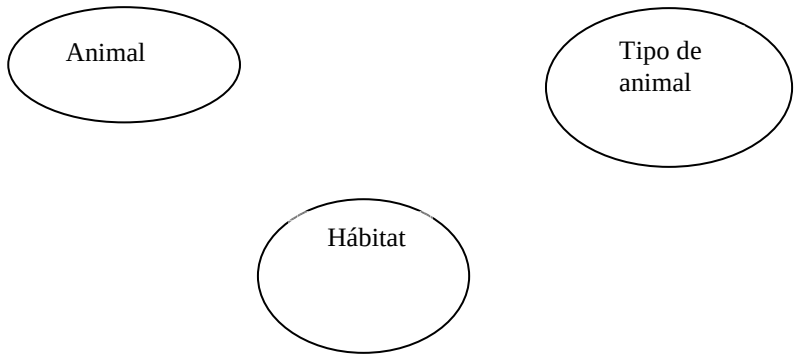
N el número de casos totales que tenemos con clasificación v_j.
N_c el número de casos en que se da el valor a_i con clasificación v_j.
K el número de valores diferentes que toma el atributo a_i con clasificación vj.

5.4 Dependencia/ independencia condicional.-

Una variable A y otra B son dice que son dependientes mutuamente, si saber lo que vale A me ayuda a conocer lo que vale B.

La dependencia entre A y B se dice que es una dependencia condicional, si A y B son dependientes si sé o no sé valores de otras variables.

Por ejemplo:



Si yo sé que el tipo de animal es mamífero, esta información me ayuda a adivinar que hábitat es tierra, luego tipo de animal y hábitat son dependientes mutuamente.

Ahora bien, si yo sé que animal es ballena, esto me ayuda a adivinar que hábitat es agua.

En este caso saber que tipo de animal es mamífero no nos ayuda para adivinar que hábitat es agua.

Es decir, que si yo sé el animal, la información sobre tipo de animal no me ayuda a saber el hábitat. Luego tipo de animal y hábitat no son dependientes mutuamente si yo sé el animal.

Mas formalmente diremos :

Sean X, Y y Z tres variables aleatorias discretas, se dice por definición que X es condicionalmente independiente de Y dada Z, si la distribución de probabilidad que gobierna X es independiente del valor de Y conociendo el valor de Z , es decir, si

P(X =x | Y = y, Z =z) = P(X =x | Z =z), para todo valor de x , y y z.

Por comodidad la definición anterior se escribe de la siguiente forma:

P(X | Y, Z) = P(X | Z).

Se puede ampliar esta definición al caso de varias variables aleatorias, por definición se dice que un conjunto de variables aleatorias X_1, ... X_h son condicionalmente independientes de un conjunto de variables aleatorias Y_1, ... Y_m dado el conjunto de valores Z_1, .. Z_n si

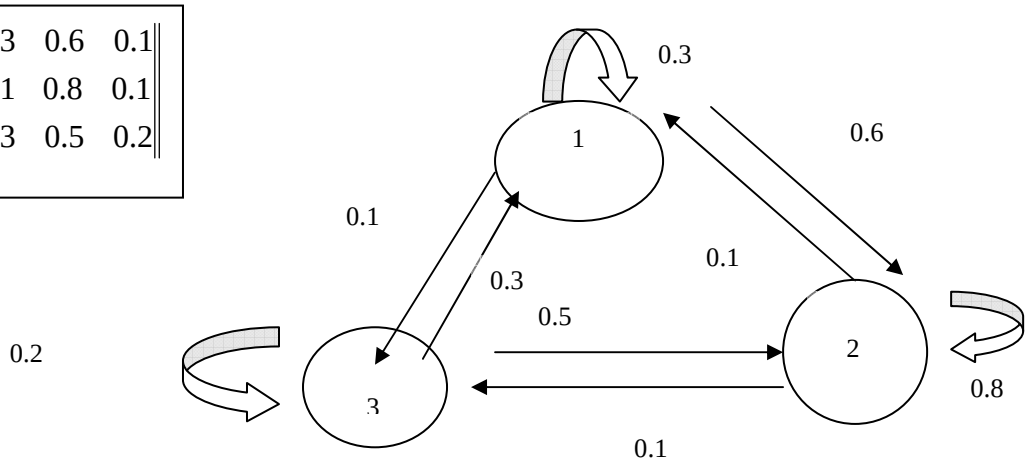
$$P(X_1, ... X_h | Y_1, ... Y_m, Z_1, ... Z_n) = P(X_1, .. X_h | Z_1, ... Z_n)$$

En el clasificador bayesiano sencillo se asume que existe independencia condicional entre el atributo A_i y el atributo A_j de la misma instancia conocido el valor de la clasificación v.

Ejemplo de las cadenas de Markov.-

El estado en el tiempo t sólo depende del estado en el tiempo t-1.

$$\begin{Vmatrix} 0.3 & 0.6 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.3 & 0.5 & 0.2 \end{Vmatrix}$$



P(x = 1, t | x =1 , t-1) = 0.3 por ejemplo, luego son probabilidades condicionadas.

5.5 Redes bayesianas.-

5.5.1 Introducción.-

La restricción, para la aplicación de la técnica del clasificador bayesiano sencillo, de que todos los atributos deben de ser independientes condicionalmente entre sí conocido el valor de la clasificación v_j, permite poder realizar la siguiente asignación.

$$P(a_1, a_2, \ldots a_n \mid v_j) = P(a_1 \mid v_j) * P(a_2 \mid v_j) * \ \ldots \ \ P(a_n \mid v_j).$$

Pero es una condición muy restrictiva para poderla aplicar a una gran cantidad de problemas en los que esto no se da, **buscando una solución intermedia entre el clasificador bayesiano sencillo y el óptimo, surgen las redes bayesianas** en las que la independencia condicional se exige entre un subconjunto de atributos y no entre todos.

Las redes bayesianas siguen la línea de intentar descomponer una distribución de probabilidad multivariada en varios productos de funciones de distribución, denominadas locales.

Si consideramos un conjunto de variables aleatorias $Y = \{Y_1, Y_2 \ldots Y_n\}$, en donde cada variable puede tomar un valor del conjunto de posibles valores $V(Y_i)$, se define como espacio joint del conjunto de variables Y al producto cartesiano $V(Y_1)$ x $V(Y_2)$ x ... x $V(Y_n)$. Por lo tanto cada elemento del espacio joint corresponde a una posible asignación de los valores de cada tupla de variables $Y_1, Y_2, \ldots Y_n$.

La función de distribución que se forma sobre el espacio joint se denomina distribución joint de probabilidad, especificando una red bayesiana la distribución joint de probabilidad para un conjunto de variables aleatorias.

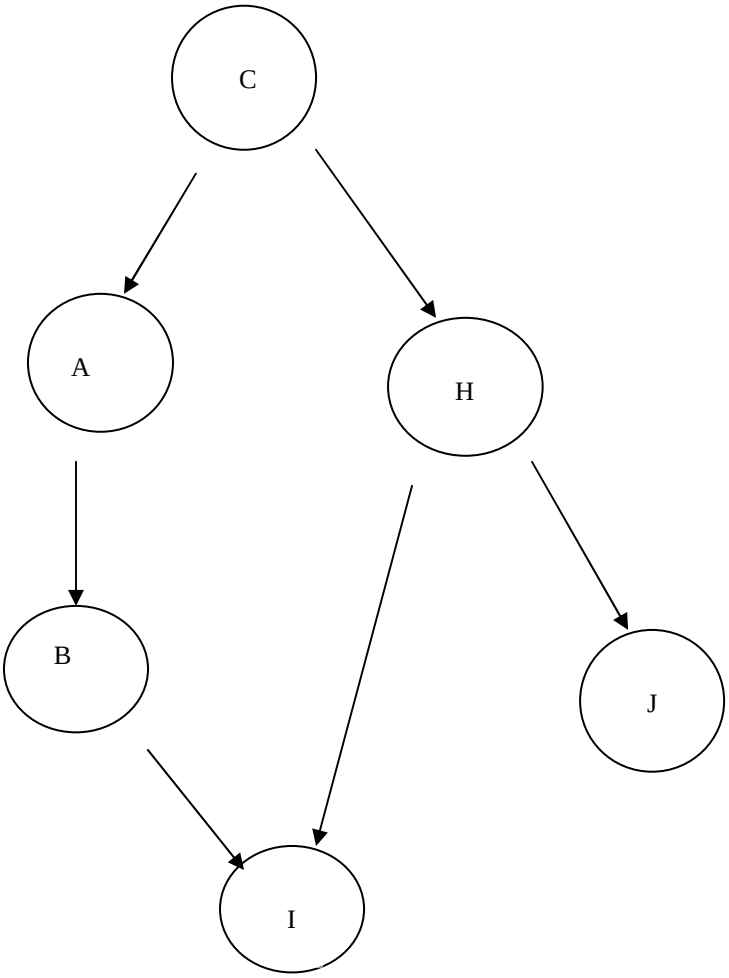
Hay **dos formas de enfocar** las dependencias / independencias:

- La **visión objetivista**, que consiste en A depende de B o no depende, no dejando nada aleatorio en esta dependencia.

- La **visión bayesiana,** se centra en la certidumbre que se tiene sobre la dependencia existente, por ejemplo A depende de B pero 0.9, es decir que A casi seguro que depende de B, según la evidencia.

Las probabilidades bayesianas manejan **probabilidades de estados**, por ejemplo que P(mantenimiento =caro) = 1, significa que estamos seguros de que la variable precio de mantenimiento está en el estado caro, teniendo en cuenta que según se vaya incorporando la evidencia estos valores se van a ir modificando. (ESTO ES precisamente el corazón del razonamiento bayesiano).

Veamos un ejemplo de red bayesiana:



Una red bayesiana tiene dos componentes principales: cualitativo y cuantitativo.

- En el campo cualitativo tenemos un **grafo acíclico dirigido** en el que cada nodo corresponde a un atributo (variable), y arcos dirigidos implicando que toda variable es condicionalmente independiente de todos sus no descendientes en la red siempre que se conozcan los valores de sus inmediatos predecesores (padres). Una variable Z es descendiente de otra variable Y, si en el grafo existe un camino dirigido desde Y a Z, por ejemplo en el grafo expuesto I es descendiente de A.

- En el campo cuantitativo cada nodo tiene asociada la **distribución de probabilidad de esa variable** teniendo en cuenta sus padres en el grafo. Por ejemplo para el nodo I, tendremos P(I | B,H).

Cada variable tiene un conjunto de posibles valores llamado espacio de estados que consiste de mutuamente exclusivos y exhaustivos valores de las variables.

La probabilidad joint de cualquier elemento sería:

$$P(y_1,...y_n) = \prod_{i=1}^{n} P(y_i \mid Padres(y_i))$$

Esto se conoce como chain-rule.

Padres (y_i) son las variables predecesoras inmediatas de la variable y_i en la red, precisamente $P(y_i \mid padres(y_i))$ son los valores que se almacenan en el nodo que corresponde a la variable y_i.

Una definición formal de una red bayesiana sería:

Una tripleta (N,D,P) en donde

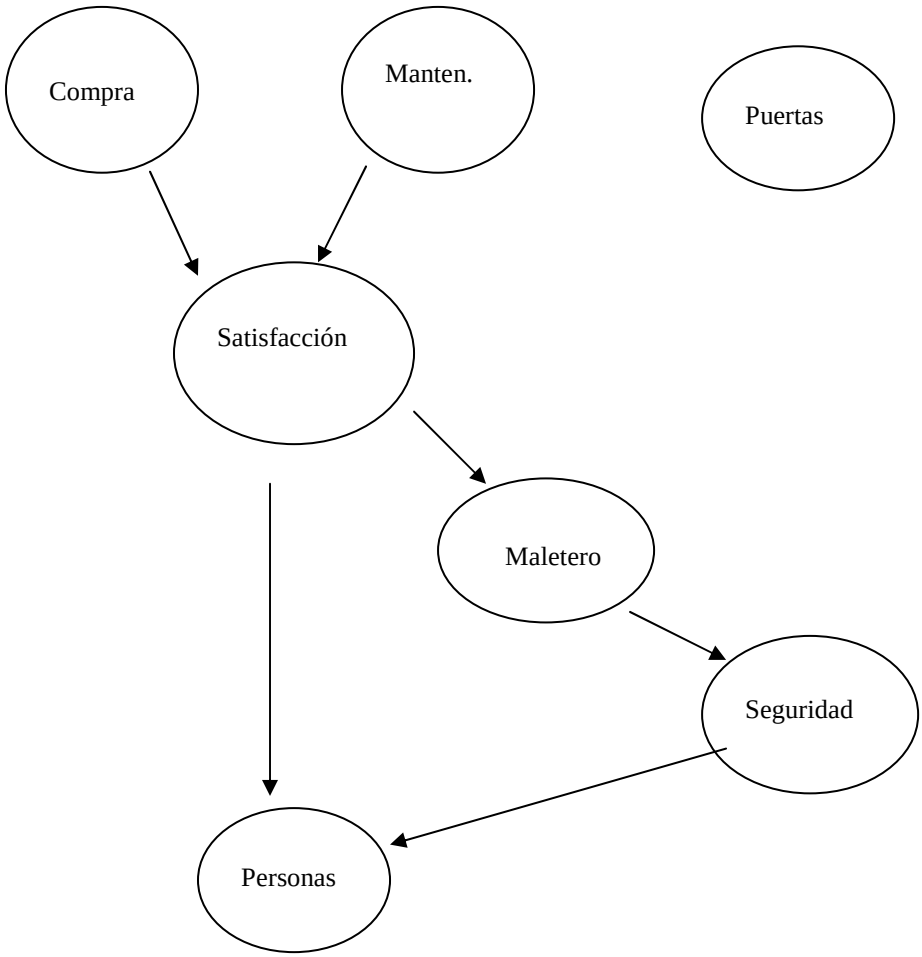
- N es un conjunto de variables del dominio.
- D es una DAG (Grafo acíclico dirigido) cuyos nodos están etiquetados con los elementos de N y los arcos dirigidos indican relación de influencia y en algunos casos relación causal.
- P es una distribución joint sobre N.

Veamos el caso de la Base de datos de coches (UCI REPOSITORY).

Variables de cada tupla:

- Puertas(2, 3, 4 , 5 o más).
- Personas(2, 4, más de 4).
- Seguridad (baja, media, alta).
- Maletero (grande, mediano, pequeño).
- Precio compra(muy alto, alto, medio, bajo).
- Coste mantenimiento (muy alto, alto, medio, bajo).
- Grado de satisfacción (muy bueno, bueno, satisfecho, insatisfecho).

Generada la red bayesiana se obtiene la red que figura en la página siguiente, de dicha red se puede inferir que conocido el precio de compra del vehículo y el coste de mantenimiento se puede dar perfectamente la función de distribución del grado de satisfacción del cliente, ya que el resto de variables no influyen en la satisfacción, es decir son condicionalmente independientes.



Por la regla de la multiplicación:

P(Compra=muy cara, Mantenimiento=caro, puertas =4, satisfacción =buena, maletero=grande, seguridad=alta, personas=4) =

P(Compra = muy cara) * P(mantenimiento=caro | compra = muy cara) * P(puertas=4 | compra= muy cara, mantenimiento=caro) * P(satisf = buena | compra =Muy cara, manten=caro,puertas=4) * P(malet=grande | satisf = buena, compra =Muy cara, manten=caro,puertas=4) * P(segu=alta | malet=grande, satisf = buena, compra =Muy cara, manten=caro,puertas=4) * P(personas=4 | segu=alta, malet=grande, satisf = buena, compra =Muy cara, manten=caro,puertas=4) =

P(compra=muy cara) * P(mant=caro) * P(puertas=4) * P(satis=buena | compra=muy cara, mant=caro) * P(malet=grande | satis=buena) * P(segu=alta | malet=grande, satif=buena) * P(personas=4 | satisf=buena, segu=alta)

Observando la red P(mant=caro | compra= muy cara) = P(mant = caro) , es decir mant y compra son independientes condicionalmente.

P(satisf = buena | compra =Muy cara, manten=caro,puertas=4) = P(satis=buena | compra=muy cara, mant=caro), luego conocidos los valores del precio de compra y del

precio de mantenimiento en la variable grado de satisfacción no influye el número de puertas, es decir satisfacción y número de puertas son independientes condicionalmente.

Y así se podría ir razonando sobre las conclusiones del grafo.

5.5.2 Obtención de redes bayesianas.-

Para construir una red bayesiana (modelo) es preciso definir las variables en primer lugar, a continuación obtener la estructura de la red y finalmente obtener las distribuciones de probabilidad locales.

Hay distintos enfoques para obtener la estructura de la red bayesiana, bien porque la red se conoce de antemano o bien porque se infiere de los datos de entrenamiento, y por otro lado si todas las variables que intervienen en la red son observables o bien si hay algunas que no lo son.

Casi todas las técnicas que se proponen trabajan con la técnica del gradiente [Rusell 1995], en cualquiera de los casos supone siempre la obtención del Maximum Likelihood(ML).

Recordamos que:

MAXIMUM LIKELIHOOD.- Aquel modelo que obtiene un máximo de probabilidad de obtener los datos observados.

$$\mathbf{argmax}_{\mathbf{M}}\ \mathbf{P(D|M)}$$

El número de modelos (redes bayesianas) diferentes que son posibles, se eleva de manera considerable en función del número de variables a considerar, se obtiene:

Para una variable sólo un modelo.
Para dos variables son dos modelos.
Para tres variables se obtienen 8 modelos posibles.
Para cuatro variables se obtienen 64 modelos, y si continuamos así podemos obtener como conclusión que el número de posibles candidatos a modelos como red bayesiana sería:

$$2^{\frac{n * (n-1)}{2}}$$

Ya que el número de modelos es realmente grande cuando las variables son muchas, **se impone un método de búsqueda** para la elección del mejor modelo ya que es un problema NP-Hard demostrado en [Chickering 1995].

5.5.3 Cálculo de la red bayesiana más probable.-

Sea M el modelo a considerar (red bayesiana) y sean D los datos de entrenamiento, entonces

P(M $\cap$ D) = P(M) * P(D | M) = P(D) * P(M | D), por la regla de la multiplicación.

P(M | D) = P(M) * P(D | M) / P(D).

Un problema surge cuando queramos calcular P(D), es decir la probabilidad de que los datos D se den. Para evitar este cálculo tan extraño y ya que estamos interesados en calcular qué modelo es más probable, realizamos la división de probabilidades:

$$\frac{P(M_1 \mid D)}{P(M_2 \mid D)} = \frac{P(M_1) * P(D \mid M_1)}{P(M_2) * P(D \mid M_2)}$$

Factor de Bayes

Ahora bien, se puede calcular P(M) para un modelo a priori, es decir antes de conocer los datos. Una tendencia consiste en dejar hablar a los datos, que sean ellos los que expresen qué modelo es más probable, es decir todos los modelos sean equiprobables inicialmente y por lo tanto $P(M_1) = P(M_2) = ...$, así nos quedará que la fórmula para poder comparar dos modelos es la siguiente:

$$\frac{P(M_1 \mid D)}{P(M_2 \mid D)} = \frac{P(D \mid M_1)}{P(D \mid M_2)}$$

Si el factor de Bayes es mayor que 1 se toma el modelo M_1, si es menor que 1 se toma el modelo M_2.

Como esta fórmula va a producir muchos productos de valores comprendidos entre 0 y 1, ya que son probabilidades, se suele tomar más bien el logaritmo del factor de Bayes.

$$\ln \frac{P(M_1 \mid D)}{P(M_2 \mid D)} = \ln P(D \mid M_1) - \ln P(D \mid M_2)$$

Entonces si el logaritmo neperiano del ML da mayor que cero se toma el modelo M_1, en caso contrario se toma el modelo M_2.

Nos debemos plantear el cálculo de P(D|M), es decir la probabilidad de obtener los datos partiendo de un modelo.

5.5.4 Cálculo de P(D|M) para una red bayesiana.-

5.5.4.1 Para v.a. con sólo dos valores.-

Supongamos v. a. X (con atributos X_1, X_2...X_n) con dos valores sólo (0,1).

Sea q la probabilidad de que haya un 1, es decir q va de 0 a 1.
Los estados serían entonces X = 1 y X = 0, siendo
q = P(X=1), y como sólo hay dos estados
1 – q = P(X=0)

$$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \int_0^1 P(x_1, ..., x_n \mid q) * f(q)\, dq$$

siendo f(q) la función de densidad de la variable aleatoria q.

$P(x_1, .. , x_n \mid q) = q^k * (1-q)^{n-k}$, ya que será la probabilidad de que haya k unos (éxitos) en n pruebas.

Ya que hay dos estados, tendremos q y 1-q como probabilidades de cada estado, y por lo tanto tendremos que utilizar una distribución con dos parámetros α y β.

Supongamos que la v. a. q en nuestro caso inicial sigue una distribución Beta(α,β) a priori, se toma así por varias razones:

- q va entre 0 y 1.
- Si la distribución a priori de q es Beta, la distribución a posteriori de q dados los datos D también es una distribución Beta, ya que la distribución Beta y la distribución Binomial pertenecen a la familia de distribuciones conjugadas, es decir si la distribución a priori es de la familia , entonces la distribución a posteriori también está en la familia.
- Cuando tenemos Beta(1,1) estamos refiriéndonos a la distribución uniforme, ya que la función de densidad será f(q) = 1.

Conforme el número de casos aumenta se va aproximando a una curva de gauss, ya que al aumentar el número la varianza disminuye.

Por lo tanto:

$$f(q) \;=\; \frac{1}{B(\alpha,\beta)}\, q^{\alpha-1}\,(1-q)^{\beta-1} \;=\; \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)}\;\; q^{\alpha-1}\,(1-q)^{\beta-1}$$

Sabemos que:

$$\Gamma(\alpha) \;=\; (\alpha-1)!$$

siempre que α sea entero.

$$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) \;=\; \int_0^1 q^{k} * (1-q)^{n-k}\, \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)}\;\; q^{\alpha-1}\,(1-q)^{\beta-1}\; dq \;=$$

$$\underbrace{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)}}_{\text{Factor}} \underbrace{\int_0^1 q^{k+\alpha-1}\,(1-q)^{n-k+\beta-1}\,dq}_{\text{Beta}(k+\alpha,\ n-k+\beta)}$$

Luego:

$$P(X_1 = x_1, X_2 = x_2, ...X_n = x_n) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)}\,\frac{\Gamma(k+\alpha)\,\Gamma(n-k+\beta)}{\Gamma(\alpha+\beta+n)}$$

Por lo tanto si a priori la v. a. q sigue una distribución Beta(α,β), a posteriori la v. a. q sigue una distribución Beta(k + α, n – k + β), siendo k el número de 1 y n el número de casos de la evidencia aportada.

Esto se puede ver como que hay a priori α éxitos y β fallos, y observados n casos más con k éxitos, tendremos una observación total de k + α éxitos con n – k + β fallos, aunque con un factor de corrección

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)}$$

Se puede ver también de la siguiente manera, tenemos en una bolsa n bolas de dos colores diferentes (rojo y amarillo) sin saber cuantas hay de cada color.

Tomamos dos cajas(la primera urna para color rojo y la segunda para color amarillo), inicialmente ponemos α bolas rojas en la urna uno y β bolas amarillas en la urna dos.

A continuación extraemos una bola de la bolsa y la depositamos en la urna de su color, y así continuamos hasta que las n bolas han sido depositadas en su urna correspondiente.

El conocimiento a priori de la persona que realiza el experimento se aporta poniendo α bolas rojas inicialmente en la urna uno y β bolas amarillas en la urna dos. Es decir comienza con unos valores a priori:

q = P(bola = roja) = α / (α+β).
1-q = P(bola = amarilla) = β / (α+β)

Sacar n bolas de la bolsa, comprobando su color y poniéndolas en la urna de su color corresponde a la aportación de la evidencia al estudio en cuestión.

La probabilidad a posteriori, después de sacar la primera bola de la bolsa y depositarla en su urna, supongamos que es roja:

q = P(bola = roja) = (α+1) / (α+β+1)
1-q = P(bola = amarilla) = β / (α+β+1)

Entonces q aumenta y 1-q disminuye.

Si después de repetir el proceso con las n bolas hemos tenido n_1 de color rojo y n_2 de color amarillo, la probabilidad a posteriori será:

q = P(bola = roja) = (α+n_1) / (α+β+n)
1-q = P(bola = amarilla) = (β+n_2) / (α+β+n)

Se puede observar que el orden de las cajas en este experimento no tiene ningún efecto.

Para el caso de Beta(1,1) a priori como distribución del vector de v. a. q y 1-q, es decir un caso cierto y un caso fallo, equiprobable entonces, nos dará como a posteriori Beta(k+1, n–k+1)

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \frac{\Gamma(1+1)}{\Gamma(1)*\Gamma(1)} \quad \frac{\Gamma(k+1)\,\Gamma(n-k+1)}{\Gamma(1+1+n)}$$

$$= \quad \frac{k!\,(n-k)!}{(n+1)!}$$

Ejemplo.-

X	1	1	1	1	1	1	1	1	0	0	0	0	0
Y	1	1	1	1	1	0	0	0	1	1	0	0	0

Supongamos a priori una distribución Beta(1,1), a posteriori será:

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \frac{k!\,(n-k)!}{(n+1)!}$$

Consideremos los dos modelos que hay a continuación:

Modelo M_1 Modelo M_2

Calculemos para los modelos M1 y M2 su P(D | M).-

Modelo M_1.-

$$P(x,y \mid M1) = P(x \mid M1) * P(y \mid M1) = \frac{8!\,5!}{14!} * \frac{7!\,6!}{14!} = 2.31*10^{-9}$$

Ya que las v. a. x e y son independientes según el modelo M_1.

Modelo M_2.-

$$P(x,y \mid M2) = P(x \mid M2) * P(y \mid x, M2) = P(x \mid M2) * P(y \mid x=0, M2) * P(y \mid x=1, M2) =$$
$$\frac{8!5!}{14!} * \frac{2!3!}{6!} * \frac{5!3!}{9!} = 1.84*10^{-9}$$

Comparativa de modelos.-

$$\frac{P(x,y \mid M_1)}{P(x,y \mid M_2)} = \frac{2.31*10^{-9}}{1.84*10^{-9}} > 1$$

Como el modelo M_1 da un valor más alto que el modelo M_2 nos quedaríamos con el primero.

También señalar que son probabilidades relativas, ya que el objetivo es encontrar una estructura buena, no estimar la probabilidad de una estructura.

La distribución Beta de la v. a. q tiene como esperanza matemática(E) y varianza(Var):

$$E[q] = \alpha / (\alpha+\beta).$$
$$Var[q] = \alpha\beta / ((\alpha+\beta)^2 (\alpha+\beta+1))$$

De estas fórmulas se deduce que la varianza a priori de q será:

$Var[q] = \alpha\beta / ((\alpha+\beta)^2 (\alpha+\beta+1))$, ya que es la función Beta.

Siendo la varianza a posteriori de q, después de los datos de evidencia:

$Var[q \mid D] = (k+\alpha)(n-k+\beta) / ((k+\alpha + n-k+\beta)^2 (k+\alpha+n-k+\beta+1))$,
ya que la función es Beta$(k+\alpha, n-k+\beta)$,

es decir

$Var[q \mid D] = (k+\alpha)(n-k+\beta) / ((\alpha + n+\beta)^2 (\alpha+n+\beta+1))$

De esta fórmula final se puede concluir que cuando n aumenta, es decir el número de casos que figuran en la evidencia aumenta se obtiene una precisión mayor en el valor de q.

Por lo tanto las redes bayesianas utilizan la evidencia para estimar un modelo bueno, es decir "corrige" el conocimiento a priori que teníamos.

5.5.4.2 Para v. a. con n valores.-

Veamos el caso para tres valores de las variables(0, 1 y 2).-

3 casos de 0,0	4 casos de 0,1	6 casos de 0,2
5 casos de 1,0	1 caso de 1,1	2 casos de 1,2
3 casos de 2,0	3 casos de 2,1	3 casos de 2,2

Al ser tres valores posibles de las variables deberemos utilizar la distribución una generalización de la distribución Beta que sólo tiene dos parámetros(α y β), dicha generalización recibe el nombre de distribución Dirichlet que tiene n parámetros, uno por cada posible valor de la variable.

p = P(X =0),
q = P(X =1)
1-p-q = P(X =2).

Necesitando entonces una distribución con tres parámetros α_1, α_2, α_3, teniendo en cuenta que tanto p como q como 1-p-q van entre 0 y 1.

Se puede ver también de la siguiente forma, tenemos en una bolsa n bolas de tres colores diferentes (rojo, amarillo y verde) sin saber cuantas hay de cada color.

Tomamos tres cajas(la primera urna para color rojo, la segunda para color amarillo y la tercera para el color verde), inicialmente ponemos α_1 bolas rojas en la urna uno y α_2 bolas amarillas en la urna dos y α_3 bolas verdes en la urna tres.

A continuación extraemos una bola de la bolsa y la depositamos en la urna de su color, y así continuamos hasta que las n bolas han sido depositadas en su urna correspondiente.

El conocimiento a priori de la persona que realiza el experimento se aporta poniendo α_1 bolas rojas inicialmente en la urna uno , α_2 bolas amarillas en la urna dos y α_3 bolas verdes en la urna tres. Es decir comienza con unos valores a priori:

p = P(bola = roja) = $\alpha_1 / (\alpha_1+\alpha_2+\alpha_3)$.
q = P(bola = amarilla) = $\alpha_2 / (\alpha_1+\alpha_2+\alpha_3)$.
1-p-q = P(bola = verde) = $\alpha_3 / (\alpha_1+\alpha_2+\alpha_3)$.

Sacar n bolas de la bolsa, comprobando su color y poniéndolas en la urna de su color corresponde a la aportación de la evidencia al estudio en cuestión.

La probabilidad a posteriori, después de sacar la primera bola de la bolsa y depositarla en su urna, supongamos que es roja:

p = P(bola = roja) = $(\alpha_1+1) / (\alpha_1+\alpha_2+\alpha_3+1)$.
q = P(bola = amarilla) = $(\alpha_2+1) / (\alpha_1+\alpha_2+\alpha_3+1)$.
1-p-q = P(bola = verde) = $(\alpha_3+1) / (\alpha_1+\alpha_2+\alpha_3+1)$.

Entonces p aumenta , q disminuye y 1-p-q disminuye.

Si después de repetir el proceso con las n bolas hemos tenido n_1 de color rojo, n_2 de color amarillo y n_3 de color verde, la probabilidad a posteriori será:

p = P(bola = roja | D) = $(\alpha_1+n_1) / (\alpha_1+\alpha_2+\alpha_3+n)$.
q = P(bola = amarilla | D) = $(\alpha_1+n_2) / (\alpha_1+\alpha_2+\alpha_3+n)$.
1-p-q = P(bola = verde | D) = $(\alpha_1+n_3) / (\alpha_1+\alpha_2+\alpha_3+n)$.

Se puede observar que el orden de las cajas en este experimento no tiene ningún efecto.

La distribución Dirichlet ($\alpha_1,\alpha_2,...\alpha_n$) es una generalización multivariada de la distribución Beta(α,β), se supone entonces que tenemos una variable aleatoria de n-dimensiones (también considerada como n v. a. estudiadas de forma conjunta) (q_1,q_2, ...,q_n) , q_i es la probabilidad de que la v. a. se encuentre en el estado i.

De tal forma que :

- Sólo valores de q_i comprendidos entre 0 y 1 tienen probabilidad.
- Deben de ser independientes q_1, q_2 ... q_n entre sí, y su suma debe de ser constante.
- Sólo aquellos valores de q_i para los que su suma es 1 tienen probabilidad. Ya que la suma de probabilidades de todos los estados debe de ser uno.
- La esperanza matemática(E) será $E[q_i] = \alpha_i / (\alpha_1 + ... + \alpha_n)$.
- Si $\alpha=\alpha_1+\alpha_2+ ... +\alpha_n$, entonces la varianza será

$$Var(X_i) = \frac{\alpha_i(1-\frac{\alpha_i}{\alpha})}{\alpha(\alpha+1)}$$, aunque tiene otras formas.

- Dirichlet(1,1, ... , 1) es la distribución uniforme.
- Si (q_1,q_2, ...,q_n) tiene una distribución Dirichlet($\alpha_1,\alpha_2,...\alpha_n$) , entonces q_1 tiene una distribución Beta(α_1, $\alpha_2+..\alpha_n$).
- La distribución Dirichlet($\alpha_1,\alpha_2,...\alpha_n$) tiene como función de densidad:

$$\frac{\Gamma(\alpha_1+\alpha_2+...+\alpha_n)}{\Gamma(\alpha_1)\Gamma(\alpha_2)...\Gamma(\alpha_n)} x_1^{\alpha_1-1}x_2^{\alpha_2-1}...x_{n-1}^{\alpha_n-1}(1-x_1-x_2-...-x_{n-1})^{\alpha_n-1}$$

Fijémonos qué parecido tiene con la distribución Beta:

$$f(q) = \frac{1}{B(\alpha,\beta)} q^{\alpha-1}(1-q)^{\beta-1} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)} q^{\alpha-1}(1-q)^{\beta-1}$$

- Si a priori es una distribución Dirichlet($\alpha1,\alpha2,\alpha3$), a posteriori será una distribución Dirichlet($\alpha1$ + n1, $\alpha2$ + n2, $\alpha3$ + n3), ya que las distribuciones Dirichlet y multinomial son de la misma familia conjugada.

$$P(X_1=x_1..X_n=x_n) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)*\Gamma(\alpha_2)*\Gamma(\alpha_3)} \frac{\Gamma(\alpha_1+n_1)\Gamma(\alpha_2+n_2)\Gamma(\alpha_3+n_3)}{\Gamma(N+\alpha_1+\alpha_2+\alpha_3)}$$

Para el caso de Dirichlet(1,1,1) en particular saldrá:

$$P(X_1 = x_1 \ldots X_n = x_n) =$$

$$\frac{\Gamma(3)}{\Gamma(1) * \Gamma(1) * \Gamma(1)} \quad \frac{\Gamma(1+\mathrm{n}_1)\,\Gamma(1+\mathrm{n}_2)\Gamma(1+n_3)}{\Gamma(3+N)} =$$

$$2 \quad * \quad \frac{\Gamma(1+\mathrm{n}_1)\,\Gamma(1+\mathrm{n}_2)\Gamma(1+n_3)}{\Gamma(3+N)} \quad = \quad 2 \quad * \quad \frac{n_1!\, n_2!\, n_3!}{(N+2)!}$$

Modelo M1.-

$$\mathrm{P(x, y \mid M1) = P(x \mid M1) * P(y \mid M1)} = 2 \quad * \quad \frac{13!\ 8!\ \ 9!}{32!} \quad * \quad 2 \quad * \quad \frac{11!\, 8!\, 11!}{32!}$$

Modelo M2.-

$$\mathrm{P(x, y \mid M2) = P(x \mid M2) * P(y \mid x = 0, M1) * P(y \mid x = 1, M2) * P(y \mid x = 2, M2)} =$$

$$2 \quad * \quad \frac{13!\ 8!\ \ 9!}{32!} \quad * \quad 2 \quad * \quad \frac{3!\, 4!\, 6!}{15!} * \quad 2 \quad * \quad \frac{5!\, 1!\, 2!}{10!} * \quad 2 \quad * \quad \frac{3!\, 3!\, 3!}{11!}$$

Debemos tener especial cuidado de no olvidarnos del factor de corrección.

Hemos mencionado que si un vector de v. a. tiene una distribución Dirichlet(α_1, α_2, ...α_n) se tiene como valor esperado para cada v. a. :

$E[q_i] = \alpha_i \ / \ (\alpha_1 + \ldots + \alpha_n)$.

Sea n el número de casos totales y sea n_i el número de casos en que la v. a. se encuentra en el estado i que se aportan en la evidencia:

$E[q_i] = (\alpha_i + n_i) \ / \ (\alpha_1 + \ldots + \alpha_n + n) =$

$$= \frac{\alpha_i}{\alpha_1 + \ldots + \alpha_n + n} \quad + \quad \frac{n_i}{\alpha_1 + \ldots + \alpha_n + n} \quad =$$

$$\frac{\alpha_1 + \ldots + \alpha_n}{\alpha_1 + \ldots + \alpha_n + n} * \frac{\alpha_i}{\alpha_1 + \ldots + \alpha_n} \quad + \quad \frac{n}{\alpha_1 + \ldots + \alpha_n + n} * \frac{n_i}{n}$$

Peso * A priori + (1-peso) * Datos

Es decir que se ven los pesos tanto del conocimiento a priori, como de la importancia que se da a los datos.

Por lo tanto la esperanza a posteriori es la suma de la esperanza a priori y de la media de los datos con su ponderación correspondiente.

Para la varianza (Var) obtendremos:

$Var[X_i] = (\alpha_i\alpha_1+\alpha_i\alpha_2 + \alpha_i\ \alpha_3 + ... +\alpha_i\alpha_n) / ((\alpha_1+\alpha_{2...} +\alpha_n)^2 (\alpha_1+\alpha_{2...} +\alpha_n+1))$

Excepto para $\alpha_i\alpha_i$

Otra forma más sencilla, pero idéntica es haciendo $\alpha_1+\alpha_2+...+\alpha_n = \alpha$

$Var[X_i] =\alpha_i *(1-\alpha_i/\alpha) / (\alpha(\alpha+1))$

Ya que conforme el número de datos de evidencia va ascendiendo (α es mayor) la varianza va siendo menor implica que el razonamiento es consistente ya que converge y la estimación a priori va teniendo menos efecto en los resultados.

5.5.4.3 Métrica BD y K2.-

El documento que ha servido de base para estos cálculos es [Heckerman 1995], en este documento se obtiene una fórmula muy similar a la anterior, denominada Métrica BD (Bayesian Dirichlet)

$$P(D \mid M) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\frac{N'}{q_i})}{\Gamma(\frac{N'}{q_i} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\frac{N'}{r_i q_i} + N_{ijk})}{\Gamma(\frac{N'}{r_i q_i})}$$

i recorre cada variable.
j recorre cada combinación posible de los padres de i.
r_i es el número de diferentes valores de la variable i en cada combinación de sus padres.

La fórmula que obteníamos en la página anterior era:

$$P(X_1 = x_1 .. X_n = x_n) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma(\alpha_1)*\Gamma(\alpha_2)*\Gamma(\alpha_3)} \quad \frac{\Gamma(\alpha_1+n_1)\Gamma(\alpha_2+n_2)\Gamma(\alpha_3+n_3)}{\Gamma(N+\alpha_1+\alpha_2+\alpha_3)}$$

Se puede realizar comparaciones entre estas dos fórmulas:

N'/r_iq_i sería α_i
N'/q_i sería $\alpha_1+\alpha_2+ ... +\alpha_n$

Luego N' = $(\alpha_1+\alpha_2+ ... +\alpha_{n)} * q_i$, siendo q_i el número de combinaciones diferentes de los padres de i.

N’ es el tamaño del ejemplo equivalente, se dice que si N’ es muy grande entonces pensamos que los datos no deben de influir rápidamente en las probabilidades. Por lo tanto, escoger un valor diferente puede llevarnos a probabilidades diferentes y por lo tanto a redes bayesianas diferentes.

Un caso especial de la métrica BD es hacer $N'/r_i q_i = 1$,como se propone en [Cooper 1992], recibiendo el **nombre de métrica K2.**

5.5.5 Búsqueda para la selección del mejor modelo.-

El mejor modelo es aquél que es el más probable. La teoría bayesiana nos proporciona herramientas para calcular la probabilidad de un modelo, como hemos visto en los apartados anteriores, y posteriormente comparar las probabilidades de varios modelos, detectando cual es el más probable.

Si bien es tan grande (más bien astronómica) la cantidad de modelos existentes que un recorrido exhaustivo por todos los modelos buscando aquél más probable llega a ser una utopía, por ello se impone una búsqueda heurística que nos lleve a encontrar datos válidos en un tiempo razonable.

Un método sencillo consistiría en escoger un modelo aleatoriamente, obtener su ML (maximum likelihood). Se escoge otro modelo de forma aleatoria, se comparan sus ML y nos quedamos con aquél que tenga mejor valor, y así sucesivamente . Como no tiene punto final deberemos indicarle por ejemplo que pasado un tiempo finalice, o cuando llegue a un valor umbral mínimo de probabilidad.

Un método más sofisticado es el que se describe a continuación, denominado algoritmo K2.

Las distribuciones locales dados los padres serán Dirichlet(1,1..1).

Paso 1.- Grafo inicial sin arcos.
Paso 2.- Elección de un arco para añadir al grafo.
    Paso 2.1.- Calcula la probabilidad de la nueva red con un arco nuevo en cada caso.
    Paso 2.2.-Escoge el arco que da probabilidad mayor.
Paso 3.- Si el arco nuevo aumenta la probabilidad de la nueva red , se añade y se va al paso 2, en caso contrario Fin, esa es la red.

Es aquí donde, en mi opinión, se puede introducir la técnica de algoritmos genéticos para obtener el modelo más probable, necesitando un métrica que sea una aplicación del conjunto de modelos en el conjunto de los números reales.

5.5.6 Inferencia en redes bayesianas.-

Inferencia se refiere a obtener conclusiones basadas en premisas, es decir basada en una nueva información, permitiendo realizar predicciones en caso de intervenciones que se hagan en base a las nuevas probabilidades.

Las relaciones presentadas en redes bayesianas no son necesariamente relaciones causales (causa-efecto), esto es debido a la posible existencia de lo que se denomina variables latentes, es decir variables que no se han incluido en el estudio y que pueden provocar relaciones ficticias de dependencia.

Inferencia, también denominada inferencia probabilística en una red bayesiana se basa en la noción de propagación de la evidencia, refiriéndose al proceso de cálculo de funciones de distribución a posteriori joint una vez tenidos en cuenta los valores de las variables observadas.

El razonamiento bayesiano proporciona una aproximación probabilística a la inferencia, se basa en que las cantidades de interés están gobernadas por funciones de distribución y que las decisiones óptimas pueden ser razonadas sobre estas probabilidades junto con los datos observados, proporcionando una aproximación cuantitativa a la evidencia, consistiendo en diversas hipótesis.

Aprovechando las redes bayesianas vamos a inferir el valor de alguna variable dando los valores observados de otras. Como estamos trabajando con variables aleatorias no se inferirán valores concretos , sino más bien funciones de distribución para esa variable, que señalará la probabilidad de que esa variable tome cada uno de los posibles valores partiendo de los valores observados de otras variables.

Para el procedimiento de propagación de la evidencia, es decir, cómo se propaga la información entre las diferentes variables de la red bayesiana, se han desarrollado varios métodos, en especial el propuesto en [Pearl 1986].

Ahora bien, la solución que planteó sólo sirve para cuando se aplica a redes simplemente conectadas (llamadas polytrees), es decir sin loops, por lo tanto no debe de existir ciclos sin tener en cuenta la dirección de las flechas.

Hay dos puntos de vista: si el interés radica en obtener la función de distribución para alguna variable, conocidas otras (conocido como sum propagation), o bien si estamos interesados en obtener la configuración más probable de todas las variables y no estamos interesados en la obtención para variables concretas (conocido como max propagation).

Las inferencias pueden ser de varios tipos:

- Diagnóstica. De efectos a causas.
- Causales. De causas a efectos.
- Intercausales. Entre causas de un efecto común.
- Mixtas. Combinación de dos o más de las anteriores.

6.- PROYECTO DE INVESTIGACIÓN.-

6.1 Introducción.-

El modelo va a tener como **función principal** el estudio de la relación de dependencia de parámetros, sobre todo aquellos que influyen en el parámetro TIRO CM desarrollado en esta página y la **forma** adoptada es la red bayesiana.

Comenzó el proyecto con el estudio de cada uno de los parámetros que figuraban en los datos, sobre todo la detección de si eran parámetros de entrada, o bien de salida. Este detalle es fundamental desde mi punto de vista ya que va a delimitar claramente el trabajo diferente con cada variable dependiendo del tipo en que sea encuadrada.

Un parámetro se entiende que es de entrada, en este estudio, cuando los valores son introducidos por el usuario en la fábrica con total libertad.

Un parámetro se clasifica de salida, cuando el usuario no actúa directamente sobre el parámetro, sino que el valor que tiene es resultado de la manipulación de otros, y sólo se puede leer su valor directamente.

Los parámetros de entrada se marcan con (E) y los de salida con (S)

Tiro en HF (S)
Indica el tiro que tenemos en el Horno Flash.

Tiro en caldera (S)
Indica el tiro que tenemos en la caldera.

Tiro CM (S)
Este parámetro es el que queremos optimizar. Se trata de conseguir que siempre haya tiro en la cámara de mezcla y que éste sea lo más estable posible. Valores de consigna buenos podrían ser: Alimentación HF de 140 aprox., 2 convertidores soplando a 35000, Tiro en CM de –25 (Tomar como muy malos los valores mayores de –18) . El tiro en el electrofiltro no debe pasar de 500.

Aspiración S1 (E)
Indica la aspiración de la soplante de la línea 1 de convertidores.

Aspiración S2 (E)
Indica la aspiración de la soplante de la línea 2 de convertidores.

Caudal S1 (E)
Caudal de la soplante 1 de convertidores. Los caudales nos dan una idea de las revoluciones a las que trabajan las soplantes.

Caudal S2 (E)
Caudal de la soplante 2 de convertidores.

Soplado C1 (E)
Indica la cantidad de aire que se está inyectando al convertidor 1.

Ángulo C1 (E)
Indica la posición en la que se encuentra el convertidor 1. Cuando está en campana, el ángulo estará próximo a 0.

Soplado C2 (E)
Indica la cantidad de aire que se está inyectando al convertidor 2.

Ángulo C2 (E)
Indica la posición en la que se encuentra el convertidor 2. Cuando está en campana, el ángulo estará próximo a 0.

Soplado C3 (E)
Indica la cantidad de aire que se está inyectando al convertidor 3.

Ángulo C3 (E)
Indica la posición en la que se encuentra el convertidor 3. Cuando está en campana, el ángulo estará próximo a 0.

Soplado C4 (E)
Indica la cantidad de aire que se está inyectando al convertidor 4.

Ángulo C4 (E)
Indica la posición en la que se encuentra el convertidor 4. Cuando está en campana, el ángulo estará próximo a 0.

Tiro salida electro filtro (S)
Indica el tiro a la salida del electro filtro. Debe ser <= 500.

Rpm FFC101 (E)
Sirve para regular la alimentación del HF.

Rpm FFC102 (E)
Sirve para regular la alimentación del HF.

Alimentación HF (S)
Es un campo que se calcula a partir de Rpm FFC101 y Rpm FFC102. Se consigue una alimentación de HF determinada manipulando los dos parámetros anteriores.

% Válvula dilución P1 (E)—DIRTM1
Indica el % de apertura de la válvula de dilución de la planta 1.

Cola P1 (S)—AV8090
Indica las emisiones de la planta 1 que se lanzan a la atmósfera.

% SO2 Entrada P1 (S)-- AV8095
Indica el % de SO2 que hay a la entrada de la planta 1.

Caudal P1 (E)
Indica el caudal (y por tanto, las revoluciones) de la soplante de la planta 1.

% Válvula dilución P2 (E)-- DIRTM2
Indica el % de apertura de la válvula de dilución de la planta 2.

Cola P2 (S)
Indica las emisiones de la planta 2 que se lanzan a la atmósfera.

% SO2 Entrada P2 (S)-- AV8210
Indica el % de SO2 que hay a la entrada de la planta 2.

Caudal P2 (E)-- AV8208
Indica el caudal (y por tanto, las revoluciones) de la soplante de la planta 2.

% Válvula dilución P3 (E)-- 3DKD051P
Indica el % de apertura de la válvula de dilución de la planta 3.

Cola P3 (S)-- Q3Q061
Indica las emisiones de la planta 3 que se lanzan a la atmósfera.

% SO2 Entrada P3 (S)-- Q3Q100M
Indica el % de SO2 que hay a la entrada de la planta 3.

Caudal P3 (E)-- F3F100
Indica el caudal (y por tanto, las revoluciones) de la soplante de la planta 3.

A continuación se calculó la matriz de correlación de los 32 parámetros, para saber si existía relación lineal entre los parámetros.

Se representa únicamente la fila segunda que corresponde con el parámetro Tiro CM, que es el importante en nuestro estudio:

-0.0425237; 1.00000000; -0.0443718; -0.0344702; -0.0296574;
-0.0243756; -0.0523369; -0.0426560; -0.0173607; -0.0247896;
-0.0501547; -0.0427716; -0.0349729; 0.03134123; -0.0494441;
-0.0349340; -0.0274894; -0.0538769; -0.0517187; -0.0424875;
0.28658231; -0.0247982; -0.0426576; 0.02877684; -0.0384841;
-0.0249813; -0.0539359; 0.91725690; -0.0337070; -0.0588332;
-0.0445656; -0.0425012

Se puede observar una relación lineal entre este parámetro (que ocupa la posición dos) y el que ocupa la columna 28, que corresponde al parámetro F3F100.

6.2 Discretización.-

6.2.1 Caso general.-

Los algoritmos genéticos son una técnica de optimización, perteneciente a la Inteligencia Artificial, realizando una simulación de las generaciones de individuos, mediante la reproducción, mutación y supervivencia de los más adaptados.

Dentro del aprendizaje automático (rama de la Ingeniería del conocimiento) hay dos grandes grupos: técnicas supervisadas (aquellas en las que los datos están clasificados previamente) y las técnicas NO supervisadas (en las que no existe clasificación previa).

El planteamiento se basa en el hecho de que los AG. sirven como optimizadores de funciones (minimizándolas) utilizando técnicas heurísticas, aunque tienen el inconveniente de su coste computacional grande y por lo tanto durante la fase de entrenamiento no se podrían utilizar en tiempo real.

Veámoslo aplicado al caso de h atributos en cada tupla.

Sea Z = <Z_i> el conjunto de datos observados desde i =1 hasta t, es decir hay t instancias observadas, siendo cada instancia Z_i =(A_i ... X_i) una tupla con h atributos diferentes.

Supongamos que los datos observados han sido generados por m distribuciones normales de dimensión h cada una.

La probabilidad de que un dato Z_i haya sido generado por la j distribución normal (denominada n_j) h-dimensional será:

$$P(Z_i \mid n_j) = \frac{1}{|M|^{\frac{1}{2}} (2\pi)^{\frac{h}{2}}} \; e^{\frac{-1}{2}(Z_i - u_j)' M^{-1} (Z_i - u_j)}$$

Siendo M la matriz de varianzas-covarianzas y u_j el vector de medias de la j distribución normal.

La matriz de varianzas-covarianzas (M) es la matriz cuadrada simétrica que tiene en la diagonal principal las varianzas de las observaciones y fuera de ella las covarianzas entre variables.

$$M = \begin{bmatrix} S_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & S_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & S_3^2 \end{bmatrix}$$

Siendo

$$\sigma_{ij} = \frac{\sum (X_i - \overline{X})(Y_j - \overline{Y})}{n}$$

| M| es el determinante de la matriz de covarianzas.

M^{-1} es la matriz inversa de M, recordando que sólo se puede calcular la inversa de una matriz si su determinante es distinto de cero (es decir es una matriz regular, no singular).

$$M^{-1} = \frac{1}{|M|} adj(M)'$$

M' es la traspuesta de M, es decir la matriz resultante de cambiar ordenadamente las filas por las columnas.

Las hipótesis iniciales (hipot) serían las m distribuciones normales h-dimensionales, caracterizada cada una por el vector de medias de dimensión h.

Nuestro interés radica en buscar la hipótesis de máxima verosimilitud, conocida como likelihood, recordamos del apartado conceptos básicos de Redes bayesianas desarrollado previamente que:

$$\text{hipot}_{ML} = \text{argmax}_{\text{hipot}} \ln P(Z|\text{hipot})$$

Siendo Z los datos observados.

$$hipot_{ML} = \arg\max_{n_j} \ln P(Z \mid n_j) = \arg\max_{n_j} \ln \prod_{i=1}^{t} P(Z_i \mid n_j)$$

Se supone que las t tuplas son mutuamente independientes para poder aplicar el paso anterior.

Aplicando la fórmula de la página anterior:

$$hipot_{ML} = \text{argmax}_{n_j} \sum_{i=1}^{t} \ln P(Z_i \mid n_j) = \text{argmax}_{n_j} \sum_{i=1}^{t} \ln \frac{1}{|M|^{\frac{1}{2}} (2\pi)^{\frac{h}{2}}} e^{\frac{-1}{2}(Z_i - u_j)' M^{-1} (Z_i - u_j)} =$$

$$\arg\max_{n_j} \left(\sum_{i=1}^{t} \ln \frac{1}{|M|^{\frac{1}{2}} (2\pi)^{\frac{h}{2}}} - \frac{1}{2} \sum_{i=1}^{t} (Z_i - u_j)' M^{-1} (Z_i - u_j) \right)$$

Dado que el primer término es constante cuando se modifica el vector de medias u_j, nos queda:

$$hipot_{ML} = \arg\max_{n_j} -\sum_{i=1}^{t}(Z_i - u_j)' M^{-1} (Z_i - u_j) =$$

$$\arg\min_{n_j} \sum_{i=1}^{t}(Z_i - u_j)' M^{-1} (Z_i - u_j)$$

Los métodos conocidos derivan e igualan a cero para obtener un mínimo, es decir utilizan la técnica del gradiente, de sobra es conocido que estas técnicas adolecen del grave problema de que garantizan un mínimo, pero local, en ningún momento aseguran un mínimo global.

En este sentido se aprovecha la técnica de los algoritmos genéticos que realizan una búsqueda heurística, minimizando una función, que en nuestro caso vendrá dada por la fórmula:

$$hipot_{ML} = \arg\min_{u_j} \sum_{i=1}^{t}(Z_i - u_j)' M^{-1} (Z_i - u_j)$$

En donde :

M^{-1} es la inversa de la matriz de varianzas-covarianzas de los atributos.
u_j es el vector de medias de una distribución h-dimensional.
Z_i es el vector de cada instancia o tupla.
$(Z_i - u_j)'$ es la traspuesta de la matriz $(Z_i - u_j)$.
t es el número de instancias (tuplas) observadas.

6.2.2 Aplicación para un atributo.-

Veamos aplicado el apartado anterior al caso de un atributo únicamente, a través de la función:

$$hipot_{ML} = \arg\min_{u_j} \sum_{i=1}^{t}(Z_i - u_j)' M^{-1} (Z_i - u_j)$$

Dado que las hipótesis están caracterizadas por los valores de las medias de las distribuciones gaussianas, cuando se aplica a un atributo nos dará el valor de la media que tiene mayor probabilidad de haber generado un dato en concreto y por lo tanto se realiza una discretización de los datos de partida.

El atributo escogido para estos estudios es el denominado 3DKD5-1P.

Un elemento de la población sabemos que será una posible solución, es decir en el caso que tenemos ahora de un atributo en cada tupla, será un vector de medias de distribuciones normales, y partiendo de un número de cluster (agrupamientos) iniciales "m" de 30, el vector de medias será de dimensión 30.

El número de elementos iniciales en la población es de 100.

Se tomará como desviación típica constante igual a 2 (Este valor no influye al ser un parámetro único a considerar).

Población inicial.-

Se realiza un recorrido de todos los datos X_i, detectando el mínimo y el máximo.

Se divide dicho rango en m intervalos (siendo m el número de cluster) de idéntico tamaño.

Para cada valor del vector de medias se calcula, desde j = 1 hasta m:

$$u_j = \min + \frac{\max - \min}{m} * j$$

Cada elemento hasta completar los 100, se genera de idéntica forma , a excepción de que se desplaza

$$\frac{\max - \min}{m*n} * i$$

Variando i desde 1 hasta 100, para completar la población inicial.

De esta forma se obtiene una población inicial de 100 elementos posibles soluciones, ya que se encuentran en el rango de los valores de las medias.

Utilizando este método para inicializar la población se obtiene un valor 100 veces mejor en la solución inicial al problema, que dejándolo totalmente arbitrario.

Función de evaluación.-

Como ya se ha indicado anteriormente será:

$$\operatorname{argmin}_{u_j} \sum_{i=1}^{t} (Z_i - u_j)' M^{-1} (Z_i - u_j)$$

Método de reemplazo.-

Se ha optado por una estrategia elitista, en la que los 4 elementos mejores de los n de la población, es decir los 4 que tienen una función de evaluación menor pasan directamente a la población de la siguiente generación.

El motivo es claro, ya que todo método de reemplazo que utiliza la técnica elitista converge, consiguiendo que la aptitud del mejor de la población sea siempre igual o superior que la correspondiente de la generación anterior.

Método de selección.-

Ya que se ha utilizado la estrategia elitista con los cuatro mejores, se seleccionan para aplicarles los operadores genéticos el resto de la población.

Operador genético de cruce.-

Para cada elemento de la población seleccionado se intercambia una media elegida al azar con la correspondiente de otro elemento de la población escogido aleatoriamente.

Operador genético de mutación.-

El operador de mutación se aplica a todos los elementos seleccionados, incrementando o decrementando aleatoriamente el valor de la media de una distribución normal elegida al azar.

La cantidad elegida es del uno por ciento del valor del dato, por lo tanto no hace falta normalizar los datos.

Operador genético de inversión y complemento.-

Ya que un individuo de la población está formado por m valores de medias, invertir el orden de las medias por la mitad por ejemplo, o bien mutar las posiciones, no tienen ningún efecto real sobre las posibles soluciones. Luego estos dos tipos de operadores no se han aplicado en la resolución del problema.

Criterio de finalización.-

El proceso acaba bien cuando se llega a un umbral de 0.01, o bien cuando después de 20 generaciones no se consigue mejorar la solución del problema.

Solución final seleccionada. (OBTENCION DEL CLUSTER).-

Una vez alcanzado uno de los motivos de que el proceso de generaciones acabe, se escoge al individuo que da una función de evaluación mejor.

Para cada Z_i y con el fin de asignarlo a un cluster se recorre las m distribuciones normales, detectando aquélla que da un likelihood mayor:

$$\arg \min_{u_j} \sum_{i=1}^{t} (Z_i - u_j)' M^{-1} (Z_i - u_j)$$

Esta media u indicará la que tiene una probabilidad mayor de que el dato Z_i haya sido generado por la distribución que tiene esa media, si una clase está caracterizada por el valor medio, se podrá decir que este valor pertenecerá a esa clase.

Una vez realizada la clasificación de los datos, se genera una salida con el valor de la función de evaluación alcanzado.

Si el número de datos asignado es cero este cluster no se tiene en cuenta, en caso contrario se obtiene la media de la distribución normal, así como el número de puntos que pertenecen a ella.

Además se obtiene en un fichero la asignación de cada valor al cluster correspondiente, y dado que es el valor de la media de la distribución, se realiza una discretización de los valores continuos que tenemos.

Número máximo de cluster.-

Dado que si la distribución normal no tiene asignado ningún punto no se tiene en cuenta, el número de cluster(grupos) iniciales es el número máximo de agrupamientos, por lo tanto al algoritmo sólo debe de dársele este valor, sin tener que conocer cuantos hay en realidad.

El algoritmo utilizado resumido es el siguiente:

```
Algoritmo genético;
inicializardatos(datos);
crear_y_cargar_ población inicial;
evaluar(población,datos);
vueltas := 0;
es_solucion(población,vueltas);
mientras (no objetivo) y (vueltas_sin_mejorar <20) hacer
        inicio
        elite(población);
        selección(población);
        cruce(población);
        mutación(población);
        evaluar(población,datos);
        es_solucion(población,vueltas);
        fin;
escribir_solucion(población,datos);
falgoritmo
```

6.2.3 Ajuste de parámetros.-

Numero de elementos de la población.-

Se hicieron pruebas con el algoritmo variando el número de elementos de la población con 50, 100 y 150 elementos , los resultados fueron los siguientes:

Elementos de la población	Función de evaluación	Tiempo
50	0.43	2 m. 7 seg.
100	0.50	3 m.
150	0.43	5 m. 30 seg.

De las pruebas realizadas se observa que con 50 y 150 elementos, se obtiene un valor para la función de evaluación similar, sin embargo el tiempo que tarda en llegar a su finalización es muy superior, más del doble del tiempo.

Se recuerda que el objetivo consiste en obtener una solución con el mínimo de valor para la función de evaluación.

Por todo lo cual se puso como ajuste del número de elementos de la población a un valor de 50.

Número de cluster iniciales (máximo).-

Se hicieron pruebas con el algoritmo variando el número de cluster iniciales con 20, 30 y 40, los resultados fueron los siguientes:

Cluster iniciales	Función de evaluación	Tiempo	Cluster obtenidos
20	0.60	2 m.	17
30	0.43	2 m. 7 seg.	21
40	0.21	2 m. 7 seg.	21

Se observa que con un número de cluster iniciales mayor se obtiene una evaluación mejor, sin embargo el tiempo de ejecución es muy similar y además, mención especial consiste en reseñar que con 30 cluster iniciales se obtienen 21 cluster finales, el mismo número que si los cluster iniciales fueran 40, por lo tanto aumentar el número de agrupamiento iniciales no afecta al número de cluster finales que se obtienen.

Por todo ello, se ha decido utilizar 40 como número de cluster iniciales.

Si el número de cluster finales fuera muy próximo a 40, sería un indicativo de que el número de estados diferentes es superior a este valor, debiendo aumentar el valor del número de cluster iniciales, que no ofrece ninguna dificultad ya que es una constante en el programa desarrollado.

Selección.-

Se han realizado pruebas haciendo variar el número de elementos seleccionados para la élite de 4 a 1, y además se incluye una selección con réplica consistente en realizar los operadores de mutación y cruce también al elemento seleccionado para la élite además de pasarlo directamente a la siguiente generación para mantener la convergencia del algoritmo, siendo los resultados los siguientes:

Selección	Función de evaluación	Tiempo
Élite de 4 elementos	0.21	2 m. 7 seg.
Élite de un elemento	0.21	2 m.
Élite de un elemento con réplica	0.0935	2 m.

Dados los resultados hay que mencionar que aplicar élite con un elemento con réplica consigue un valor de la función de evaluación de 250% mejor que si no se aplicara réplica al elemento seleccionado para la élite. Si además se observa que no aumenta el tiempo de ejecución el ajuste adoptado será claramente élite de un elemento con réplica del mismo.

Operador genético de mutación.-

Se han realizado pruebas haciendo que los valores de las medias variaran un 1 por mil, 1 y 5 por ciento, siendo los resultados los siguientes:

Mutación	Función de evaluación	Tiempo
1 %	0.0935	2 m.
5 %	0.1097	4 m.
1 ‰	0.0839	10 m.

Mención hay que realizar al hecho de que variar del 1 al 5 por ciento no supone una mejora en el valor de la función a minimizar, siendo el tiempo que tarda en obtener resultados superior incluso.

Si variamos los valores de las medias un uno por mil el tiempo se dispara a cinco veces superior, no obteniendo en mi opinión una mejora sustancial en el valor a minimizar.

Debido a todo esto se ha optado por el operador de mutación del 1 por ciento.

Operador genético de cruce.-

Se han realizado pruebas consistentes en cambiar el valor de una media de un elemento de la población, cambiar 4 medias y cambiar 10 medias entre cluster, siendo los resultados siguientes:

Cruce	Función de evaluación	Tiempo
Un valor de media	0.0935	2 m.
Cuatro medias	0.0946	2 m. 45 seg.
Diez medias	0.0969	2 m. 45 seg.

De los resultados se puede indicar que la función de evaluación no alcanza mejores valores aunque son muy similares, pero el tiempo es superior.

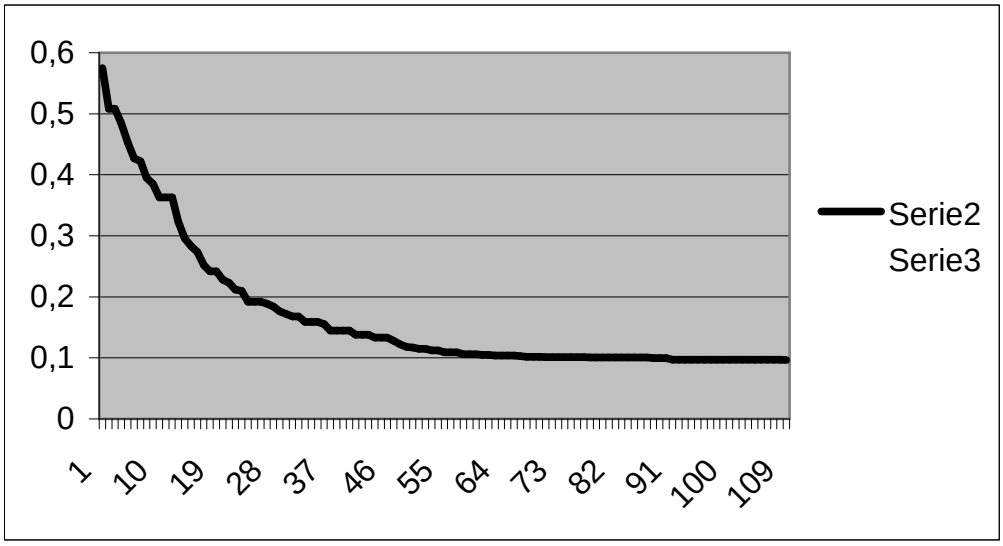
Por lo tanto se opta por mantener el intercambio de un valor de una media de un elemento de la población con otro.

Criterio de finalización.-

Como criterio de finalización se han realizado pruebas observando en cada generación la variación que se obtenía mejorando el valor de la función de evaluación obteniendo que si el número de generaciones consecutivas sin obtener mejora en la función de evaluación era de 25 , se estabiliza prácticamente dicho valor.

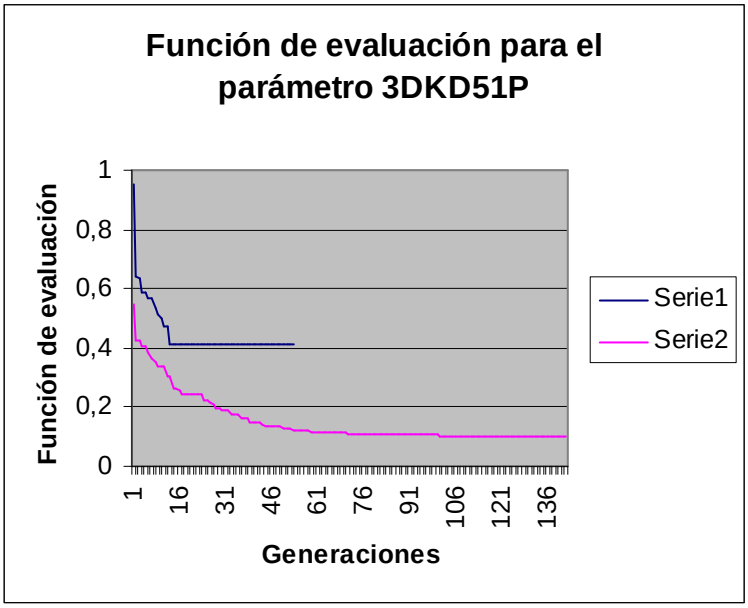
También se podría haber optado por limitar el número de generaciones a 100.

Ejecutado el programa con los parámetros ajustados da la siguiente gráfica:



Ajustados los parámetros en el algoritmo genético propuesto quedan de la siguiente forma:

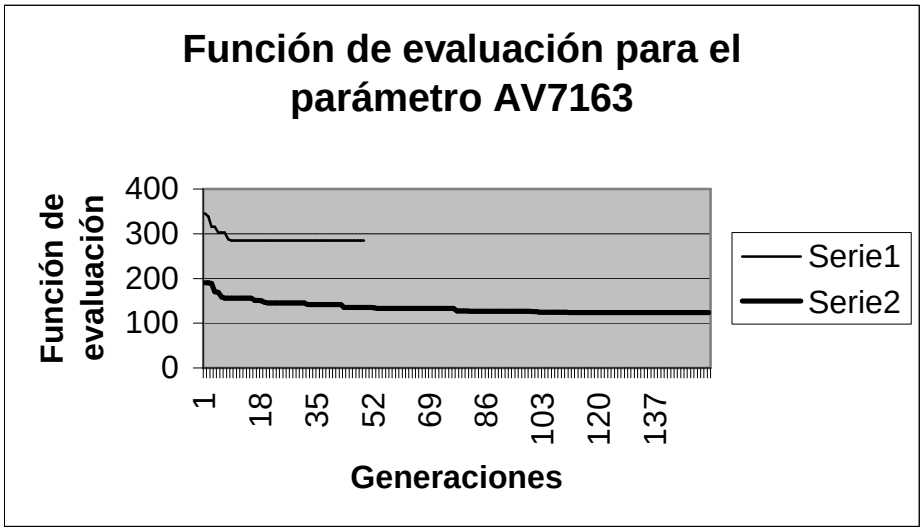
Elementos de la población:	**50.**
Número de cluster iniciales:	**40.**
Reemplazo:	**Élite de un elemento con réplica.**
Criterio de finalización:	**25 generaciones consecutivas sin obtener mejora en la función de evaluación.**
Operador de mutación:	**1% del valor de la media.**
Operador de cruce:	**Intercambio del valor de una media entre elementos de la población.**

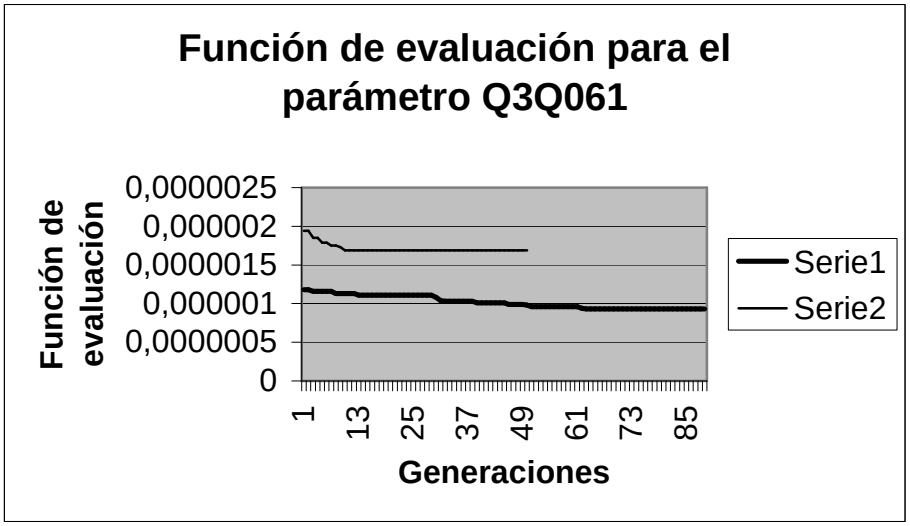


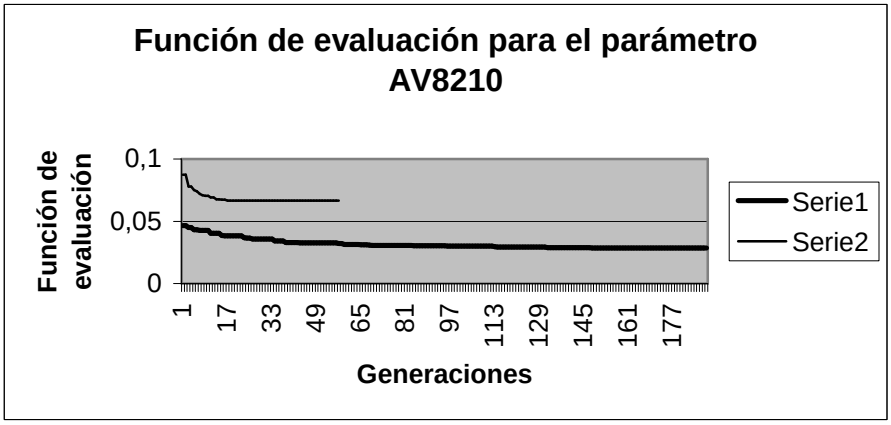
Serie1 Parámetros sin ajustar.
Serie2 Parámetros ajustados.

Esta tendencia se confirma aplicándolo a otros tres parámetros, escogidos de entre los datos de la empresa por sus rangos de valores diferentes al utilizado en el estudio anterior, siendo en concreto AV7136, Q3Q061 y AV8210, cuyas gráficas de las funciones de evaluación aparecen en la página siguiente.

En las tres gráficas que vienen a continuación, la línea fina corresponde a la utilización del programa sin ajustar los parámetros, y la línea más gruesa corresponde a la ejecución con parámetros ajustados.







6.2.4 Test de validación cruzada.-

Se va a utilizar el algoritmo dividiendo los datos en 10 grupos, de tal forma que se va a utilizar para aprendizaje / entrenamiento del algoritmo 9 grupos y el restante se utiliza para testear, es decir utilizaremos el algoritmo de validación cruzada al 90%, repitiéndolo en total diez veces, de tal forma que en cada ejecución se deja un grupo diferente para testear.

Para pocos datos está demostrado que el algoritmo de validación cruzada da resultados muy satisfactorios [Goutte 1997].
ftp://eivind.inm.dtu.dk/dist/1997/goutte.nflcv.ps.gz).

Para comprobar el grado de similitud entre la distribución original de los datos y los valores de centroides de los cluster se aplica la distancia de Kullback-Leibler [Kullback 1951], aplicado en las diez ejecuciones nos da los siguientes valores:

$$K(f \,\|\, f') = \sum_{i=1}^{m} f_i * \ln \frac{f_i}{f_i'}$$

$$K(f' \,\|\, f) = \sum_{i=1}^{m} f_{\acute{\imath}}' * \ln \frac{f_i'}{f_i}$$

		P	R	U	E	B	A	S		
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª
K(f||f')	1.02	0.11	1.01	1.16	0.01	0.47	0.12	0.97	1.02	1.05
K(f'||f)	0.96	0.05	0.95	0.78	0.04	1.53	0.18	0.98	0.94	0.91

Cuando la distancia sea pequeña entre estos dos valores se obtiene que las distribuciones son similares.

Observando la tabla anterior se concluye que hay una diferencia media de 0.193 demostrando la gran similitud entre las dos distribuciones, ya que este valor es pequeño, sin llegar a ser cero que indicaría un sobreaprendizaje, ya que sería la misma.

También se han tomado los errores cuadráticos medios que se obtenían así como cuantas desviaciones típicas estaba más alejado del segundo cluster que del primero, siendo los valores los siguientes:

Error medio	0.02	0.01	0.02	0.02	1.06	0.02	0.01	0.01	0.02	0.02
Desviac típicas	229	20	132	193	10	130	10	150	179	164

Obteniendo un valor de 0.121 como error cuadrático medio de las diez ejecuciones del algoritmo.

Otro detalle interesante es el número de cluster diferentes que se obtiene con estas diez ejecuciones diferentes siendo los valores : 15 (en tres ocasiones) , 14 (en 6 ocasiones) y 13 (en una).

Mención especial cabe reseñar que la primera prueba que se hizo para la validación cruzada se tomaron los componentes de cada grupo de manera consecutiva de los datos, obteniendo en algunas ejecuciones un error cuadrático medio de importancia (4 o 5), concluyendo que era debido a que no se captaban situaciones que se producían en estos intervalos no considerados. Para evitar esta situación los datos anteriormente citados han sido obtenidos llevando cada elemento leído a un grupo consecutivo diferente, haciendo que el primer grupo estuviera formado por los elementos 1, 11, 21..., el grupo segundo por los elementos 2, 12, 22 ... y así sucesivamente. La diferencia como se puede observar por los datos expuestos ha sido muy grande.

6.3 Generación de la red bayesiana.-

Ya que la búsqueda del grafo más adecuado es un problema NP- Hard se impone algún método de búsqueda heurístico, en este trabajo se utilizan los algoritmos evolutivos.

Mención hay que hacer a un artículo similar en algunos puntos al trabajo que se va a exponer:

[Larrañaga 1996] lo hace sobre datos ficticios usando PLS, primero genera un grafo, luego genera con PLS los datos desde el grafo y luego intenta aprender el grafo desde los datos. Además utiliza para la representación de los individuos un string y por lo tanto utiliza una estructura fija, en cambio en esta memoria de investigación se utiliza un estructura dinámica, independiente del número de variables antes de comenzar la ejecución del programa.

En el artículo mencionado anteriormente se comparan los grafos a través de:

- La similaridad entre estructuras. Número de arcos que sobran y que faltan entre ellos.

- La calidad de la estructura generada. Se usa la distancia de Kullback-Leibler [Kullback 1951] entre la distribución de los casos de estudio y la distribución de la red aprendida.

- La velocidad de convergencia, como el número de generaciones hasta que el algoritmo se detiene.

En el trabajo propuesto se va a **utilizar la técnica de algoritmos genéticos para maximizar la función de adecuación de una red bayesiana, siendo la métrica utilizada Bayesian Dirichlet con parámetros (1,1,...1), distinguiendo claramente entre parámetros de entrada y de salida, así como utilizando asignación dinámica de memoria para poder variar el número de parámetros a estudiar.**

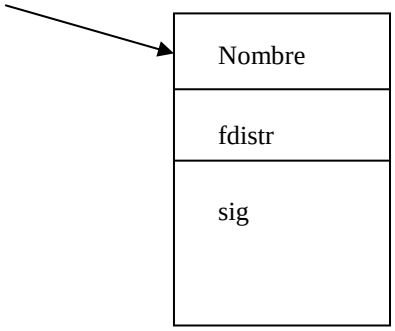
El algoritmo se encuentra resumido en la página siguiente, a continuación se va detallando cada procedimiento.

Crear consistirá en generar una lista simplemente enlazada teniendo un nodo para cada variable.

Habrá una lista para los nodos de entrada (aquellos que no van a tener padres, ya que son variables que introduce el usuario a su criterio) y otra para las variables de salida (que si pueden tener padres).

```
Crear(entrada);
Crear(salida);
Cargarfdistrinicial (entrada);
Cargarfdistrinicial (salida);

Crearycargarpoblacioninicial;
Evaluar(poblacion);

vueltas := 0;
Fin(poblacion, solucion,vueltas);

while not solucion and (vueltas < 300)  do
      begin
      Elite(poblacion,nuevapoblacion);
      Seleccion(poblacion);
      Cruce(poblacion);
      Mutacion(poblacion);
      Nuevageneracion(poblacion, nuevapoblacion);

      Evaluar(poblacion);
      Fin(poblacion, solucion,vueltas);
      end;
Escribirsolucion(poblacion);
```

Entrada/ Salida



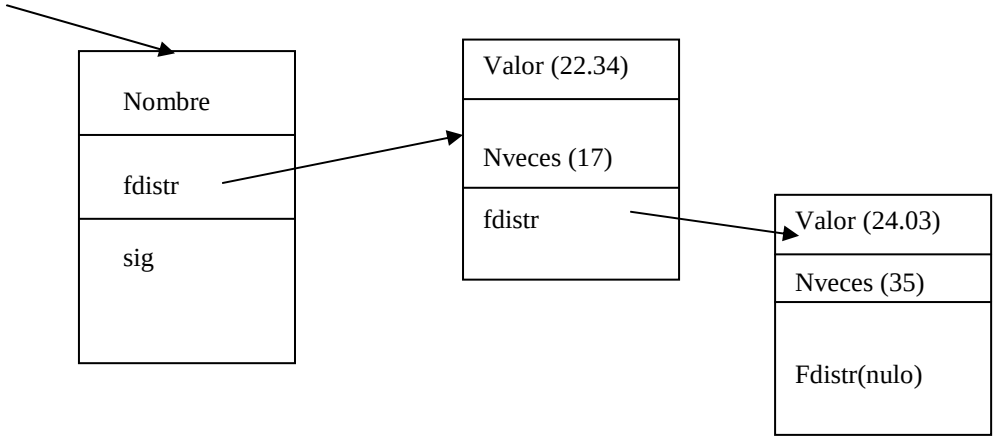
A su función de distribución

Al siguiente nodo

Cargarfdistrinicial consistirá en que partiendo de los datos de entrenamiento generará una lista simplemente enlazada con un nodo por cada valor de la variable correspondiente y el número de veces que se repite este dato en concreto.

Esto se hace tanto para las variables de entrada como para las variables de salida.

Entrada o Salida



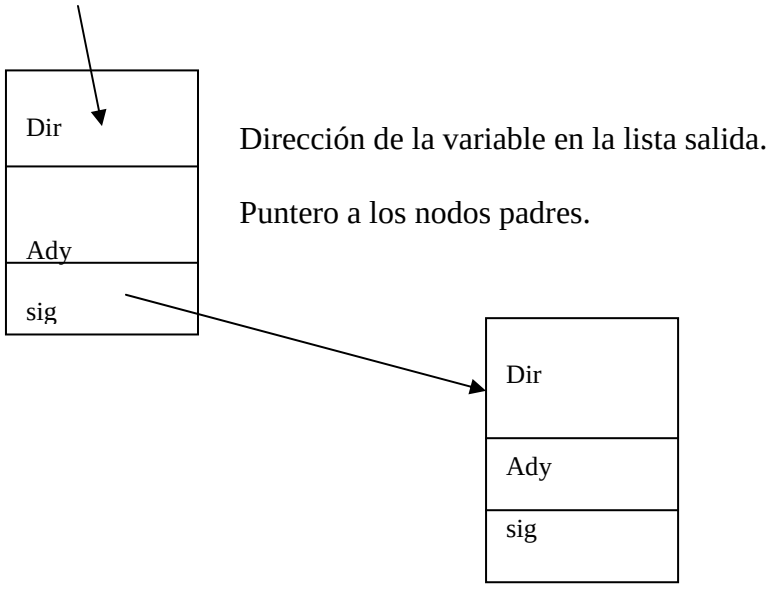
Crearycargarpoblacioninicial consistirá en

- Crear la población.
- Crear la estructura de un individuo de la población.

La información que debe mantener un individuo que lo diferencia de los otros, únicamente reside en los padres/ hijos que tiene concretamente en cada variable, además de un valor numérico que expresa la adecuación de este individuo al problema plantado.

El resto de la información reside en una estructura única que es entrada o salida mencionada anteriormente.

Respecto a la elección de los padres o los hijos se tomó la decisión de mantener a los padres directos de cada variable, ya que la función de distribución depende de la combinación de valores de los padres, lo que facilita el cálculo posterior de la función de distribución, en lugar de mantener la información de sus hijos.



Dirección de la variable en la lista salida.

Puntero a los nodos padres.

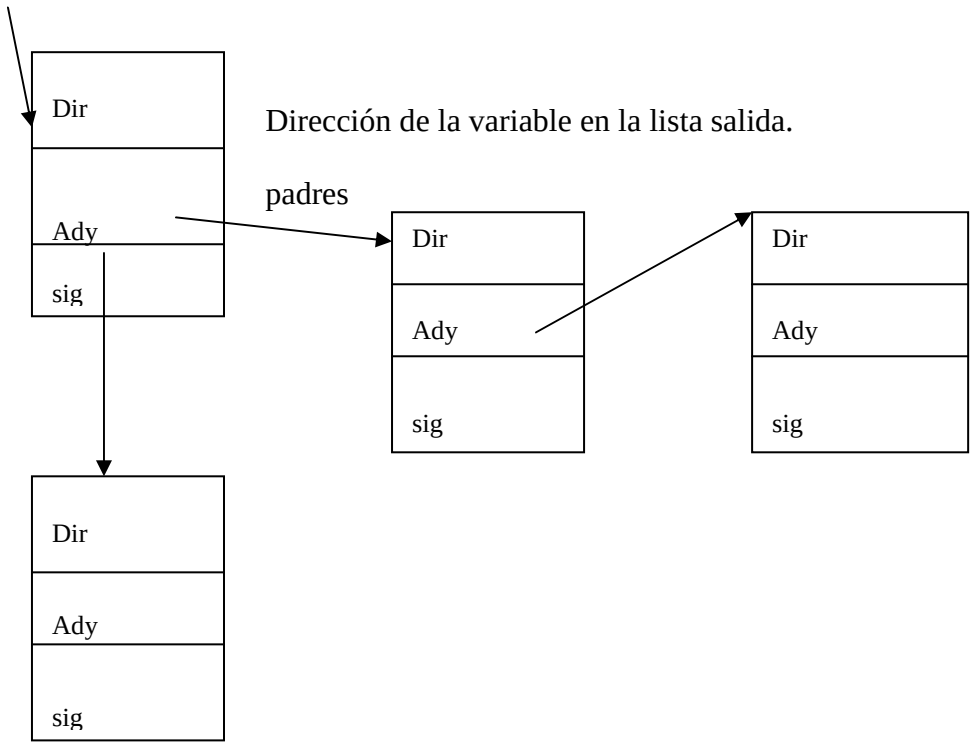
- Crear todos los individuos restantes(29):

  Copiando la estructura de un individuo e insertando aleatoriamente padres en cada individuo en número doble del anterior y empezando por dos, excepto el primero que no tiene padres.

  Para la inserción de un padre el nodo hijo debe de ser uno que corresponde exclusivamente a una variable de salida, y para el nodo padre puede ser de entrada y de salida siempre que no sea el mismo que el hijo, ya que provocaría ciclo.

  Especial mención cabe reseñar que dado que la red bayesiana es un grafo acíclico dirigido, cada vez que se intenta insertar un individuo se debe de comprobar antes que no provoca ciclo, si es así este nodo padre no se inserta

  Para el procedimiento ciclo se guarda en una pila una variable de salida, cuando se saca una, se señala el nodo visitado y se añaden a la pila sus nodos padres siempre que no sea la variable inicial, en cuyo caso hay ciclo, o bien ya haya sido visitado ese nodo. El proceso acaba con una variable cuando la pila queda vacía o bien hay ciclo. Este proceso se repite para cada variable de salida en el caso de que no se haya localizado ciclo.



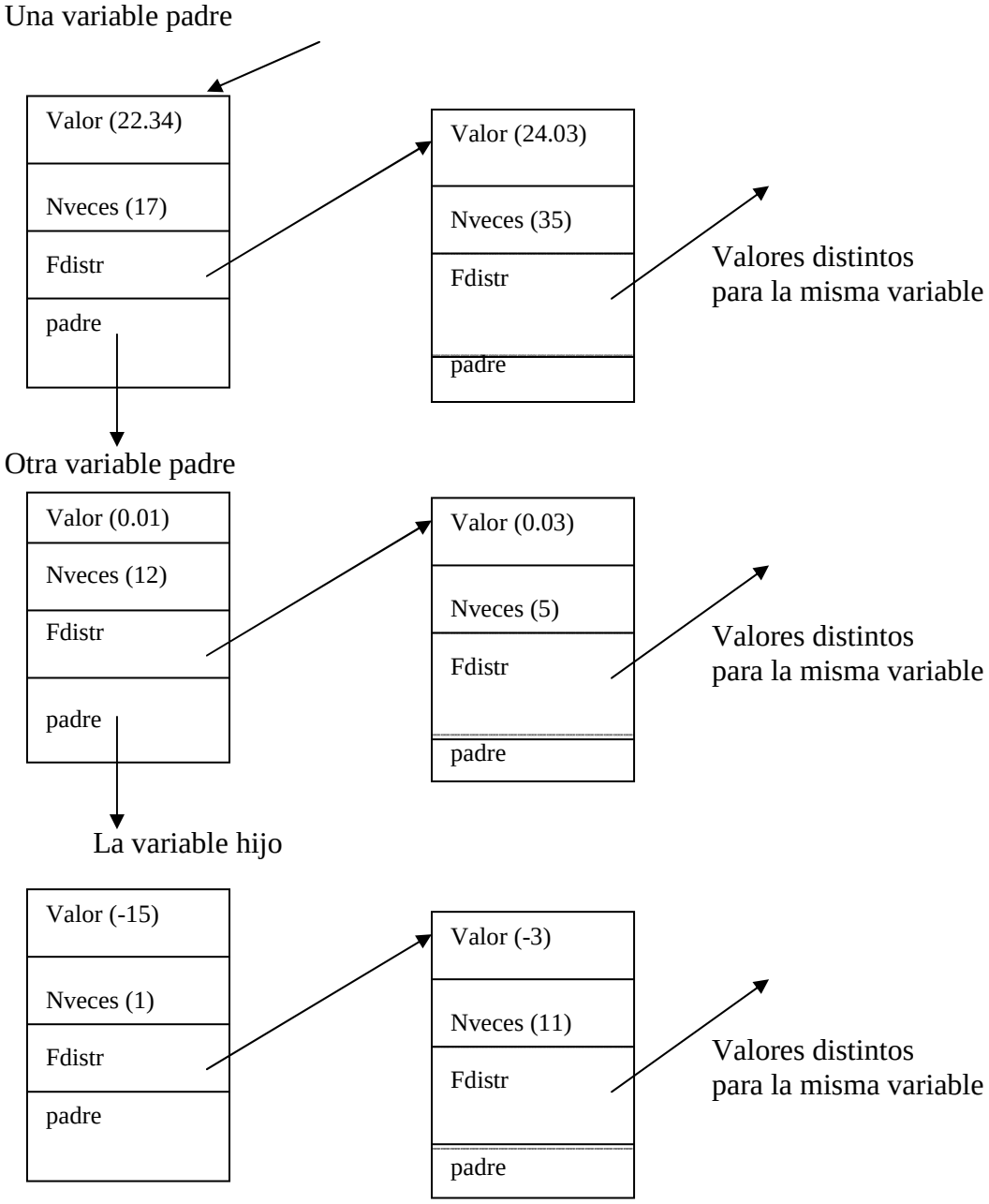
Para evaluar a cada individuo se tiene en cuenta la métrica Bayesian Dirichlet, enunciada anteriormente, y cuya fórmula es:

$$P(X_1 = x_1 .. X_n = x_n) =$$

$$\frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3 + \ldots + \alpha_n)}{\Gamma(\alpha_1) * \Gamma(\alpha_2) * \Gamma(\alpha_3) * \ldots * \Gamma(\alpha_n)} \quad \frac{\Gamma(\alpha_1 + n_1)\Gamma(\alpha_2 + n_2)\Gamma(\alpha_3 + n_3) .. \Gamma(\alpha_n + n_n)}{\Gamma(N + \alpha_1 + \alpha_2 + \alpha_3 + .. \alpha_n)}$$

partiendo de unos valores iniciales Dirichlet(1, 1, 1, ... 1) , es decir considerando equiprobables inicialmente todos los estados.

Por lo tanto es necesario conocer cuantos elementos hay en cada combinación de valores diferentes de los padres, para ello se generará la siguiente estructura para cada variable:



Dado que son productos de muchos factores, todos menores que uno ya que es probabilidad, se ha optado por trabajar con el logaritmo neperiano de este valor, evitando muchos problemas de desbordamiento en este cálculo al pasar las multiplicaciones / divisiones a sumas o restas.

Dicho procedimiento colocará en cada individuo de acuerdo con la métrica enunciada el valor de su adecuación al problema planteado.

Fin consistirá en pasar por todas los individuos detectando si alguno consigue llegar al umbral marcado, que es 0, ya que el logaritmo neperiano de 1 es cero y estamos hablando de probabilidades, si esto es así devolverá cierto en solución y acabará el programa.

Si no se ha mejorado la solución de la generación anterior sumará uno a las vueltas sin mejorar, en caso contrario la pone a cero. Si llega a 300 sin mejorarla el programa concluye con la solución obtenida.

Elite consistirá en obtener el individuo que tenga la adecuación mejor y extraerlo automáticamente para la siguiente generación. Con esto se consigue que el algoritmo sea convergente.

Además se escoge también este mismo individuo para realizarle las operaciones de mutación y cruce (a esto se le llama élite con réplica).

Selección, se escogen para realizar las operaciones de mutación y cruce a todos los individuos.

Cruce consistirá en intercambiar los padres de una variable al azar entre dos individuos separados n div 2, siendo n el número de individuos totales. Todo esto se repite (n div 2) veces.

Mutación para todos los elementos de la población se escoge aleatoriamente si se quita/añade un padre. Sólo se quita si tiene algún padre y sólo se inserta en el caso de que no se produzca ciclo.

Nuevageneración consistirá en cambiar la generación pasada con la nueva, una vez aplicado élite, cruce y mutación.

Escribirsolucion consistirá en representar al individuo que ha obtenido la adecuación mejor en la última generación y por lo tanto es la solución (red bayesiana) que obtiene este algoritmo.

La salida que genera el programa exactamente es la siguiente:

ADECUACION.->
0.00000000
##########

ORIGEN ALIHFN
---------- PADRES : DIRTM2 AV8095 F1F101 ANGC21

ORIGEN TIROMEZC (TIRO CM)
---------- PADRES : FCI105 SOPLADC3 ANGC2 F3F100 ANGC1 F1F101

ORIGEN ASPS2N
---------- PADRES : SOPLADC3 AV8095 COLAP3 ANGC1

ORIGEN TIROCALD
---------- PADRES : SOPLADC3 COLAP2 F3F100 ANGC1

ORIGEN 3DKD051P
---------- PADRES : F3F100 F1F101 F1F102 SOPLADC3

ORIGEN AV7163
---------- PADRES : F1F101 Q3Q100M ANGC3 CACL1 F1F102

ORIGEN AV8090
---------- PADRES : CACL2 CAUDALS2 ANGC2 TIROMEZC ANGC1 F1F102 ASPS2

ORIGEN AV8095
---------- PADRES : TIROMEZC F1F101 DIRTM1

ORIGEN COLAP2
---------- PADRES : CACL2 COLAP3 CAUDALP2 TIROMEZC TOFLASH ANGC1

ORIGEN AV8210
---------- PADRES : SOPLADC4 CAUDALP2 PCMEZC AV8090 ASPS2 TOFLASH SOPLADC3

ORIGEN CACL1
---------- PADRES : COLAP2 AV8095 F1F101 AV8210 DIRTM2 Q3Q100M CAUDALP2

ORIGEN CACL2N
---------- PADRES : F1F102 SOPLADC3 3DKD051P CAUDALS2 F3F100 FCI105

ORIGEN DIRTM1
---------- PADRES : CACL2 ANGC4 F1F101 TIROMEZC F3F100

ORIGEN DIRTM2
---------- PADRES : F1F101 COLAP3 TIROMEZC 3DKD051P TIROCALD SOPLADC3 F1F102

ORIGEN PCMEZC
---------- PADRES : CAUDALP2 COLAP3 FCI104 ALIHF TIROCALD ANGC3 ANGC1 FCI105 CAUDALS2 F3F100

ORIGEN COLAP3
---------- PADRES : Q3Q100M CAUDALS1 ANGC1 CACL2 SOPLADC4 TIROMEZC

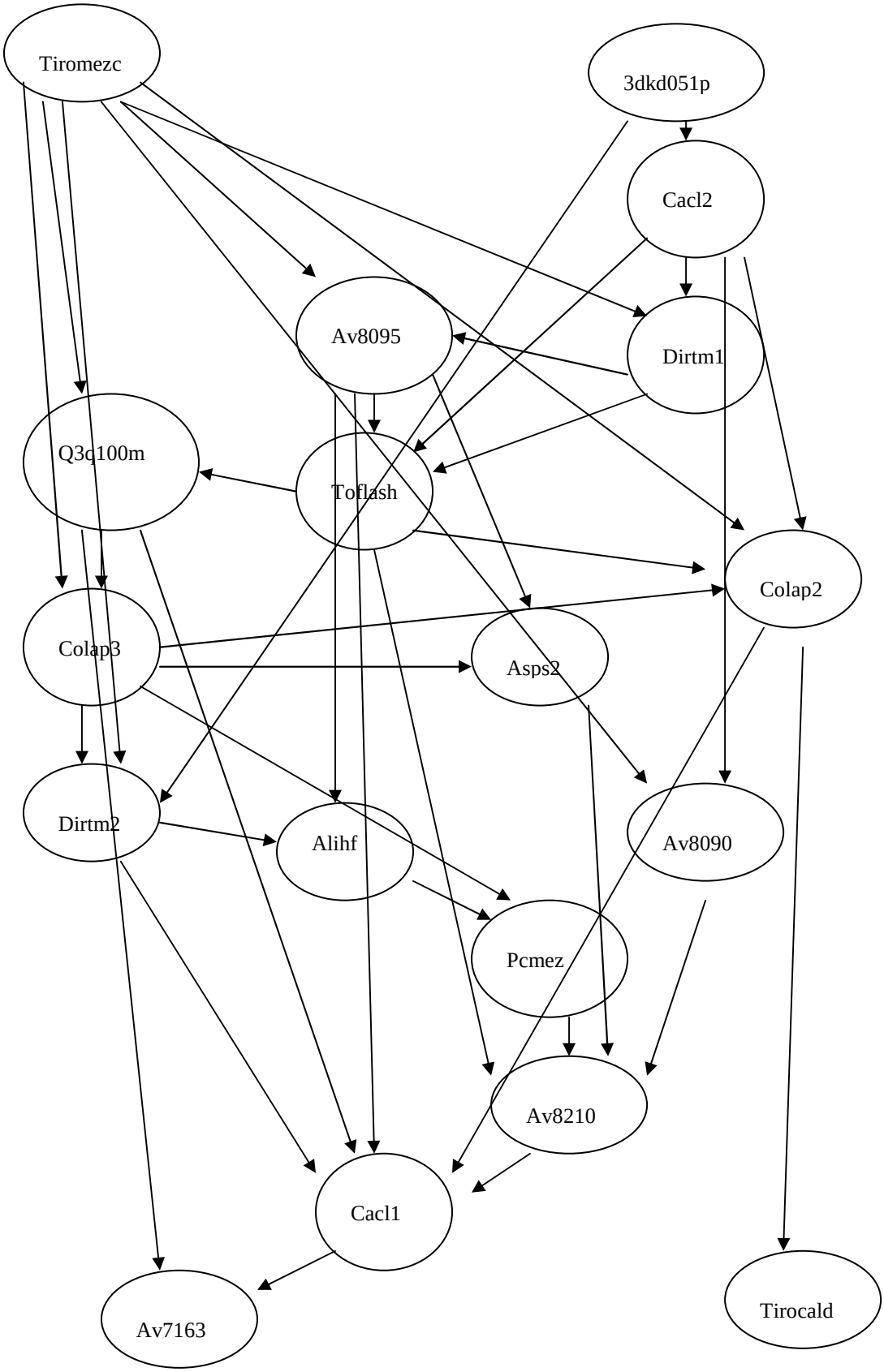
ORIGEN Q3Q100M
---------- PADRES : CAUDALS1 TIROMEZC CAUDALS2 SOPLADC4 TOFLASH SOPLADC3

ORIGEN TOFLASH
---------- PADRES : ANGC2 AV8095 DIRTM1 F3F100 SOPLADC4 CAUDALS1 FCI105 CACL2 ANGC1

De la representación gráfica que viene en la hoja siguiente se puede concluir que en la variable TIROMEZC (TIRO CM) sólo influyen variables de entrada, en ningún momento aparecen variables de salida y entre las variables que determinan la función de distribución de la variable mencionada aparece la variable F3F100, que ya aparecía en el estudio inicial de la matriz de correlación.

Además cualquier modificación del valor de TIROMEZC afecta de forma directa a las funciones de distribución de muchas variables de salida: DIRTM1, COLAP2, AV8095, AV8090, DIRTM2, COLAP3 y Q3Q100M.

Sólo aparecen las variables clasificadas como de salida.



6.4 Trabajos pendientes.-

- **Ajuste de parámetros** en la generación de la red bayesiana.

- Estudio con los **datos nuevos** enviados por Atlantic Cooper.

  Dada la gran cantidad de datos suministrados deberá de hacerse un estudio profundo para la selección de los mismos tanto de tuplas como de atributos.

- Detección de **outliers**.

  Se tiene en principio pensado no considerar aquellos valores que sobrepasen las tres desviaciones típicas respecto de la media del atributo.

- Relaciones **causa-efecto**.-

  Cómo puedo pasar de una flecha que indica dependencia condicional a que exprese causa-efecto.

- **Problemas dinámicos**.-

  Surgen como una generalización natural de los modelos estáticos, cuando el dominio a modelar se relaciona en el tiempo de un conjunto de variables aleatorias.

- **Retardos** que se producen cuando se manipulan los datos de entrada.

- **Variables discretas – continuas**.

  El hecho de que haya un número finito de estados para que una variable se considere como discreta, no se ve claro en este trabajo, ya que por ejemplo el número de revoluciones de un motor es un valor discreto, pero es tan grande la relación de estados diferentes que hace pensar en otro concepto diferente que exprese claramente cuando se debe de aplicar una discretización previa.

- **Diagramas de influencia**.-

  En una clase de problemas reales no estamos interesados únicamente en conocer el valor de la probabilidad sino en determinar qué opción de entre un conjunto de decisiones posibles es la que maximiza la utilidad esperada.

  Los diagramas de influencia son modelos gráficos que permiten la modelización de este tipo de situaciones mediante la inclusión de nodos de decisión que representan las posibles opciones y nodos de utilidad que valoran los resultados.

7.- REVISION BIBLIOGRAFICA.-

Discretización.-

[Dagneli 1997]
Analyse Statistique a plusieurs variables, by Pierre Dagneli.Diffusion.

[Dempster 1997]
Maximum likelihood with incomplete data via the EM algotihm, by A. Dempster, N. Laird, D. Rubin. J. Roy, Statist. Soc. Ser. B (1997) 1-22.

[Peña 1997]
Estadística. Modelos y métodos. 1. Fundamentos, by Daniel Peña Sánchez, Catedrático de la Univ. Carlos III de Madrid. Alianza Universidad Textos. 1997.

Medida del closeness.-

[Kullback 1951]
Annals of Mathematics and Statistics p. 79-86. Kullback-Leibler 1951.

[Kullback 1969]
Medida de la divergencia, Ku ,H.H. and S. Kullback. 1969 Approximating discrete probability distributions. IEEE Trans. Inform. Theory 15 (4) 444-47.

Test de validación cruzada.-

[Goutte 1997]
Note on free lunches and cross-validation, Neural Computation, 9, 1211-1215. ***ftp:// eivind.inm.dtu.dk /dist /1997/ goutte.nflcv.ps.qz***

Redes bayesianas.-

[Blackmond 2001]
INFT 819 Computational models for probabilistic inference unit 5 Learning bayesian networks from data. Katheryn Blackmond Laskey.

[Chickering 1995]
5ª Conference on Artificial Intelligence and Statistics. P. 112-128.

[Cooper 1992]
Computational complexity of probabilistic inference using bayesian belief networks. Artificial Intelligence, 42, 393-405.

[Etxeberria 1997]
Proc of Causals Models and Statistical Learning p. 151-168.

[Gamma 2001]
A linear bayes classifier.

[Heckerman 1995]
Heckerman, Geiger y Chickering 1995. Learning Bayesian Networks: The combination of knowledge and statistical data. Machine Learning, 20, 197,Kluwer Academic Publishers.

[Jensen 1996]
An introduction to bayesian networks. New York. Spronger Verlag, UCL Express.

[Jensen 2001]
Aalborg University. A brief Overview of the three main paradigms of expert systems.

[Larrañaga 1996]
Structura Learning of bayesian networks by genethic algorithms: An análisis of control parameters. IEEE.

[Lauritzen 1988]
Local computation with probabilities on graphical structures and their application to expert systems, Journal of the Royal Statistical Society, B 50, 157-224 .

[Mitchell 1998]
Machine Learning . Mac Graw Hill.

[Neapolitan 1990]
Probabilistic Reasonning in Expert Systems. Theory and Algorithms. Wiley / Interscience, New York.

[Pearl 1986]
Fusion, propagation and structuring in belief networks. Artificial Intelligence 29, p. 241-288.

[Pearl 1988]
Probabilistic reasoning in intelligent systems: Networks of plausible inference, San Mateo, Morgan-Kaufmann.

[Peña 2000]
Peña, J. A. Lozano, P. Larrañaga 2000. An improved bayesian structural EM algorithm for learning bayesian networks for clustering.

[Rusell 1995]
Artificial Intelligence: A modern approach. Englewood Cliffs, NJ: Prentice Hall.

[Whiltaker 1990]
Capítulo 3. Graphical Models in Applied Multivariate Statistics, Wiley, Chichester, UK.

[Xiang 1999]
Belief updating in multiply sectioned bayesian networks without repeated local propagations.

Árboles de decisión, reglas, listas de decisión, redes neuronales, aprendizaje basado en ejemplos.

[Agrawal 1993]
Efficient similarity search in sequence databases. Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms.

[Bezdek 1981]
Pattern recognition with fuzzy objective functions algorithms. Plenium press. New York.

[Breiman 1984]
Classification and regression trees. Wadsworth Belmont, CA.

[Clark 1989]
Knowledge representation in Machine Learning.

[Dasarathy 1991]
Nearest neighbor pattern classification techniques. IEEE Computer Society press. 1991.

[DeJong 1993]
Learning search control knowledge for deep. Machine Learning.

[Janikow 1993]
A knowledge intensive GA for supervised learning. Machine Learning, 13, 198-228.

[Kohonen 1982]
Self-Organized formation of topologically correct feature maps. Biological Cybernetics 43, 59-69.

[McCulloch 1943]
A logical calculus of the ideas immanent in nervous activity. Bulletin of the Mathematical Byophisics 5, 115-133.

[Michalski 1987]
Constraints and preferences in inductive learning: An experimental study of human and machine performance. Cognitive Science 11 (3): 299-339.

[Pawlak 1991]
Rough sets: Theoretical aspects of reasoning about data.

[Quinlan 1993]
C4.5: Programs for Machine Learning. Morgan Kauffmann 1993. 9 Machine Learning.

[Rivest 1987]
Learning decision trees. Machine Learning Vol. 2, 229-246.

[Riquelme 1998]
Cogito. Una herramienta para obtener un clasificador jerárquico en aprendizaje supervisado. CAEPIA 1997.

[Rosenblatt 1958]
The Perceptron: a probabilistic model of information storage and organization in the brain.

[Sugeno 1993]
A fuzzy logic based approach to qualitative modeling.

8.- PERSONAS Y FOROS RELACIONADOS.-

ESPAÑA.-

- Enrique Castillo. Universidad de Cantabria.

- Elena Hernando. ETS de Telecomunicaciones. Universidad. Politécnica de Madrid.

- Pilar Lasala. Universidad de Zaragoza.

- Hay un grupo de investigación repartido entre las universidades del País Vasco, Almería, Granada, UNED (F. J. Díez, tiene su Tesis doctoral sobre aplicación de las redes bayesianas a ecocardiografía, Local conditioning in Bayesian networks, Artificial Intelligence 87, 1996. p. 1-20.).

OTROS LUGARES.-

- HUGIN, creado por un grupo de la Universidad de Aalborg (Dinamarca). Muchas empresas utilizan Hugin para la construcción de sistemas expertos probabilistas. (***http://www.hugin.dk***).

- Grupo de investigación en Microsoft Research (http://www.reseach.microsoft.com). Intentan normalizar la representación de modelos gráficos probabilistas.

- Src = source code included?
- Lib = linkable library/API included? (N means the program is a standalone executable.)
- Exec = Executable runs on W = Windows (95/98/NT), U = Unix, M = Mac, or - = any.
- Cts = continuous-valued nodes supported? (D = discretized)
- GUI = Graphical User Interface included?
- Learns parameters?
- Learns structure?
- Sample = sampling methods (e.g., likelihood weighting, MCMC) supported?
- Utility = utility and decision nodes (i.e., influence diagrams) supported?
- Free = Is a free version available? Yes, No or Restricted. (Commercial products often have free versions which are restricted in various ways, e.g., the model size is limited, or models cannot be saved; in this case, we say R for "restricted".)

Name	Authors	Src.	Lib.	Exec.	Cts	GUI	Learn Params	Learn Struct	Sample	Utility	Free	Comments
Analytica	Lumina	N		W	Y	W	N	N	N	Y	R	Spread sheet compatible.
Bayda	U. Helsinki	Java	-	-	Y	Y	Y	N	N	N	Y	Bayesian Naive Bayes classifier.
BayesBuilder	Nijman (U. Nijmegen)	N	N	W	N	Y	N	N	Y	N	R	-
Bayesian	KMI/Ope	N	N	WUM	D	Y	Y	Y	N	N	Y	Uses "bound and collapse" for

Knowledge Discoverer	n U.											learning with missing data.
B-course	U. Helsinki	N	N	-	D	Y	Y	Y	N	N	Y	Runs on their server: view results using a web browser.
Bayonnet	Motomura (ETL)	Java	-	-	NN	Y	Y	N	N	N	Y	For learning, represents BN as a neural net.
Belief net power constructor	Cheng (U.Alberta)	N	W	W	N	Y	Y	Y	N	N	Y	Uses cond. indep. tests to learn structure.
BN Toolbox	Murphy (U.C.Berkeley)	Matlab	-	-	Y	N	Y	Y	Y	N	Y	Also handles dynamic models, like HMMs and Kalman filters.
BucketElim	Rish (U.C.Irvine)	C++	-	WU	N	N	N	N	N	N	Y	Uses variable elimination for inference.
BUGS	MRC/Imperial College	N	N	WU	Y	W	Y	N	Y	N	Y	Uses Gibbs Sampling for inference.
CABeN	Cousins et al. (Wash. U.)	C	Y	-	N	N	N	N	Y	N	Y	Implements 5 different sampling algorithms.
CoCo	Badsberg (U. Aalborg)	C	-	WUM	N	N	Y	Y	N	N	Y	Designed for statistical analysis of contingency tables by discrete undirected graphical models.
CoCo+Xlisp	Badsberg (U.	C/lisp	-	U	N	Y	Y	Y	N	N	Y	Extends CoCo with GUI and block recursive models.

	Aalborg)											
CIspace	Poole et al. (UBC)	Java	N	-	N	Y	N	N	N	N	Y	Uses variable elimination for inference.
Ergo	Noetic Systems	N	N	WM	N	Y	N	N	N	N	R	-
Genie/Smile	U. Pittsburgh	N	WU	WU	N	W	N	N	Y	Y	Y	-
Hugin Light	Hugin	N	Y	W	Y	W	N	N	N	Y	R	-
Ideal	Rockwell	Lisp	-	-	N	Y	N	N	N	Y	Y	GUI requires Allegro Lisp.
Java Bayes	Cozman (CMU)	Java	-	-	N	Y	N	N	N	Y	Y	-
MIM	HyperGraph Software	N	N	W	Y	Y	Y	Y	N	N	R	Designed for chain graphs
MSBN	Microsoft	N	N	W	N	W	N	N	N	Y	Y	-
Netica	Norsys	N	WUM	W	Y	W	Y	N	Y	Y	R	-
Pronel	Hugin	N	N	W	N	W	Y	Y	N	N	R	Learns structure from fully observed discrete data.
RISO	Dodier (U.Colorado)	Java	-	-	Y	Y	N	N	N	N	Y	Only handles polytrees. Distributed implementation.
Tetrad	CMU	N	N	WU	Y	N	Y	Y	N	N	Y	Uses cond. indep. tests to learns causal structure.
Web	Xiang	Java	-	-	N	Y	N	N	N	Y	Y	-

Weaver	(U.Regina)											
XBAIES 2.0	Cowell (City U.)	N	N	W	Y	Y	N	N	N	Y	Y	Also handles chain graphs.

ANEXO.-

APLICACION REAL PARA H-ATRIBUTOS DE LA DISCRETIZACIÓN DE UN PARÁMETRO VISTO EN EL APARTADO 6 DEL PROYECTO DE INVESTIGACIÓN.-

Anexo.1 Introducción.-

Este trabajo realizado está puesto como anexo, ya que en principio se pensó en que la función del modelo sería la búsqueda de agrupamientos, con el fin de detectar en qué situación se encontraba la fábrica y poder predecir situaciones futuras con la consiguiente acción previa a tomar.

Una vez detectada que la función principal del modelo residía en la relación de dependencia entre parámetros se dejó dicho trabajo, para abordar la generación de la red bayesiana expuesta en el trabajo de investigación.

Sin embargo, creo interesante que aparezca dicho trabajo, ya que **representa una forma de utilizar los algoritmos evolutivos para técnicas NO supervisadas.**

Anexo.2 Desarrollo del trabajo.-

En el caso de un atributo la desviación típica que se tomaba era constante de 2, valor que no influía en el algoritmo para el caso de un atributo, ahora bien en el caso que tratamos de 17 atributos es necesario tener en cuenta las desviaciones típicas evitando de esta forma tener que normalizar los datos antes de su tratamiento. Ya que aparece la matriz de covarianzas se tiene en cuenta el efecto de la desviación típica en el cálculo de las probabilidades de cada distribución de dimensión h, no teniendo que normalizar los datos. Como ya se demostró previamente:

$$hipot_{ML} \quad = \quad \arg\min_{u_j} \quad \sum_{i=1}^{t} (Z_i - u_j)' \, M^{-1} (Z_i - u_j)$$

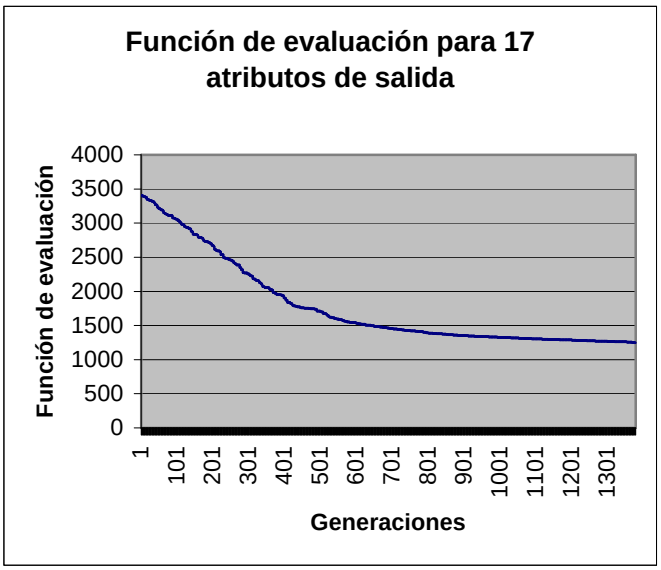
Los datos de Atlantic Cooper proporcionados son dos ficheros, que fusionados por el tiempo en que están tomadas las muestras, quedan 109 tuplas.

- Resultado de quitar parámetros que en gran medida tienen errores de lectura, como son
    1) Aspiración S1.
    2) Soplado C1.
    3) Soplado C2.
    4) Tiro en HF.

- Mezcla de los ficheros con muestras el primero cada 1m. 30 seg. y el segundo cada 30 seg.

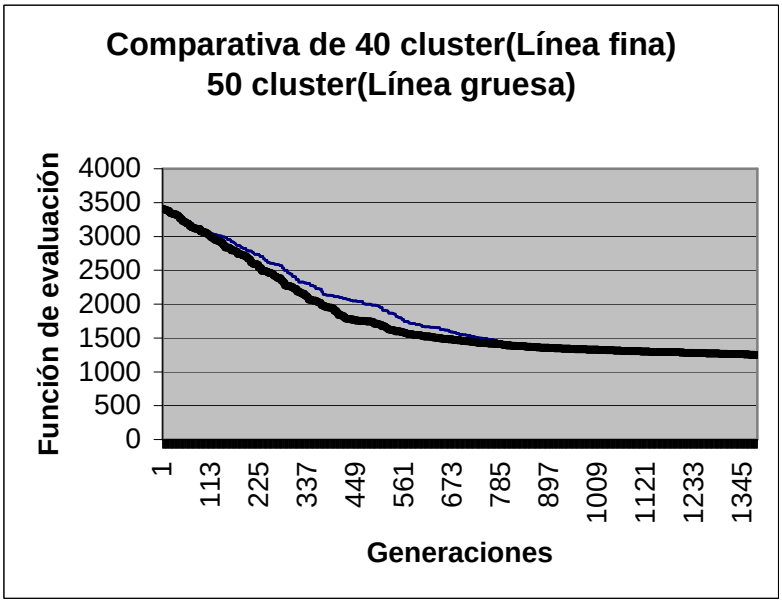
- Sólo aplicado a los parámetros que son de Salida y no a aquellos que son de entrada. Resultan 17 atributos de salida.

En concreto se va a realizar clustering sobre 100 tuplas, escogiendo en cada una de ellas 17 atributos. Es decir en nuestro caso concreto el vector es de 17 dimensiones, formado por los 17 valores reales de las medias.

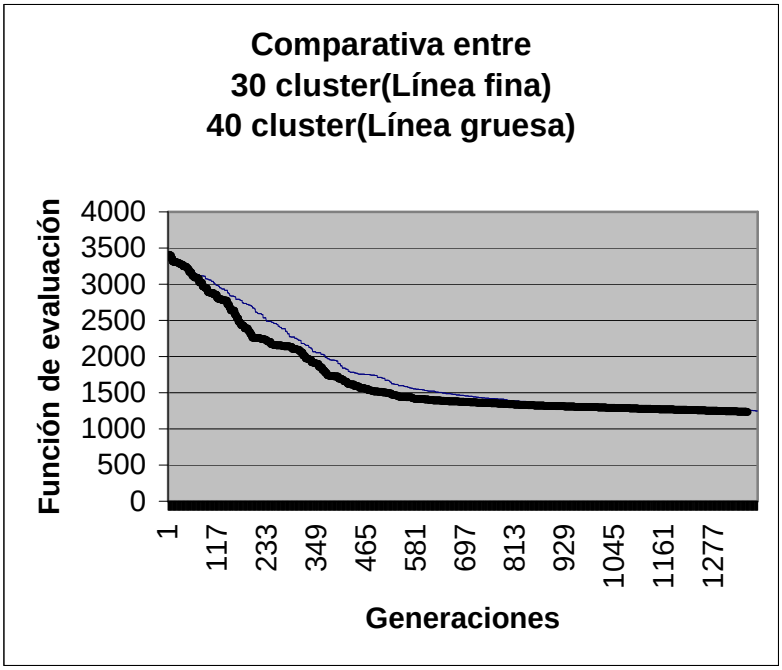


El inconveniente principal es la gran duración de tiempo, alrededor de 24 horas que tarda el algoritmo en realizar clustering sobre 17 atributos, por eso es necesario intentar optimizar esta función en el valor del tiempo.

Anexo.3 Ajuste de parámetros.-



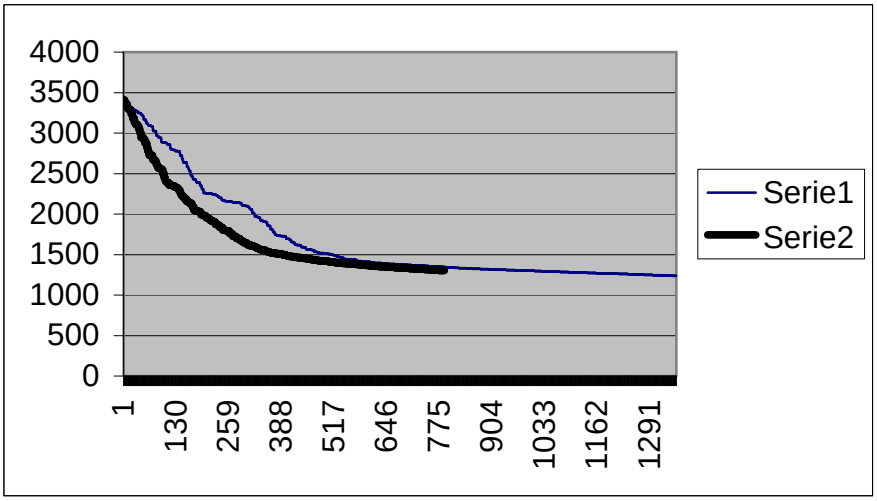
En la gráfica inferior de la página anterior se puede observar que si aumentamos el número de cluster iniciales la función de evaluación es muy similar, pero si tenemos en cuenta que el tiempo que tarda desde una generación a otra en el caso de utilizar 50 cluster es 1m. 40 seg. y el tiempo que tarda desde una generación a otra es de 1m. 10 seg. en el caso de utilizar 40 cluster, obtenemos como conclusión que no se alcanza mejora en el tiempo de ejecución aumentando el número de cluster iniciales.



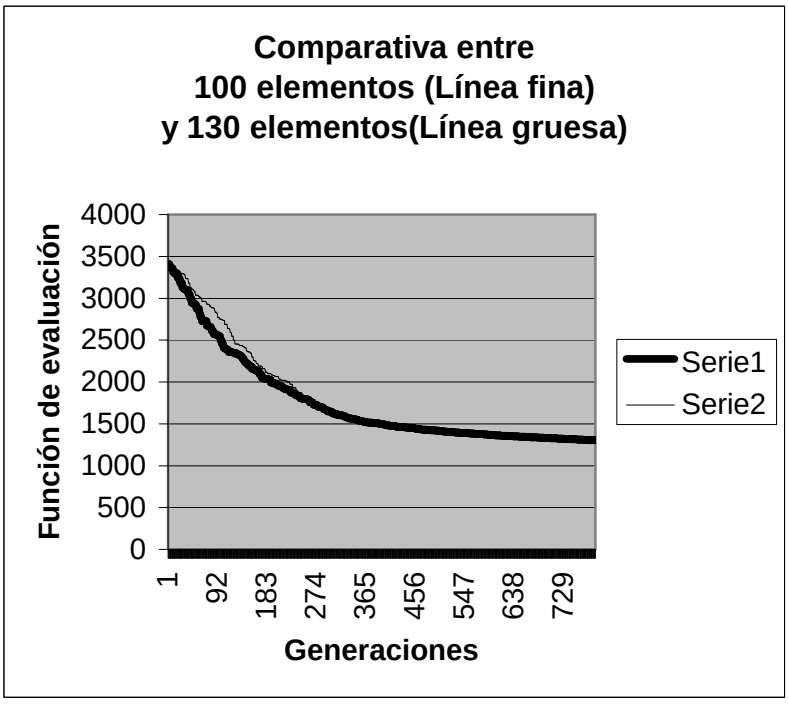
En la gráfica se puede observar que si disminuimos el número de cluster iniciales la función de evaluación es muy similar, pero si tenemos en cuenta que el tiempo que tarda desde una generación a otra en el caso de utilizar 40 cluster es 1m. 10 seg. y el tiempo que tarda desde una generación a otra es de 50 seg. en el caso de utilizar 30 cluster, obtenemos como conclusión que es conveniente disminuir el número de cluster iniciales a 30.

Esto es posible dado que el número de agrupamientos obtenidos finales se mantiene en 25.

Partiendo de que el número de clusters iniciales es de 30, variamos el número de elementos de la población 50 y 100, siendo el resultado el siguiente:

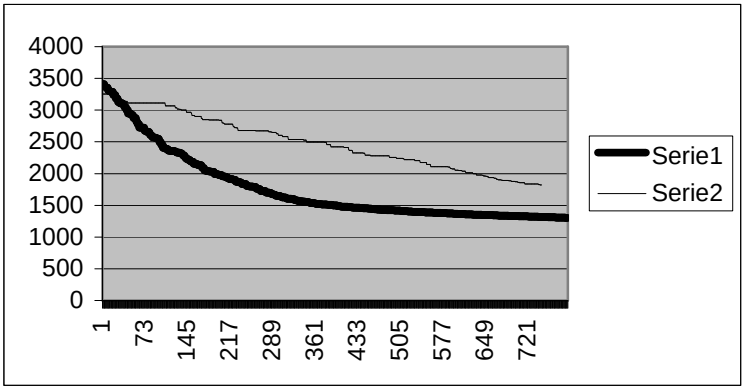


Si tenemos en cuenta que el tiempo que tarda desde una generación a otra es de 50 seg. en el caso de utilizar 50 elementos (Línea fina) y el tiempo que tarda desde una generación a otra es de 1m. 50 seg. (Línea gruesa) en el caso de utilizar 100 elementos, la conclusión es que el algoritmo es más rápido en el caso de utilizar 50 elementos en la población.

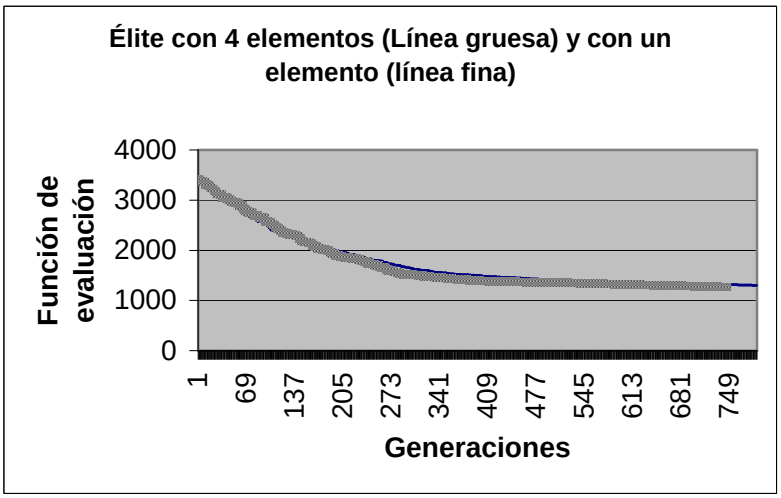


En la gráfica se puede observar que la función de evaluación es muy similar, pero si tenemos en cuenta que el tiempo que tarda desde una generación a otra es de 2 m. 15 seg. en el caso de utilizar 130 elementos en la población(Línea gruesa) y el tiempo que tarda desde una generación a otra es de 1m. 50 seg. en el caso de utilizar 100 elementos (Línea fina), la conclusión es mantener el número de elementos iniciales de la población en 100.

Hagamos la prueba de realizar la selección de un elemento para élite, con réplica (Línea gruesa) del elemento para mutación y cruce y en la otra prueba sin réplica (Línea fina), es decir sin tenerlo en cuenta para mutación y cruce.

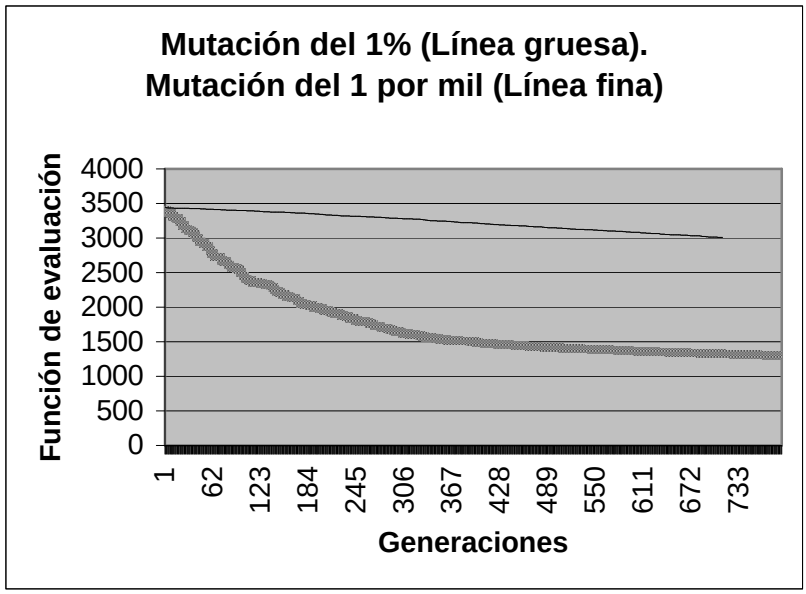


De la gráfica se concluye fácilmente que la rapidez en conseguir un mejor valor de la función de evaluación se decanta claramente por la versión de realizar mutación y cruce incluyendo también al elemento seleccionado como élite.

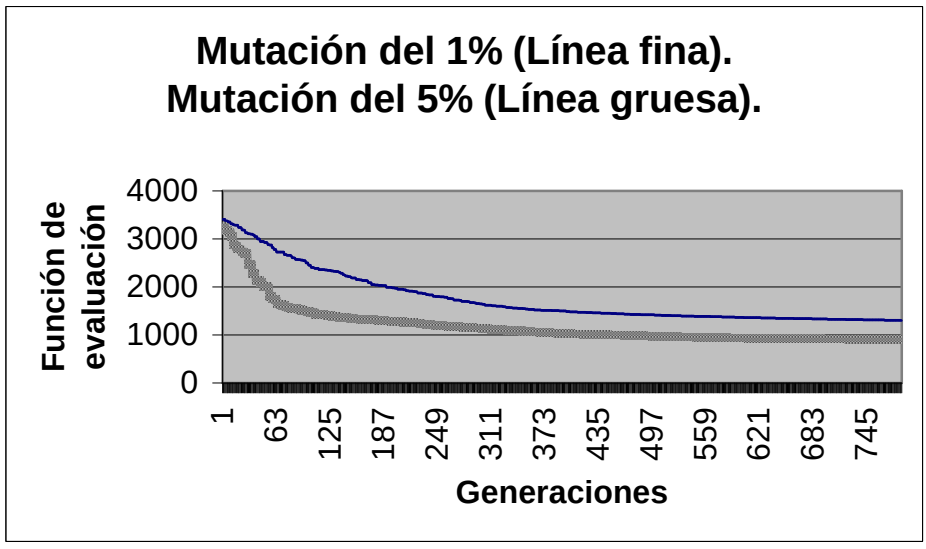


Se pasa a realizar pruebas con élite de un solo elemento como se está realizando hasta el momento y realizarla seleccionando como élite a un grupo de 4 elementos, concluyendo de la gráfica expuesta que los resultados son muy similares, con lo que se decide seguir con élite de un elemento.

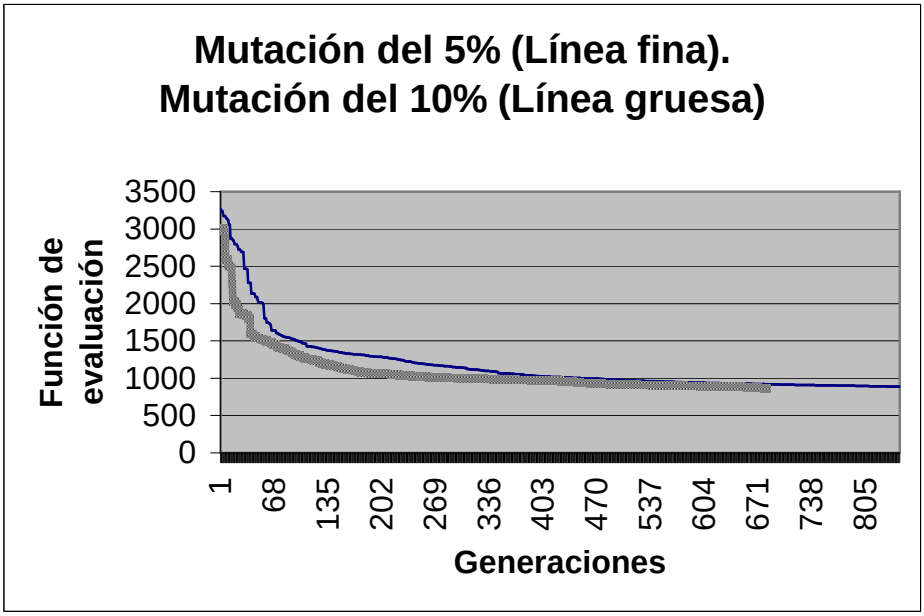
Se pasa a realizar estudio con una mutación del 1‰ y del 1%



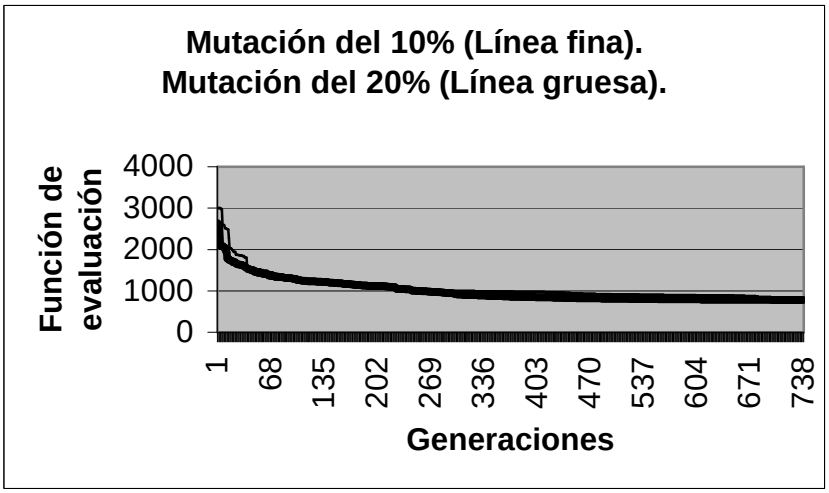
De la gráfica se puede obtener que una mutación del 1% obtiene mucho mejores resultados que si la mutación la hacemos del 1‰, luego se mantiene la mutación del 1%.



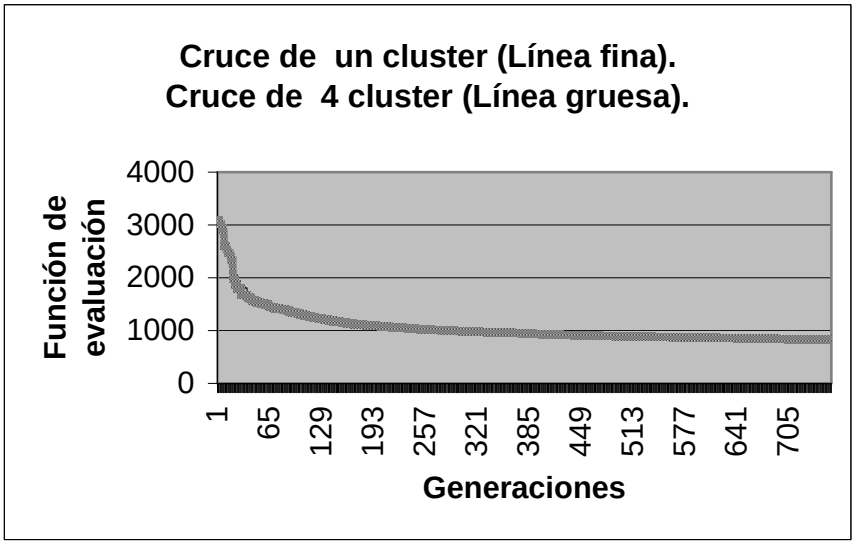
De la gráfica se puede obtener claramente que una mutación del 5% obtiene resultados muy rápidos en comparación con la mutación del 1%, luego se cambia a mutación del 5 %.



De la gráfica se puede obtener claramente que una mutación del 10% obtiene resultados muy rápidos en comparación con la mutación del 5%, luego se cambia a mutación del 10 %.



De la gráfica se puede obtener que una mutación del 10% obtiene resultados muy similares en comparación con la mutación del 20%, luego se mantiene la mutación del 10%.



De la gráfica se puede obtener que un cruce de un cluster o bien de 4 cluster no afecta para nada en los resultados finales.

Ajustados los parámetros del algoritmo evolutivo quedan de la siguiente manera:

- **Reemplazo: Élite de un elemento con réplica.**
- **Mutación del 10%.**
- **Cruce de un único cluster.**
- **Elementos de la población : 100.**
- **Número de cluster iniciales: 30.**