

Error de posición en clasificadores bayesianos para la clasificación de series temporales

Oscar J. Prieto, Carlos J. Alonso

Grupo de Sistemas Inteligentes
Departamento de Informática
Universidad de Valladolid, España.
{oscapri, calonso}@infor.uva.es

Juan José Rodríguez

Lenguajes y Sistemas Informáticos
Universidad de Burgos, España
jjrodriguez@ubu.es

Resumen

En este trabajo se propone el uso del error de posición de los clasificadores obtenidos al aplicar técnicas de clasificación bayesiana sobre series temporales como medida de utilidad de los mismos. El error de posición mide la posición media de la hipótesis correcta dentro de las propuestas por clasificadores que puedan establecer una ordenación de las clases. El error de posición nos proporciona una medida de confianza sobre los clasificadores que permiten la ordenación de hipótesis. La clasificación de series temporales consiste en determinar la pertenencia de un ejemplo a una determinada clase, y puede ser usada en la diagnosis de procesos continuos. En este tipo de procesos puede ser conveniente contar con hipótesis alternativas a la dada, en el caso de encontrarnos con una clasificación errónea. Los clasificadores bayesianos estudiados permiten asignar un orden dentro de las hipótesis. Los resultados que se presentan son los obtenidos sobre diferentes tipos de datos, por un lado se ha usado conjuntos de datos sintéticos en los que se considera una única serie temporal, también se ha trabajado con datos simulados de un proceso continuo realizado en una planta piloto. Además se han usado características que capturan propiedades interesantes de las series combinadas con técnicas de clasificación mediante redes bayesianas.

1. Introducción

La clasificación de series temporales es un tema de investigación abierto, para él se han

usado una gran variedad de técnicas [1]. El objetivo principal es la obtención de clasificadores sobre series precisos, aunque también se buscan clasificadores comprensibles, con el fin de entender los procesos continuos a los que suelen asociarse las series temporales. El problema más simple consiste en la clasificación de una única serie temporal. Los problemas de mayor complejidad aparecen cuando estas series temporales se corresponden con datos de sistemas dinámicos, en los que hay un gran número de componentes y un conjunto pequeño de variables. Los modelos de comportamiento de este tipo de sistemas no se conocen bien y son difíciles de estimar, existiendo interacciones entre componentes que tampoco se conocen con exactitud y donde la presencia de sistemas de control puede ocultar posibles fallos. Dentro de este tipo de problemas, los procesos continuos industriales son un ejemplo típico.

En este tipo de problemas es conveniente en algunos casos contar con hipótesis alternativas a la hipótesis dada por el clasificador, ya que en caso de error se hace necesaria una nueva clasificación. Los clasificadores bayesianos asocian una probabilidad a cada una de las hipótesis de clasificación, lo que permite realizar una ordenación de las hipótesis en función de esa probabilidad para un clasificador dado.

En este trabajo se presentan los resultados obtenidos en el cálculo del error de posición de los clasificadores bayesianos *Naive Bayes* y *Bayes Net*, entendiéndose como error de posición el que mide la posición media de la hipótesis correcta dentro de las propuestas por

clasificadores que presenten las mismas ordenadas.

Los datos usados son de distinta naturaleza, por una lado contamos con un conjunto de datos sintético, como es el *Cilindro, campana y embudo*; también hemos realizado pruebas con datos simulados correspondientes a un proceso continuo realizado en una planta piloto.

Dentro de cada conjunto de datos se ha trabajado con las series temporales sin preprocesar y con características que capturan propiedades interesantes de las series. Estas características han sido propuestas en otros artículos [1,8], en los que se aplican técnicas de aprendizaje diferentes a las usadas en la presente experiencia.

Otros trabajos que utilizan técnicas de aprendizaje automático en la diagnosis se pueden encontrar en [5, 13, 11, 12, 9].

El resto del artículo se organiza como sigue. La sección 2 realiza una introducción al método de aprendizaje bayesiano y de redes bayesianas. La sección 3 explica la metodología seguida en la generación de los experimentos de clasificación, tanto con las series temporales sin preprocesar, como con características que capturan propiedades importantes de las mismas. La sección 4 presenta los diferentes tipos de datos correspondientes a las series temporales usadas. La sección 5 incluye los resultados experimentales logrados. La sección 6 presenta las conclusiones más relevantes.

2. Redes Bayesianas en la Clasificación de Series

2.1. Aprendizaje Bayesiano

La clasificación bayesiana es un método basado en estadísticos. Su funcionamiento usa el cálculo de probabilidades a partir del teorema de Bayes, presentado en la ecuación 1.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (1)$$

Donde:

- $P(h)$ es la probabilidad a priori de la hipótesis h . En nuestro caso sería la probabilidad de que se diera una determinada clasificación de la serie.
- $P(D)$ es la probabilidad de observar el conjunto de entrenamiento D , cuando es usado para clasificar.
- $P(D|h)$ es la probabilidad de observar el conjunto de entrenamiento D en un universo donde se verifica la hipótesis h .
- $P(h|D)$ es la probabilidad a posteriori de h , cuando se ha observado el conjunto de entrenamiento D .

El aprendizaje bayesiano puede verse como el proceso de encontrar la hipótesis más probable, dado un conjunto de ejemplos de entrenamiento D y un conocimiento a priori sobre la probabilidad de cada hipótesis [7]. El modo de aplicación del teorema de Bayes para la clasificación es el cálculo de la hipótesis con mayor probabilidad a posteriori, como se muestra en la ecuación 2.

$$h_{MAP} \equiv \arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \quad (2)$$

Siendo D los datos de entrenamiento y h cada una de las hipótesis, en nuestro caso modos de fallo, que tenemos para clasificar. El subíndice MAP se corresponde con *Máximo a posteriori*. De este modo, clasificamos la instancia como aquella que tiene una mayor probabilidad a posteriori. Podemos observar que si tenemos en cuenta el valor de esta probabilidad para todas las hipótesis, en lugar de centrarnos sólo en la más probable, podemos obtener una ordenación de las hipótesis en función de dicha probabilidad.

La aproximación más simple para la aplicación de este método de clasificación es la

conocida como *Naive Bayes*, que asume que cada uno de los atributos de una instancia es independiente del resto. De este modo podemos calcular las probabilidades a posteriori siguiendo la aproximación de la ecuación 3.

$$P(a_1, a_2 \dots a_n | v_j) = \prod_i P(a_i, v_i) \quad (3)$$

Siendo v_j cada una de las clases que queremos clasificar y a_i los valores de los datos con los que clasificamos.

Dentro de las características más importantes de este tipo de aprendizaje está que cada ejemplo de entrenamiento afecta a la probabilidad de las hipótesis. Esto es más efectivo que descartar directamente las hipótesis incompatibles. además se puede incluir conocimiento a priori, como la probabilidad de cada hipótesis y la distribución de probabilidades de los ejemplos. También es sencillo asociar un porcentaje de confianza a las predicciones, y combinar predicciones en función de esos porcentajes.

Como dificultades que presenta este método tenemos la necesidad de un conocimiento a priori y el alto coste computacional.

A pesar de ser este un método en el que se tiene una restricción tan fuerte como la independencia de atributos, se pueden obtener clasificadores precisos aun cuando no se cumple ésta [4].

2.2. Redes Bayesianas

En general, la asunción de independencia entre los atributos es demasiado restrictiva, por lo que se plantea como alternativa el uso de redes bayesianas. Una red bayesiana representa la distribución de probabilidades conjunta para un conjunto de variables [7]. En ellas podemos establecer dependencias entre los atributos mediante un grafo dirigido acíclico, en el que los arcos representan las dependencias entre las

variables, y los nodos las variables, siendo asignado a cada uno de los nodos del grafo una tabla de probabilidades condicionadas. De este modo podemos determinar posibles dependencias entre los atributos de las instancias haciendo una aproximación menos restrictiva que en el caso de Naive Bayes.

A partir de una red bayesiana, se calculan las probabilidades a posteriori usando la ecuación 4:

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | \text{Padres}(Y_i)) \quad (4)$$

En esta expresión $\text{Padres}(Y_i)$ se refiere a los inmediatos predecesores de Y_i en la red bayesiana. Los valores de $P(y_i | \text{Padres}(Y_i))$ son los que se almacenan en la tabla de probabilidades condicionadas asociada al nodo Y_i .

Dentro de las redes bayesianas debemos diferenciar entre dos tipos de aprendizaje:

1. **Aprendizaje estructural:** Se refiere al aprendizaje de la estructura (dependencia) gráfica de la red.
2. **Aprendizaje Paramétrico:** Se refiere al aprendizaje de las probabilidades. En el lenguaje de los estadísticos, el aprendizaje paramétrico se llama estimación.

Todo método de aprendizaje de este tipo consta de dos elementos:

1. Una *medida de calidad*, usada para decidir cuál es la mejor de un conjunto de redes bayesianas. Este tipo de medidas de calidad pueden ser de diferentes tipos. Así distinguimos entre medidas de calidad locales y globales [2]. Las locales son aquellas basadas en métricas en las que la calidad global de la red puede ser descompuesta como el sumatorio o producto de la calidad de cada uno de sus nodos. Las globales son aquellas que calculan la medida de calidad a partir de toda la red.
2. Un *algoritmo de búsqueda*, que se usa para seleccionar un conjunto de redes bayesianas

de alta calidad, para tomar la mejor. Se ha de tener en cuenta que el número de redes que podemos llegar a tener para un mismo conjunto de variables es muy alto.

Por lo tanto, para aprender la estructura de la red y la probabilística de las redes bayesianas, necesitamos seguir dos etapas: elegir una medida de calidad y un algoritmo de búsqueda y usar el algoritmo de búsqueda para seleccionar una red. Esto requiere estimar los parámetros de la red resultante.

Las redes bayesianas nos permiten la definición de dependencias entre los atributos de los datos usados para la clasificación, además de ofrecer una estructura visual de mejor comprensión en el clasificador obtenido.

3. Método aplicado

En el presente trabajo se ha calculado el error de posición para los algoritmos *NaiveBayes* y *BayesNet* en los experimentos realizados sobre los datos *CBF* y para los simulados de una planta piloto. Ambos algoritmos han sido proporcionados por WEKA [14]. El algoritmo de redes bayesianas fue usado con una medida de calidad local especificada más adelante, usando el algoritmo de búsqueda K2 [3] y como estimador de parámetros el denominado *SimpleEstimator*. Para el cálculo del error de posición fue necesaria la modificación de la clase *Evaluation* de WEKA, del paquete *weka.classifiers*, con el fin de conseguir esta medida. Para un ejemplo dado, el error de posición esta entre 0 y 1. Se considera que la salida del clasificador es una ordenación de las clases. En concreto, el error de posición es la posición que ocupa la clase correcta menos uno dividido por el numero de clases menos uno, ecuación que se muestra a continuación.

$$e_{\text{posicion}} = \frac{P_{\text{clasecorrecta}} - 1}{N_{\text{clases}} - 1} \quad (5)$$

El algoritmo K2 es propuesto por Cooper y Herskovits. Este método comienza con la red más simple posible, es decir, una red sin arcos, y supone que los nodos están ordenados. Para cada variable X_i el algoritmo añade a su conjunto de padres Π_i el nodo con número menor de X_i que conduce a un máximo incremento de la calidad correspondiente a la medida de calidad elegida para el proceso de búsqueda. El proceso se repite hasta que, o bien no se incrementa la calidad, o se llega a una red completa. Hemos ejecutado este algoritmo a partir de una red vacía, como proponen en su trabajo Cooper y Herskovits [3]. También hemos usado la variación propuesta en el algoritmo de WEKA [14] en la que se toma como primer atributo la clase, Esta variación es denominada por WEKA *initAsNaive*. La medida de calidad usada ha sido la bayesiana implementada en WEKA [2, 14].

Una vez lograda la estructura, la estimación de las probabilidades ha sido realizada con el estimador simple *SimpleEstimator* [2].

Para poder usar los algoritmos de redes bayesianas que han sido utilizados, los datos deben ser discretos, por lo que las series temporales han sido discretizadas por el propio algoritmo. Además las instancias no deben tener valores desconocidos, algo que ya ocurría con nuestros datos.

3.1. Uso de características

Además de usar clasificadores bayesianos sobre los datos originales, se considera como alternativa la selección previa de características que capturen propiedades interesantes de las series. Estas características ya han sido propuestas en anteriores trabajos [1, 8]. Las que hemos usado están basadas en intervalos de dos tipos: absolutos y relativos. Consideramos un predicado basado en intervalos absoluto el predicado *media*; y un predicado basado en intervalos relativo el predicado *desviación*.

Este tipo de predicados tienen en cuenta alguna propiedad que se cumple en un intervalo de la serie. La distinción se realiza en función de

si la propiedad establecida sobre el intervalo depende de valores absolutos o de los relativos.

Estas características nos dan mayor información sobre el funcionamiento del proceso. De este modo, podemos pensar que los resultados mejorarán al realizar este tratamiento previo de los datos, en nuestro caso las series temporales. En [8] se describe cómo seleccionar características adecuadas para su uso en un sistema de aprendizaje.

4. Conjuntos de datos

En esta sección se describen los conjuntos de datos con los que se han probado los métodos de clasificación bayesiana usados.

4.1. Conjuntos de datos sintéticos

El uso de datos artificiales es discutible en tanto en cuanto los clasificadores obtenidos no tienen utilidad práctica. Por ejemplo, algunas definiciones de Minería de Datos establecen explícitamente el requisito de que los datos con los que se trabaja deben corresponderse con observaciones.

A pesar de esto, el uso de este tipo de datos forma parte de la tradición del área del Aprendizaje Automático. La justificación de su uso viene dada porque pueden permitir alcanzar una mayor comprensión de los distintos métodos de aprendizaje considerados.

El conjunto de datos sintético usado en el presente trabajo es el *Cilindro, campana y embudo*, este dominio artificial fue introducido por Saito [10] y utilizado por, entre otros, Kadous [6]. La tarea es distinguir tres clases: cilindro (*c*), campana (*b*) o embudo (*f*).

Se generaron 266 ejemplos de cada clase.

4.2. Conjuntos de datos simulados

Se han usado series temporales simuladas del funcionamiento de una planta piloto. Estos datos están formados por varias series temporales asociadas a un mismo experimento,

lo cual, como ya hemos dicho en la introducción, aumenta la dificultad de su clasificación. La planta simulada se muestra en la figura 1.

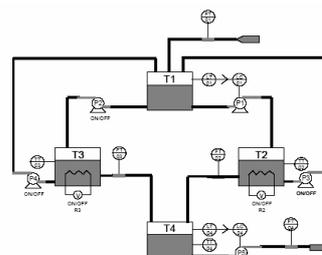


Figura 1. Esquema de la planta

Dentro de las simulaciones disponemos de las medidas de los niveles de los tanques T_1 y T_4 , los valores de los controladores PID de las bombas P_1 y P_2 , el flujo de entrada al tanque T_1 , el flujo de salida de los tanques T_2 , T_3 y T_4 y la temperatura en los tanques T_2 , T_3 y T_4 . Esto hace un total de 11 variables distintas.

La Tabla 1 muestra cuáles son los modos de fallo que se tienen en cuenta.

Identificador	Componentes	Tipo
MF01	T1	Fuga pequeña en T1
MF02	T1	Fuga grande en T1
MF03	T1	Bloqueo tubería T1 (salida izquierda)
MF04	T1	Bloqueo tubería T1 (salida derecha)
MF05	T3	Fuga en T3
MF06	T3	Bloqueo tubería T3 (salida derecha)
MF07	T2	Fuga en T2
MF08	T2	Bloqueo tubería T2 (salida izquierda)
MF09	T4	Fuga en T4
MF10	T4	Bloqueo tubería T4 (salida derecha)
MF11	P1	Rotura / pérdida de rendimiento
MF12	P2	Rotura / pérdida de rendimiento

MF13	P5	Rotura / pérdida de rendimiento
MF14	R2	Rotura resistencia en tanque T2

Tabla 1. Modos de Fallo considerados.

Para cada modo de fallo se realizaron 20 simulaciones. En cada una de ellas se simula el proceso durante 15 minutos de tiempo real. El periodo de muestreo es de un segundo, por lo que la longitud de las series de 900. Dado que en una resolución tan fina no es necesario que el tiempo de los métodos de inducción depende de la longitud de la serie se comprimieron las series a una longitud de 300 tomando la media de los datos tres a tres.

5. Resultados

Las pruebas realizadas han sido llevadas a cabo tanto con los datos de las series temporales sin preprocesar, como con estos mismos datos preprocesados, es decir con las características importantes de las series expuestas en el apartado 3.1.

En el caso de los datos CBF se contaba con unas 700 series en cada experimento. En los datos simulados se contaban con 20 conjuntos de series por experimento, no olvidemos que en este caso cada clase tiene en cuenta varias series temporales asignadas a diferentes variables.

Una vez creadas las redes se realizó validación cruzada con 10 grupos.

A continuación mostramos los resultados obtenidos en el cálculo del error de posición. En primer lugar mostramos los correspondientes al trabajo realizado con las series temporales sin preprocesar.

Después presentaremos los resultados con los datos preprocesados, usando las características importantes de las series.

En las siguientes tablas se muestra la media de los errores de posición para cada uno de los ejemplos usados. Recordamos que, como ya se dijo en la sección 3, el error de posición esta

III Taller de Minería de Datos y Aprendizaje

entre 0 y 1. En concreto, el error de posición es la posición que ocupa la clase correcta menos 1 dividido por el número de clases menos 1.

5.1. Resultados sobre las series temporales

A continuación se presentan los resultados obtenidos con las series temporales sin preprocesar. En la tabla 2 se muestra la tasa de error y en la tabla 3 el error de posición.

Tasa de error	Naive Bayes	Red Bayesiana (K2, SimpleEstimador)	Red Bayesiana (K2 iniciado a Naive, SimpleEstimador)
CBF	11.77%	7.52%	5.77%
Planta	16.41%	-	-

Tabla 2. Tasa de error sobre las series temporales.

Error de posición	Naive Bayes	Red Bayesiana (K2, SimpleEstimador)	Red Bayesiana (K2 iniciado a Naive, SimpleEstimador)
CBF	0.060	0.046	0.038
Planta	0.154	-	-

Tabla 3. Error de posición sobre las series temporales.

Para los datos CBF, estos resultados muestran que el error de posición mejora al usar redes bayesianas, sobre el uso del algoritmo *Naive Bayes*, al igual que ocurre con la tasa de error. El estudio de dicho error nos dice que como máximo se encuentra la clasificación correcta en la segunda hipótesis, teniendo en cuenta una ordenación basada en las probabilidades a posteriori. Hemos de tener en cuenta que en este caso sólo hay tres clases, pero

los resultados nos dicen que en la mayoría de los casos no llegaríamos a valorar en último lugar la clase correcta.

En los experimentos realizados sobre los datos de la planta tenemos 14 clases. Por término medio la hipótesis correcta es clasificada en tercer lugar en caso de error. Sólo hemos podido conseguir los resultados para el algoritmo *Naive Bayes*, debido a que al aplicar redes bayesianas a los datos de la planta, la alta carga computacional exigida, no permitió realizar los experimentos.

5.2. Resultados sobre las características

En las siguientes tablas se muestran la tasa de error y el error de posición respectivamente para los datos preprocesados.

Tasa de Error	Naive Bayes	Red Bayesiana (K2, SimpleEstimator)	Red Bayesiana (K2 iniciado a Naive, SimpleEstimator)
CBF	3.61%	2.01%	1.12%
Planta	1.79%	2.86%	1.79%

Tabla 4. Tasa de error con los datos preprocesados

Error de posición	Naive Bayes	Red Bayesiana (K2, SimpleEstimator)	Red Bayesiana (K2 iniciado a Naive, SimpleEstimator)
CBF	0.016	0.012	0.008
Planta	0.016	0.007	0.005

Tabla 5. Error de posición sobre los datos preprocesados

Vemos como en el caso de los datos *CBF* la tasa de error es más baja cuando se usan las redes bayesianas, mejorando entre ambos tipos

de red en el caso de iniciar la red a *Naive*. Lo mismo ocurre con el error de posición. Por lo que podemos decir que el uso de las características mejoran los resultados de los clasificadores obtenidos en el caso de los datos sin preprocesar.

También mejoran los resultados al usar las características con los datos de la planta. En este tipo de datos vemos como la tasa de error empeora al usar redes bayesianas. La explicación a este comportamiento puede ser justificada por el funcionamiento del algoritmo K2 [3], que toma los atributos de los conjuntos de entrenamiento de manera ordenada, lo que en el caso de contar con datos de diferentes series puede llevar a situaciones en las que la red alcanzada no sea la mejor posible.

Por otra parte el error de posición sí que mejora al usar redes bayesianas, lo que hace que en el caso de equivocarnos en la hipótesis seleccionada por el clasificador, las siguientes opciones sean más fiables.

6. Conclusiones

En el presente trabajo se ha estudiado el error de posición de métodos de clasificación bayesianos aplicados en series temporales. El error de posición determina la posición media de la hipótesis correcta en clasificadores que dan como salida las posibles clases de manera ordenada. Esta es una de las características de los clasificadores bayesianos estudiados, que permiten la ordenación de las hipótesis en función de la probabilidad a posteriori calculada para cada una.

Los errores de posición son mejores en el caso de usar redes bayesianas, y también se observa una notable mejoría al usar las características de las series como entrada a estos clasificadores.

En el peor de los casos, obtenido con el clasificador bayesiano sobre las series sin preprocesar de la planta, el clasificador acierta como máximo en la tercera hipótesis, siendo el número de las mismas para ese experimento de 14.

Podemos por lo tanto decir que los clasificadores bayesianos, no solamente obtienen buenos resultados en la clasificación de este tipo de series, sino que además, en los casos en los que no hallan la hipótesis correcta proporcionan buenas opciones alternativas, tomando como éstas las siguientes con mayor probabilidad a posteriori.

Agradecimientos

Este trabajo ha sido financiado por el proyecto del MCyT DPI2002-01809.

Referencias

- [1] Carlos Alonso, Juan J. Rodríguez y Belarmino Pulido. Enhancing consistency based diagnosis with machine learning techniques. Current Topics in Artificial Intelligence: 10th Conference of the Spanish Association for Artificial Intelligence, volume 3040 of Lecture Notes in Artificial Intelligence, páginas 312-321. Springer, 2004.
- [2] Remco R. Bouckaert. Bayesian Network Classifiers in Weka. <http://citeseer.ist.psu.edu/705669.html>
- [3] Gregory F. Cooper, Edward Herskovits: A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning 9: 309-347 (1992)
- [4] Domingos, P. & Pazzani, M. (1996), Beyond independence: conditions for the optimality of the simple Bayesian classifier, in L. Saitta, ed., 'Machine Learning: Proceedings of the Thirteenth International Conference', Morgan Kaufmann, pp. 105--112. <http://citeseer.ist.psu.edu/domingos96beyond.html>
- [5] C Feng. Inducting temporal fault diagnostic rules from a qualitative model. In S. Muggleton, editor, Inductive Logic Programming. Academic Press, 1992.
- [6] Mohammed Waleed Kadous. Learning comprehensible descriptions of multivariate time series. In Ivan Bratko and Saso Dzeroski, editors, Proceedings of the 16th International Conference of Machine Learning (ICML-99). Morgan Kaufmann, 1999. <http://ftp.cse.unsw.edu.au/pub/users/waleed/izm199.ps.gz>
- [7] Mitchell, T. Machine Learning. McGraw Hill. 1997.
- [8] Juan J. Rodríguez, Carlos J. Alonso y Henrik Boström. Boosting interval based literals. Intelligent Data Analysis, 5(3):245-262, 2001.
- [9] Davide Roverso. Fault diagnosis with the aladdin transient classifier. System Diagnosis and Prognosis: Security and Condition Monitoring Issues III, AeroSense2003, Aerospace and Defense Sensing and Control Technologies Symposium, 2003. http://www.ife.no/media/1167_Aladdin_AeroSense_0R14.pdf
- [10] Naoki Saito, Ronald R. Coigman. Local discriminant bases and their applications. J. Mathematical Imaging and Vision, 5(4):337-348, 1995. http://math.ucdavis.edu/~saito/publications/ldb_jmiv.html
- [11] Derek Sleeman, F. Mitchell, and R. Milne. Applying KDD techniques to produce diagnostic rules for dynamic systems. Technical report AUCS/TR9604, Department of Computing Science. University of Aberdeen, 1996. <http://ftp.csd.abdn.ac.uk/pub/reports/tr9604.ps>
- [12] Antonio J. Suárez, Pedro J. Abad, J. A. Ortega y Rafael M. Gasca. Diagnosis progresiva en el tiempo de sistemas dinámicos. IV Jornadas de ARCA, Sistemas Cualitativos y Diagnosis, JARCA'02, 2002.
- [13] V. Venkatasubramanian and K. Chan. A neural network methodology for process fault diagnosis. AIChE J., 35 (12) :1993-2002, 1989.
- [14] Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco, 2000