

Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos

Anibal Bregón, Arancha Simón, Carlos Alonso,
Belarmino Pulido, Isaac Moro

Dept. de Informática
ETS Ingeniería Informática
Univ. de Valladolid
47011 Valladolid

Juan José Rodríguez
Lenguajes y Sistemas Informáticos
Escuela Politécnica Superior
Univ. de Burgos
Burgos
jjrodriguez@ubu.es

anibbre@lab.fi.uva.es, (arancha,calonso,belar,isaac)@infor.uva.es

Resumen

El diagnóstico mediante la clasificación de fallos en plantas industriales es un cuerpo de investigación que recibe continuamente atención. En algunos dominios estos fallos vienen descritos por series temporales. En este artículo vamos a introducir algunas técnicas para la clasificación de distintos modelos de fallo obtenidos por medio de una planta de laboratorio. Presentamos un marco computacional para resolver problemas de clasificación de fallos usando Razonamiento Basado en Casos. Este artículo ilustra diferentes técnicas para la reutilización y recuperación de casos (como por ejemplo Alineamiento Dinámico Temporal o distancia Euclídea), evaluando y comparando los resultados.

1. Introducción

El Razonamiento Basado en Casos (CBR) es una metodología de Inteligencia Artificial para realizar aprendizaje que ha logrado buenos resultados en muchos campos de aplicación. Se han desarrollado varios trabajos en los que se ha aplicado CBR para determinar el estado de operación de una planta industrial [4, 6], y para llevar a cabo planificación, diagnóstico, mantenimiento y gestión de calidad en la industria [5]. En algunos de estos trabajos los

autores han manejado series temporales, y han diseñado representaciones de casos para gestionarlas y lograr una mejor recuperación del caso.

En este artículo, aplicamos una metodología CBR a un problema de clasificación de fallos en una planta industrial. El sistema CBR trabaja en paralelo con un sistema de diagnóstico basado en modelos que efectúa detección y localización de fallos [12]. Una vez que un fallo es detectado, el sistema CBR proporcionará una pista sobre el fallo candidato más probable.

Nuestro conjunto de datos está compuesto por series temporales. Comparamos la precisión obtenida usando datos sin preprocesado como representación del caso con la obtenida usando una selección de características. Esta última representación fue diseñada por algunos autores de este artículo en otros trabajos [14] y la hemos aplicado ahora a CBR.

El artículo está organizado de la siguiente manera: la Sección 2 describe la metodología CBR, el algoritmo de recuperación y las medidas de similitud que hemos usado; la Sección 3 introduce brevemente la planta industrial con sus clases de fallos. En la Sección 4 presentamos el estudio experimental: la descripción del conjunto de datos y el método de evaluación. La sección 5 muestra los resultados y la discusión, y finalmente, la sección 6 se dedica a las conclusiones y trabajo futuro.

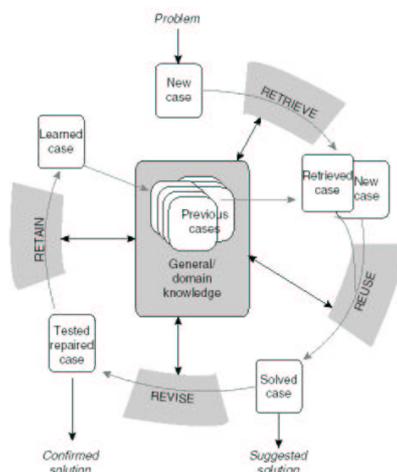


Figura 1: El ciclo CBR.

2. Clasificación de fallos usando razonamiento basado en casos

2.1. Introducción. El ciclo CBR.

El Razonamiento Basado en Casos (CBR) es un método para resolver problemas recordando situaciones previas similares y reutilizando la información y el conocimiento sobre esa situación [10, 11]. La idea original básica de Riesbeck y Schank [13] es la siguiente:

CBR resuelve problemas nuevos adaptando las soluciones dadas a otros resueltos con anterioridad.

Aamod y Plaza [1] describen CBR como un proceso que consta de cuatro pasos:

- RECUPERAR el caso o casos más similares.
- REUTILIZAR la información y el conocimiento de ese caso para resolver el problema.
- REVISAR la solución propuesta.
- GUARDAR las partes de esta experiencia que se consideren útiles para resolver futuros problemas.

Estos cuatro pasos se muestran en la Figura 1.

2.2. Recuperación de casos.

La primera tarea en el ciclo de CBR es la recuperación de uno o más casos similares de la base de casos donde está almacenada la experiencia pasada en forma de casos [2]. Un caso está formado por varios atributos que dan una descripción del problema y una solución para el caso. Para poder llevar a cabo la tarea de recuperación es necesario tener un algoritmo de recuperación y una medida de similitud que serán usados para obtener un conjunto de casos similares. En nuestro trabajo hemos elegido el algoritmo de los K -vecinos como algoritmo de recuperación y tres medidas de similitud: distancia Euclídea, distancia de Manhattan y Alineamiento Dinámico Temporal(DTW).

2.2.1. Algoritmo de los K -vecinos.

Sea $X = \{x^i = (x_1^i, \dots, x_p^i), i = 1, \dots, N\}$ una colección de ejemplos de entrenamiento de dimensión p y sea $C = \{C_1, \dots, C_M\}$ un conjunto de M clases. Supongamos que cada ejemplo x^i tiene una etiqueta de clase $L_i \in \{1, \dots, M\}$ que indica con certeza su pertenencia a una clase en C . Supongamos también que x^S es un nuevo ejemplo para clasificar. Clasificar x^S corresponde a asignarle una de las clases en C , esto es, decidir entre un conjunto de M

hipótesis: $x^S \in C_q, q = 1, \dots, M$. Sea Φ^S el conjunto de los K -vecinos más próximos de x^S en X . Para cualquier $x^i \in \Phi^S$, el conocimiento de $L_i = q$ puede considerarse como una evidencia que aumenta nuestra confianza de que x^S también pertenece a C_q . No obstante, esta evidencia no proporciona por sí misma 100 % de certeza. El clasificador de los K -vecinos más próximos (KNN), como sugurieron Duda y Hart [7], almacena los datos de entrenamiento, el par (X, L) . Los ejemplos son clasificados eligiendo la clase mayoritaria entre los K ejemplos más cercanos del conjunto de entrenamiento, de acuerdo a una medida de distancia [3]. En nuestros experimentos fueron probados varios valores de K (número de vecinos) $K = 1, 3, 5$, y se ha encontrado empíricamente que $K = 1$ ha proporcionado los mejores resultados para nuestro conjunto de datos.

2.2.2. Medidas de similitud

Distancia Euclídea. Es la distancia más común y la más usada en la mayoría de los sistemas CBR. Definimos la distancia Euclídea entre dos puntos x e y como: $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, donde x_i es la coordenada de x en la dimensión i y y_i es la coordenada de y en la dimensión i .

Distancia de Manhattan. La distancia de Manhattan entre dos puntos es la suma de los valores absolutos de las diferencias de sus componentes. Definimos la distancia de Manhattan entre un punto x y otro punto y como: $\sum_{i=1}^n |x_i - y_i|$, donde x_i es la coordenada de x en la dimensión i e y_i es la coordenada de y en la dimensión i .

Alineamiento dinámico temporal. El Alineamiento Dinámico Temporal [8, 6], abreviado DTW (*dynamic time warping*), es una técnica que permite obtener una medida de la semejanza entre dos secuencias que no están exáctamente alineadas en el eje temporal, y que pueden tener longitudes diferentes. Para este tipo de datos esta medida es bastan-

te más robusta que las medidas anteriores. A continuación, se introduce brevemente DTW:

Sean dos series temporales, Q y C , de longitudes n y m :

$$Q = q_1, q_2, \dots, q_i, \dots, q_n; \quad (1)$$

$$C = c_1, c_2, \dots, c_j, \dots, c_m. \quad (2)$$

Para alinear las dos secuencias usando DTW, se construye una matriz $n \times m$. Cada elemento (i, j) se corresponde con el alineamiento entre los puntos q_i y c_j . Un camino de alineamiento W es un conjunto de elementos de la matriz contiguos que define una correspondencia entre Q y C . El elemento k -ésimo de W se define como $w_k = (i, j)_k$. Por tanto:

$$W = w_1, w_2, \dots, w_k, \dots, w_K \quad (3)$$

$$\text{máx}(m, n) = K \leq m + n - 1. \quad (4)$$

La distancia Euclídea puede considerarse como un caso especial de DTW en el que el elemento k -ésimo de W se restringe a $w_k = (i, j)_k, i = j = k$.

En la aplicación considerada, las series son multivariable. Es decir, las series son multidimensionales. Para utilizar DTW con este tipo de datos consideramos las siguientes aproximaciones:

- Aplicar DTW para cada variable. La semejanza entre 2 series multivariable será la media de las semejanzas para cada variable.

Para calcular la distancia $d(q_i, c_j)$ entre los puntos q_i y c_j usamos tres métricas:

- Lineal: $|q_i - c_j|$
- Cuadrática: $(q_i - c_j)^2$
- Valle: $10 \times (1 - \exp(\frac{-(q_i - c_j)^2}{6}))$

- Aplicar DTW sólo una vez, considerando que los puntos de las series son multidimensionales, con tantas dimensiones como variables. La única distancia $d(q_i, c_j)$ considerada es la Euclídea.

2.3. Reutilización de casos.

El segundo paso en el ciclo de CBR es la tarea de reutilización. Aamodt y Plaza [1] describen esta tarea centrada en dos aspectos: (a) las diferencias entre el caso pasado y el actual, y (b) qué parte o partes del caso recuperado pueden ser transferidas al nuevo caso. En algunos casos la tarea de reutilización se reduce a copiar la solución pasada al nuevo caso, pero en otros casos esta solución no puede ser aplicada directamente y tiene que ser adaptada. En CBR podemos distinguir dos tipos de adaptación [16]:

- *Adaptación Estructural*: aplicar reglas de adaptación a la solución recuperada.
- *Adaptación Derivacional*: reutilizar las fórmulas que generaron la solución pasada para conseguir una nueva solución para el problema actual.

El método de reutilización usado en este trabajo está basado en el algoritmo de los K -vecinos más próximos. En nuestro trabajo podemos elegir el número de vecinos a usar y adaptar la solución del nuevo caso mediante votación de la solución de todos los casos recuperados.

2.4. Revisión de casos.

Una vez finalizada la tarea de reutilización, la solución del nuevo problema tiene que ser probada; este proceso de prueba se hace durante la tarea de revisión. En este paso, la solución generada en la tarea de reutilización se evalúa, y si el resultado es satisfactorio, el nuevo caso y la nueva solución para el caso se almacenan.

2.5. Almacenamiento de casos.

El almacenamiento es la última tarea del ciclo CBR. En este paso el nuevo caso y la solución para este caso (obtenida en la fase de reutilización) son almacenados con vistas a un posible uso futuro. Durante este proceso de aprendizaje, el sistema tiene que seleccionar qué información del caso almacenar, la forma en la

III Taller de Minería de Datos y Aprendizaje

que almacenar esta información y cómo indexar el caso en la estructura de memoria para una posterior recuperación.

3. Descripción de la planta de laboratorio

La planta de laboratorio se diseñó intentando reflejar características comunes a la mayoría de procesos industriales continuos. La Figura 2 nos muestra un esquema de la misma. Está formada por cuatro depósitos $\{T_1 \dots T_4\}$, cinco bombas $\{P_1 \dots P_5\}$, y dos controladores PI que actúan sobre las bombas $\{P_1, P_5\}$ para mantener el nivel de los depósitos próximos a su valor de consigna. Para controlar la temperatura de los depósitos $\{T_2, T_3\}$ se han usado dos resistencias, respectivamente.

En la planta tenemos el siguiente conjunto de sensores (variables medidas): los niveles de los depósitos T_1 y T_4 , el valor de la acción de control de los PI sobre las bombas, el flujo de entrada a T_1 , flujos de salida de los depósitos $\{T_2, T_3, T_4\}$, y las temperaturas en los depósitos $\{T_2, T_3, T_4\}$.

En esta planta se han considerado las siguientes familias de fallos para el protocolo descrito: 5 relacionados con fugas en los depósitos, 5 atasques en tuberías, 3 fallos en las bombas, y 1 fallo en la resistencia.

4. Estudio experimental

El estudio experimental ha sido realizado con un conjunto de datos procedentes de la simulación de una planta de laboratorio. Este conjunto de datos está formado por varios ejemplos obtenidos de simulaciones de diferentes tipos de fallos que pueden aparecer en la planta industrial previamente descrita. En este apartado vamos a describir estos conjuntos de datos, las tareas de preprocesado aplicadas y los métodos de evaluación empleados.

4.1. Descripción del conjunto de datos.

4.1.1. Datos sin preprocesado.

Durante la primera fase de desarrollo del proyecto se han usado datos sin preprocesado. Es-

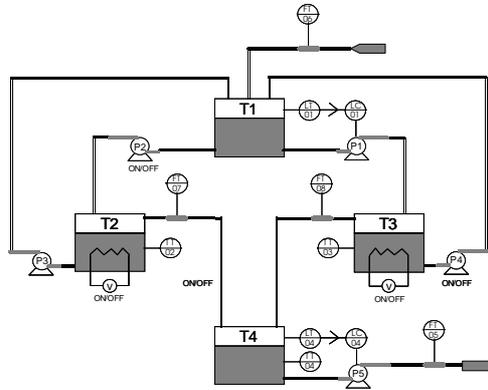


Figura 2: Esquema de la planta.

te conjunto de datos está formado por series temporales de datos que describen un fallo de la planta industrial. Hay catorce tipos de fallo diferentes y cada uno de ellos viene descrito por once series de trescientos elementos numéricos asociados a las once medidas de sensores disponibles. Nosotros hemos utilizado veinte ejemplos de cada modelo de fallo.

En nuestro sistema CBR, un caso va a estar compuesto por once series (las cuales van a describir un fallo) y el tipo de fallo, que será la solución que nuestro sistema trata de adaptar.

Puesto que la longitud de cada serie es 300, la descripción de cada caso consiste en 11×300 valores reales. La Figura 3 muestra un caso de cada modo de fallo.

Con la intención de hacer una mejor clasificación de fallos, vamos a aplicar algoritmos de filtrado y normalización sobre los datos originales.

Filtrado. Algunos valores de las series numéricas pueden exceder los límites permitidos de control, volumen, nivel o temperatura (bien por exceso o por defecto), ésta es la razón por la que aplicamos un proceso de filtrado antes de introducir los datos en la base de casos. Los valores máximos y mínimos permitidos son los siguientes:

- control: desde 0 a 1. 0 significa 0% de la acción de control, 1 significa 100%.

- flujo: desde 0 l/min hasta 5 l/min.
- nivel: desde 0% hasta 100% del nivel de cada tanque.
- temperatura: desde 0°C hasta 40°C

Normalización. Con el fin de decrementar los efectos producidos por la diferencia entre escalas realizamos un proceso de normalización previo a introducir los datos en la base de casos. Este proceso consiste en aplicar el método estadístico para conseguir llevar todos los datos a una media de 0 y una varianza de 1.

4.1.2. Selección de características.

Además de utilizar los datos en crudo, se consideró como alternativa el aplicar un preproceso de selección de características. En este trabajo las características seleccionadas son de los siguientes tipos:

- media(Ejemplo, Variable, Inicio, Fin).
- desviación(Ejemplo, Variable, Inicio, Fin).

Estas características evalúan una función (esto es, media o desviación), para una Variable del Ejemplo, en el intervalo dado por Inicio y Fin. Si se consideran todos los posibles intervalos, el número de posibles características es excesivo. La selección de algunas de estas características se realiza utilizando el método

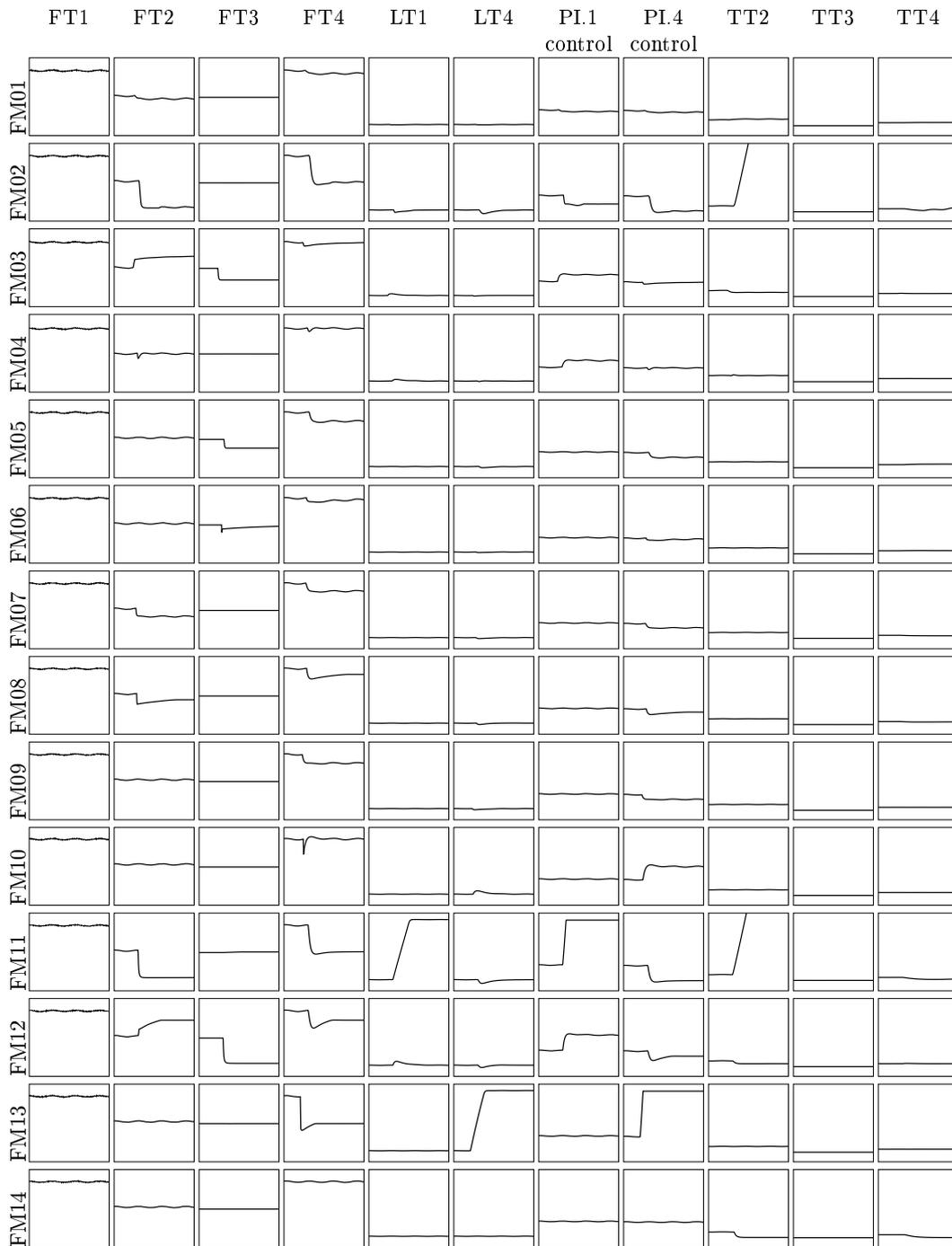


Figura 3: Ejemplos de casos. Cada fila muestra un caso de un modo de fallo. Cada columna muestra una de las variables

descrito en [14], que se apoya a su vez en el método denominado boosting [15]. En esta aplicación, el número de características seleccionadas es 100.

Uno de los objetivos que se persigue al seleccionar características es la reducción de la dimensionalidad. En los datos en crudo hay 11 series de 300 valores, mientras que los datos preprocesados sólo tienen 100 atributos. En consecuencia, los casos almacenados son mucho más compactos, lo que reduce considerablemente el tiempo necesario para encontrar los casos más similares. Por otro lado, a la hora de clasificar un ejemplo es necesario obtener los valores de las características para ese ejemplo.

4.2. Métodos de evaluación

4.2.1. Validación cruzada.

En la validación cruzada K-fold [3] un conjunto de datos D es dividido aleatoriamente en k subconjuntos mutuamente excluyentes D_1, \dots, D_k de aproximadamente igual tamaño. El clasificador es entrenado y probado k veces; cada vez $t \in 1, 2, \dots, k$ es entrenado sobre $D_i, i = 1, \dots, k$, con $i \neq t$ y probado sobre D_t . La estimación de la precisión de la validación cruzada es el número global de clasificaciones correctas dividido por el número de ejemplos del conjunto de datos.

Nosotros usamos validación cruzada estratificada para estimar la precisión del método de clasificación. El conjunto de datos fue dividido en subconjuntos de igual tamaño. La proporción de las clases es igual en cada partición (la misma que en el conjunto de datos original) porque este procedimiento proporciona resultados más precisos que una validación cruzada simple [9]. En nuestros experimentos tenemos los datos divididos en 10 subconjuntos de igual tamaño.

5. Resultados

Hemos realizado varios experimentos aplicando todas las técnicas descritas de recuperación y reutilización tanto con datos sin preprocesado como con selección de características.

5.1. Datos sin preprocesado

Hemos aplicado DTW unidimensional con diferentes métricas (lineal, cuadrática y valle), DTW multidimensional, distancia Euclídea y distancia de Manhattan. El Cuadro 1 muestra el éxito en la clasificación logrado usando diferente número de vecinos (1, 3 y 5). Los resultados han sido obtenidos usando validación cruzada estratificada con 10 subconjuntos de igual tamaño provenientes del conjunto de datos sin preprocesado.

La mejor medida de similitud ha resultado ser DTW unidimensional con métricas Cuadrática o Valle. Incrementando el número de vecinos, la tasa de aciertos desciende sistemáticamente. Una posible explicación de esta conducta podría ser el hecho que el número de ejemplos de cada clase, 20, es pequeño considerando el número de clases, 14, y la dimensionalidad de los datos. Planeamos volver a probar estas hipótesis con un conjunto mayor de datos.

Como era de esperar, los resultados obtenidos usando DTW (unidimensional o multidimensional) son mejores que los obtenidos usando las distancias Euclídea o de Manhattan. DTW es una medida de disimilitud mucho más robusta para series temporales, permitiendo que formas similares coincidan incluso si se encuentran desfasadas en su eje temporal [8]. La distancia Euclídea se ve mucho más afectada por pequeños cambios en el eje temporal. No obstante, los resultados para distancia Euclídea y de Manhattan están bastante próximos a los resultados de las distintas variantes de DTW.

5.2. Selección de características.

En este apartado vamos a mostrar los resultados obtenidos, usando validación cruzada estratificada, al aplicar el método de selección de características. El Cuadro 2 muestra el éxito en la clasificación logrado usando las distancias Euclídea y de Manhattan, ambas con diferente número de vecinos (1, 3 y 5).

Como podemos observar, los resultados son mejores usando selección de características que usando datos sin preprocesado. La mejor tasa

Cuadro 1: Tasas de acierto obtenidas al utilizar DTW unidimensional, DTW multidimensional, distancia Euclídea y Manhattan sobre los datos en crudo.

Número de vecinos	1-D DTW			M-D DTW	Distancia Euclídea	Distancia Manhattan
	Lineal	Cuadrática	Valle			
1	89.2 %	91.4 %	91.4 %	90.4 %	89.3 %	88.2 %
3	86.4 %	88.6 %	88.6 %	87.1 %	86.4 %	85.4 %
5	85.4 %	85.0 %	85.0 %	85.4 %	85.7 %	84.6 %

Cuadro 2: Tasas de acierto obtenidas al utilizar selección de características con las distancias Euclídea y Manhattan.

Número de vecinos	Distancia Euclídea	Distancia de Manhattan
1	91.8 %	96.1 %
3	87.1 %	93.6 %
5	86.8 %	91.1 %

de acierto obtenida con los datos sin preprocesado es del 91.4% mientras que la mejor usando selección de características es del 96.1%. Por lo tanto, el error ha sido reducido a menos de la mitad. Al igual que ocurría al usar datos sin preprocesado, incrementar el número de vecinos decrementa el acierto.

6. Conclusiones y trabajo futuro

Se ha desarrollado un sistema CBR para la identificación de modelos de fallos en procesos continuos. El sistema usa el algoritmo de los K -vecinos para la recuperación de casos. Puede ser usado con distintas métricas, tales como la distancia de Manhattan, la distancia Euclídea y DTW tanto unidimensional como multidimensional.

El sistema permite usar una representación compacta de los casos, donde cada uno viene definido por los valores de varias características. Cada característica evalúa una función en un intervalo dado. En este trabajo las funciones consideradas son la media y la desviación.

El sistema ha sido validado usando un conjunto de datos obtenido de las simulaciones de modelos de fallo en una planta de laboratorio. Se ha comparado la precisión de los clasificado-

res usando las distintas métricas con los datos sin preprocesado y usando selección de características. Al usar los datos sin preprocesado, considerando que son series temporales, DTW obtiene mejores resultados que las distancias que no son específicas para este tipo de datos.

Los resultados experimentales demuestran la validez del método de selección de características propuesto. La precisión ha sido incrementada desde 91.4% hasta 96.1%. Otro asunto es el tiempo de ejecución. Primero, las características deben ser seleccionadas, aunque este proceso puede realizarse fuera de línea. Para clasificar un nuevo caso, deben evaluarse las características. El tiempo de este proceso es lineal con el número de características (100 para el ejemplo actual) y la longitud de las series. Sin embargo, ahora cada caso es mucho más compacto, así que el tiempo necesario para encontrar los vecinos más próximos se ve significativamente reducido.

La validación del método de selección de características ha sido considerado desde un punto de vista de aprendizaje automático: todos los datos de entrenamiento están disponibles, así que las características son seleccionadas usando todos los datos de entrenamiento. Sin embargo, en un sistema CBR nuevos casos son incorporados al sistema en cualquier momento. Estos nuevos casos serán representados mediante un conjunto de características que han sido seleccionadas sin tenerles en cuenta. Esto podría causar una degradación del rendimiento del sistema. Por lo tanto, pensamos que sería interesante probar el sistema bajo estas condiciones.

Además, nuestra intención es proseguir con la validación de esta aproximación sobre un conjunto experimental mayor.

Referencias

- [1] A. Aamodt and E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications. IOS Press, Vol. 7: 1*, pages 39–59, 1994.
- [2] K. Althoff and B. Bartsch-Spörl. Decision Support for Case-Based Applications. *Special issue on case-based decision support.*, pages 6–14, 1996.
- [3] A. D. Anastasiadis, G. D. Magoulas, and X. Liu. Classification of Protein Localisation Patterns via Supervised Neural Network Learning. In *Proceedings of the Fifth Symposium on Intelligent Data Analysis (IDA-03), Berlin, Germany, August 2003*, pages 430–439, 2003.
- [4] D. M. Brann, D. A. Thurman, and C. M. Mitchell. Case-Based Reasoning as a Methodology for Accumulating Human Expertise for Discrete System Control. In *Proceedings of the IEEE Int. Conf. on SMC, Vancouver, B. C., Canada*, pages 4219–4223, 1995.
- [5] J. Britanik and M. Marefat. Case-Based Manufacturing Process Planning with integrated support for knowledge sharing. In *IEEE Int. Symp. on Assembly Task Planning*, pages 107–112, 1995.
- [6] J. Colomer, J. Melendez, and F. I. Gameiro. Qualitative representation of process trends for situation assessment based on cases. In *15th Triennial World Congress of the International Federation of Automatic Control, Barcelona, Spain*, 2002.
- [7] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [8] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.
- [9] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence*, pages 223–228, 1995.
- [10] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, 1993.
- [11] D. B. Leake. CBR in Context: The present and Future. *Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park: AAAI Press.*, 1996.
- [12] B. Pulido, C. Alonso, and F. Acebes. Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In *Engineering of Intelligent Systems. Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2001)*, volume 2070 of *Lecture Notes in Artificial Intelligence*, pages 135–144, Budapest, Hungary, 2001.
- [13] C. K. Riesbeck and R. C. Shank. Inside Case Based Reasoning. *Lawrence Erlbaum Associates, Hillsdale, NJ, US.*, 1989.
- [14] Juan J. Rodríguez and Carlos J. Alonso. Support vector machines of interval-based features for time series classification. In *Research and Development in Intelligent Systems XXI: Proceedings of AI-2004*, pages 244–257. Springer, 2004.
- [15] Robert E. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Non-linear Estimation and Classification*, 2002. <http://www.cs.princeton.edu/~schapire/papers/msri.ps.gz>.
- [16] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers, 1997.