# Una primera aproximación para la extracción de patrones emergentes en flujos continuos de datos

A.M. Garcia-Vico, C.J. Carmona, P. González, M.J. del Jesus
*Departamento de Informática, Data Science and Computational Intelligence*
*Universidad de Jaén*, Jaén, España
{agvico|ccarmona|pglez|mjjesus}@ujaen.es

***Resumen*—A día de hoy, la cantidad de información proveniente de fuentes que emiten constantemente datos es inmensa, por lo que se hace necesario la extracción automática de conocimiento para la mejora de servicios en la vida cotidiana de las personas. La búsqueda de patrones emergentes permite la descripción de las características discriminativas entre clases o la descripción de tendencias emergentes en los datos. En este trabajo se presenta un nuevo enfoque basado en un sistema evolutivo difuso para la extracción de patrones emergentes en flujos continuos de datos. Los resultados del estudio experimental muestran unos resultados prometedores para la extracción de este tipo de conocimiento en el ámbito de la minería de flujo de datos.**

***Index Terms*—Descubrimiento de reglas descriptivas supervisadas, minería de patrones emergentes, algoritmos evolutivos multi-objetivo, minería de flujo de datos.**

## I. Introducción

Vivimos en la era de la información. A día de hoy, el desarrollo en las tecnologías de la información y la comunicación ha permitido un aumento exponencial en la cantidad de datos generados. Esto se debe principalmente al abaratamiento de los sistemas de almacenamiento y sensores generadores de datos [1]. Toda esta cantidad de datos contiene conocimiento muy relevante para las empresas para poder mejorar sus servicios [3] lo que ha propiciado el desarrollo en los últimos años de técnicas de extracción de conocimiento en estos enormes volúmenes de información heterogénea en lo que se conoce como *Big Data* [2]. Sin embargo, existen ámbitos de trabajo, como por ejemplo en gestión de energía [4], donde los datos muy antiguos son completamente irrelevantes. En este caso, un análisis continuo de la información conforme los datos van llegando es más interesante. A este tipo de minería de datos se le conoce como minería de flujo de datos [5].

La minería de flujo de datos tiene en cuenta varios factores que hacen que la extracción de conocimiento en este tipo de datos sea un desafío en comparación con la minería de datos tradicional, como por ejemplo la actualización continua del modelo de aprendizaje o la necesidad de desechar información antigua [6], [5]. Además, muchos sensores y fuentes de datos poseen una tasa de refresco muy elevada (del orden de Khz) que implican además un aprendizaje lo más rápido posible [7].

La minería de patrones emergentes (EPM) [8], [9] es una tarea de minería de datos encuadrada dentro del marco de tareas denominado "descubrimiento de reglas descriptivas mediante aprendizaje supervisado" (SDRD) [10]. El principal objetivo de la tarea es la extracción de patrones descriptivos cuyo soporte varíe significativamente de un conjunto de datos (o clase) a otro. Esto quiere decir que EPM se encuentra a medio camino entre las inducciones descriptiva y predictiva ya que se pretende describir relaciones entre los datos utilizando para ello aprendizaje supervisado. Las principales finalidades de esta tarea son la descripción de las características discriminativas entre clases o la descripción de tendencias emergentes. No obstante, a pesar de las claras capacidades descriptivas de la tarea, esta se ha utilizado ampliamente en la literatura como un clasificador, aplicándose con éxito en campos como la Química [11], [12], Bioinformática [13], [14] o Medicina [15], [16], entre otros [17]. No obstante, un nuevo enfoque basado en en el uso de sistemas difuso evolutivos (EFSs) [18] ha sido desarrollado recientemente con propuestas capaces de extraer conocimiento con un buen balance entre la capacidad descriptiva de las reglas y su fiabilidad [19], [9], [20].

En este trabajo se presenta una primera propuesta para la extracción de patrones emergentes de calidad en entornos de minería de flujo de datos denominado SE2P (*Stream Extraction of Emerging Patterns*). Este algoritmo se basa principalmente en el empleo de dos fases: una fase *online* en la que se almacena la información hasta obtener un bloque de datos con un tamaño determinado y una fase *offline* basada en un sistema difuso evolutivo (EFS) multi-objetivo capaz de extraer conocimiento de las características discriminativas entre clases con un buen balance entre capacidad descriptiva y fiabilidad sobre dicho bloque de datos.

El trabajo se estructura de la siguiente manera: en la Sección II se introduce el problema y, en concreto, la definición y características de EPM y de la minería de flujo de datos a lo largo de la literatura. A continuación, en la Sección III se presenta el enfoque de extracción de conocimiento propuesto y en la Sección IV se muestra un estudio experimental para la validación de la calidad del conocimiento extraído. Por último, se presentan las conclusiones extraídas junto a los posibles trabajos futuros.

## II. Preliminares

En esta sección se revisan los conceptos básicos referentes a EPM y la minería de flujo de datos. En primer lugar se presenta la definición de EPM así como sus objetivos principales. A continuación, se define la minería de flujo de datos junto a una breve descripción de los diferentes enfoques utilizados. Por último, se presentan las principales medidas de calidad

empleadas en EPM y cómo son empleadas en SE2P para la minería de flujo de datos.

### II-A. *Minería de patrones emergentes*

La minería de patrones emergentes fue definida por Dong y Li [8], [9] como:

> "Sea un patrón $X$ cualquiera, y sea $\rho > 1$ un valor de umbral, $X$ se denominará como emergente si y solo si su índice de crecimiento entre dos conjuntos de datos $D_1$ y $D_2$ es mayor que $\rho$."

Este índice de crecimiento (GR) es definido con una función representada en la Ecuación 1.

$$GR(X) = \begin{cases} 0, & \text{Si } Sop_1(X) = Sop_2(X) = 0, \\ \infty, & \text{Si } Sop_1(X) = 0 \wedge Sop_2(X) \neq 0, \\ \frac{Sop_2(X)}{Sop_1(X)}, & \text{en otro caso} \end{cases} \quad (1)$$

donde $Sop_i(X)$ es el soporte del patrón $X$ en el conjunto de datos $i$.

Los propósitos principales para los que fue definida la tarea son:

- La descripción de las diferencias características entre clases o conjuntos de datos.
- La descripción de tendencias emergentes.
- La detección de diferencias entre múltiples variables.

En concreto, nuestra propuesta se centrará en el primer objetivo, es decir, lo que buscamos son diferencias características entre las clases de un flujo de datos a lo largo del tiempo. Habitualmente, estos patrones se presentan al experto en forma de reglas con el siguiente formato [21]:

$$R : Cond \rightarrow Clase \quad (2)$$

donde $Cond$ es un conjunto de características, normalmente en forma de pares atributo-valor y $Clase$ es el valor de la variable objetivo o de interés.

La principal dificultad que posee la extracción de estos patrones se encuentra en la misma definición de patrón emergente. El GR se define en función de un ratio entre soportes, lo cual propicia que el espacio de los patrones emergentes no sea convexo [22]. Esto quiere decir que patrones más específicos, y por tanto, con menor soporte, puedan poseer valores de GR más elevados que aquellos cuyos soportes sean más altos. Es por esta razón que se han definido a lo largo de la literatura diferentes tipos de patrones emergentes cuyas restricciones permiten una mayor facilidad de extracción, como los patrones *Jumping* o los patrones emergentes $\chi^2$, entre otros [23], [24], [25], [9]. Asimismo, se han desarrollado diferentes técnicas algorítmicas para la extracción de estos patrones encuadradas en cuatro categorías diferentes [9]. A pesar del amplio desarrollo de la tarea, la gran mayoría de algoritmos han sido desarrollados como clasificadores, ignorando las cualidades descriptivas de la tarea. Sin embargo, en los últimos años se ha desarrollado un enfoque basado en EFSs, donde destaca el algoritmo MOEA-EFEP [20], cuyos resultados poseen un buen balance entre las cualidades descriptivas y la fiabilidad de las reglas.

### II-B. *Minería de flujo de datos*

Un flujo de datos se define como una secuencia ordenada y potencialmente infinita de ejemplos que llegan al sistema a lo largo del tiempo a una velocidad que puede ser variable [26]. Esta definición tan simple de un flujo de datos trae consigo una gran variedad de diferencias respecto a la minería de datos tradicional, donde se destaca [5]:

- No se puede almacenar toda la información en memoria. Al ser potencialmente de tamaño infinito hay que buscar estrategias para procesar y descartar los datos.
- Todos los datos no se encuentran disponibles en el momento del aprendizaje. Los datos van llegando a lo largo del tiempo y es el modelo el que tiene que aprender conforme los datos lleguen, es decir, el modelo debe de adaptarse a lo largo del tiempo.
- Como el flujo es generado por una fuente a lo largo del tiempo, el fenómeno que subyace en la generación de dichos datos suele cambiar con mayor o menor frecuencia. A este hecho se le denomina en la literatura como cambio de concepto [27]. Esto quiere decir que el modelo aprendido con los datos en un instante $t$, si se produce un cambio de concepto, no será válido para los datos en el instante $t + x$.
- Esta velocidad de llegada es, normalmente, alta respecto a la capacidad de procesamiento que se posee. Por lo tanto, se busca que el algoritmo de aprendizaje sea capaz de aprender lo más rápido posible con el fin de evitar el encolamiento de las instancias.

Teniendo en cuenta estas características, las instancias que llegan al sistema se pueden procesar de dos formas diferentes [5]:

- Online. Las instancias llegan una a una y son procesadas por el algoritmo de aprendizaje tan pronto estén disponibles.
- Por bloques. Las instancias son almacenadas hasta obtener un bloque de datos de un tamaño predeterminado y donde todo el bloque es procesado a la vez por el algoritmo de aprendizaje.

Una vez se ha elegido la metodología de procesamiento de los datos, es necesario determinar la estrategia de aprendizaje del método. Entre las técnicas más utilizadas para el aprendizaje en minería de flujo de datos destacan [28]:

- Detectores de cambio de concepto [29]. Son métodos externos al algoritmo de aprendizaje que calculan diferentes propiedades del flujo de datos para detectar cambios. Normalmente poseen dos niveles: un nivel de alerta en donde el cambio empieza a ocurrir, donde solo se aprende usando los datos más recientes; y un nivel de alarma indicando un cambio severo donde el clasificador se reemplaza.
- Ventanas deslizantes [30]. Se almacenan en un *buffer* de memoria las instancias más recientes, deshaciéndose de aquellas más antiguas que no caben en dicho *buffer*. Esto permite al método aprender únicamente de las instancias más recientes en el flujo.

- Ensemble [31]. En donde se utilizan diferentes clasificadores que permiten hacer un seguimiento del flujo de datos. Existen actualmente dos estrategias principales: aprender un nuevo método y añadirlo al ensemble, descartando métodos antiguos si es necesario, o bien mediante un esquema de pesos en los diferentes clasificadores en función de su rendimiento.

### *II-C. Medidas de calidad*

En minería de datos tradicional, la calidad de un modelo se determina mediante diferentes mecanismos como por ejemplo la validación cruzada [32]. Sin embargo, muchos de estos mecanismos requieren del uso del conjunto de datos completo para poder realizar una evaluación correcta. Esto es imposible en minería de flujo de datos porque no poseemos el conjunto de datos al completo. Por lo que es necesario el uso de técnicas diferentes para la evaluación de los modelos.

Uno de los esquemas de evaluación más utilizado es el denominado *test-then-train* [33]. En este esquema, cuando una instancia o bloque de datos llega al sistema sirve para evaluar el modelo actual o hacer una predicción sobre la instancia concreta o el bloque de datos. Una vez se realiza este proceso, dicha instancia o bloque pasa a entrenar el modelo actual.

En EPM, la medida más importante es el GR, pues es la métrica que define la tarea. No obstante, es necesario determinar diferentes aspectos tales como generalidad, interés o fiabilidad [9], claves para la correcta extracción de reglas interpretables y precisas. La tarea fue concebida inicialmente para el análisis de problemas entre dos clases o conjuntos de datos. Sin embargo, la tarea puede ser fácilmente extendida a problemas multiclase mediante estrategias como *One vs All* (OVA) [34] donde la clase positiva es la que se encuentra descrita en el consecuente de la regla y la negativa el resto de clases del problema.

Cuadro I
MATRIZ DE CONFUSIÓN PARA UNA REGLA EN EPM.

| | Clase predicha | |
|---|---|---|
| Clase real | Positive | Negative |
| Positive | $p = tp$ | $\overline{p} = fn$ |
| Negative | $n = fp$ | $\overline{n} = tn$ |

En la Tabla I se puede observar la matriz de confusión, donde: $p$ representa el número de ejemplos correctamente cubiertos, $\overline{p}$ el número de ejemplos de la clase no cubiertos, $n$ el número de ejemplos cubiertos incorrectamente, y $\overline{n}$ el número de ejemplo correctamente no cubiertos.

Las medidas de calidad más utilizadas en EPM son las descritas a continuación [9]:

- Growth Rate (GR). Definida en la Ecuación 1, mide el poder discriminativo de una regla.
- Confianza (Conf). Se define como el ratio de la capacidad predictiva de la regla para la clase positiva [35].

$$Conf(R) = \frac{p}{p+n} \tag{3}$$

- Atipicidad (Atip). Esta medida híbrida muestra el balance existente entre generalidad y ganancia de precisión de la regla [21].

$$Atip(R) = \frac{p+n}{P+N}\left(\frac{p}{p+n} - \frac{P}{P+N}\right) \tag{4}$$

  El dominio de esta medida tiene una dependencia directa con el porcentaje de la clase a medir, por lo tanto, para realizar comparaciones es necesario normalizar esta medida. Esta normalización se ha llevado a cabo de la siguiente manera:

$$Atip\_Norm(R) = \frac{Atip(R) - (\frac{P}{T}(0 - \frac{P}{T}))}{(\frac{P}{T}(1 - \frac{P}{T})) - (\frac{P}{T}(0 - \frac{P}{T}))} \tag{5}$$

- Tasa de falsos positivos (FPR). Mide el porcentaje de ejemplos incorrectamente cubiertos respecto al total de ejemplos de la clase negativa. Esta medida debe ser minimizada para la obtención de reglas precisas [36].

$$FPR(R) = \frac{n}{N} \tag{6}$$

- Tasa de verdaderos positivos (TPR). Mide el porcentaje de ejemplos correctamente cubiertos respecto al número total de ejemplos de la clase positiva [37].

$$TPR(R) = \frac{p}{P} \tag{7}$$

- Número de reglas. Mide la cantidad de reglas extraídas.
- Número de variables. Mide el número medio de variables que se obtienen en el conjunto de reglas.

## III. SE2P: STREAM EXTRACTION OF EMERGING PATTERNS

Esta sección presenta la propuesta algorítmica para la extracción de patrones emergentes descriptivos para minería de flujo de datos llamado SE2P.

SE2P utiliza un enfoque basado en dos fases online/offline utilizando para ello el esquema de procesamiento de instancias basado en bloques. Esto implica que la fase online en un primer lugar agrupará instancias provenientes del stream hasta que se alcance el número de instancias determinado por el tamaño de bloque. Cuando hay un bloque de datos disponible, la fase offline ejecutará el núcleo de SE2P, el cual es un EFS multi-objetivo basado en el enfoque NSGA-II [38] para la extracción de un conjunto de reglas que representen las características discriminativas de las diferentes clases que se encuentren en dicho bloque. Es importante destacar que la estrategia de aprendizaje de SE2P se basa en una modificación del esquema de ventanas deslizantes donde se almacenan los modelos de reglas obtenidos previamente más una función de evaluación en el proceso evolutivo que permite el uso de esta estructura con el objetivo de adaptarse a los cambios de concepto.

Este algoritmo evolutivo utiliza una codificación "cromosoma = regla" donde se representa tanto el antecedente como el consecuente de la regla, lo que permite la extracción de reglas de todas las posibles clases en una única ejecución. La única representación disponible en este método es la forma

normal disyuntiva (DNF), ya que se ha demostrado que se obtienen mejores resultados que con otras representaciones [39]. La Figura 1 presenta una regla DNF, representada en el genotipo por un vector binario. Es importante destacar que para variables numéricas se emplea el número de conjuntos difusos correspondientes a etiquetas lingüísticas que se definen por funciones de pertenencia triangulares uniformes.

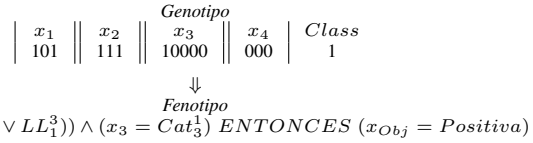


Figura 1. Representación de una regla DNF en EvAEP.

### III-A. Operador de inicialización

SE2P utiliza un operador de inicialización sesgada basado en el conocimiento previo con el fin de poder actualizarlo con los nuevos datos. Este operador añade el modelo de reglas extraído del bloque de datos anterior a la población inicial ($P_0$) mientras que el resto de individuos son inicializados de manera completamente aleatoria, pero controlando la inicialización de, como máximo, el 25 % de sus variables en el 75 % de individuos de esta población. Esto permite la obtención de individuos con gran generalidad al proceso evolutivo.

### III-B. Operadores genéticos

La población de la siguiente generación es obtenida mediante la aplicación de distintos operadores genéticos: operador de selección por torneo binario [40], un operador de cruce multipunto [41] y un operador de mutación sesgada empleado por primera vez en un algoritmo de descubrimiento de subgrupos en [42].

Asimismo, SE2P utiliza un operador de reinicialización con el fin de evitar el estancamiento de la población en un óptimo local. Esta reinicialización se lleva a cabo mediante la utilización de un operador de token competition [43] con el que se eliminan reglas solapadas. A continuación, el resto de individuos se genera aleatoriamente.

### III-C. Función de evaluación

Uno de los aspectos fundamentales en la minería de flujo de datos es la necesidad de adaptar el modelo en función de los datos que van llegando. Como hemos visto anteriormente, el fenómeno subyacente en los datos puede variar a lo largo del tiempo, pudiendo invalidar todo el conocimiento previo. Por lo tanto SE2P utiliza un esquema de ventana deslizante con pesos en donde se almacenan únicamente los $n$ últimos conjuntos de reglas extraídos. Con esta estructura, la función que se utilizará para determinar la calidad de cada objetivo en SE2P para una regla $R_i$ viene dada por la Ecuación 8.

$$Fitness_t^k(R_i) = QM_t^k(R_i) \cdot \left[1 - \sum_{j=t-n}^{t} SW(R_i, j) \cdot 2^{-(t-j)}\right] \quad (8)$$

donde $QM_t^k(R_i)$ es el valor de la medida de calidad usada como objetivo $k$ en el bloque de datos $t$ actual, $SW(R_i, j)$ es una función que devuelve cero si $R_i$ se encuentra en el conjunto de reglas devuelto para el bloque $j$ o uno en caso contrario. El objetivo de esta función de evaluación no es otro que la penilización de aquellas reglas que no definen el fenómeno subyacente en los datos, representado por las reglas extraídas anteriormente. Esto se debe a que el cambio normalmente se produce de manera gradual, por lo que el conocimiento previo sigue siendo relevante hasta que el nuevo prevalece.

### III-D. Esquema de funcionamiento

El proceso de ejecución de SE2P es el siguiente: una vez se ha obtenido un bloque de datos, se lanza la fase de aprendizaje offline. En un primer lugar, siguiendo el esquema *test-then-train* se evaluará el modelo de reglas extraído en el bloque anterior. Una vez evaluado el modelo de reglas, dicho bloque de datos pasa a ser un conjunto de entrenamiento. Es importante destacar que, al no existir un modelo de reglas previo, el primer bloque pasa a ser directamente de entrenamiento. Por lo tanto, las evaluaciones se realizan a partir del segundo bloque.

A continuación, el EFS multi-objetivo se ejecuta, comenzando con la aplicación del operador de inicialización sesgada basada en conocimiento previo para generar $P_0$. Después, el proceso evolutivo da comienzo, ejecutándose durante $g$ generaciones o hasta que un nuevo bloque de datos esté disponible. Dentro de este proceso evolutivo se aplicarán los operadores genéticos, obteniendo una población de descendientes $Q_g$ de igual tamaño que la población actual. A continuación ambas poblaciones se unen en $U_g$, se evalúan aquellos individuos no evaluados y se obtiene la población de la siguiente generación mediante la ordenación por frentes de dominancia propia del algoritmo NSGA-II. Por último, se analiza el estancamiento de la población, donde se comprueba si la población actual no ha cubierto ejemplos nuevos del bloque durante un 25 % del total de generaciones. El operador de reinicialización se aplicará en caso de que la población se estanque.

Una vez finalizado el proceso evolutivo, el algoritmo aplicará el operador de token competition [43] para eliminar aquellas reglas redundantes y el resultado de este procedimiento será el devuelto al usuario.

## IV. Estudio experimental

En este trabajo se presenta un estudio preliminar de una primera propuesta para la extracción de patrones emergentes con un buen balance entre la capacidad descriptiva de las reglas extraídas y la fiabilidad de las mismas dentro de la minería de flujo de datos. En concreto, se abordará el tratamiento de Big Data mediante técnicas de minería de flujos de datos. Para ello se ha simulado un flujo de datos mediante la herramienta de minería de datos MOA [44]. Las características de los conjuntos de datos utilizados como por ejemplo el número de instancias totales, número de variables y número de clases se presentan en la Tabla II.

Cuadro II
Conjuntos de datos utilizados en el estudio experimental.

| Nombre | # Instancias | # Variables | # Clases |
|---|---|---|---|
| Air | 539395 | 7 | 2 |
| Elec | 45312 | 7 | 2 |
| forest | 581012 | 54 | 7 |
| Higgs | 11000000 | 28 | 2 |
| Susy | 5000000 | 18 | 2 |

Los parámetros utilizados por SE2P en este estudio experimental se muestran en la Tabla III, los parámetros utilizados son similares a los utilizados por EFSs desarrollados para EPM sobre datos estáticos [20]. Sin embargo, uno de los parámetros más importantes para determinar el rendimiento de SE2P es el tamaño del bloque de datos que se procesará. En este trabajo se van a seleccionar tres tamaños de bloque diferentes: 1500, 2500 y 5000 instancias por bloque con el objetivo de determinar el más apropiado.

Cuadro III
Parámetros utilizados por SE2P en el estudio experimental.

| Parámetro | Valor |
|---|---|
| Etiquetas lingüísticas | 3 |
| Número de generaciones máximas | 60 |
| Tamaño de población | 50 |
| Probabilidad de cruce | 0.7 |
| Probabilidad de mutación | 0.1 |
| Tamaño de ventana temporal | 5 |

Cuadro IV
Resultados medios obtenidos por SE2P en los flujos de datos analizados.

| Nombre | Tam. bloque | $n_r$ | $n_v$ | ATIP | CONF | GR | TPR | FPR | Tiempo ejec. (ms) |
|---|---|---|---|---|---|---|---|---|---|
| Air | 1500 | 2,006 | **2,539** | **0,588** | **0,594** | **0,953** | **0,521** | 0,346 | **400,402** |
| | 2500 | **2,000** | 2,539 | 0,584 | 0,581 | 0,914 | 0,498 | 0,330 | 473,280 |
| | 5000 | **2,000** | 2,580 | 0,575 | 0,541 | 0,873 | 0,477 | **0,327** | 1159,472 |
| Elec | 1500 | 2,000 | 2,259 | 0,657 | 0,688 | 0,966 | 0,583 | 0,269 | **433,586** |
| | 2500 | **1,941** | **2,088** | 0,659 | **0,700** | **0,971** | **0,612** | **0,294** | 557,235 |
| | 5000 | 2,000 | 2,875 | **0,661** | 0,687 | 0,938 | 0,583 | 0,262 | 581,125 |
| forest | 1500 | **4,054** | **9,107** | **0,770** | **0,519** | **0,969** | **0,799** | **0,241** | **1692,702** |
| | 2500 | 4,134 | 9,578 | 0,739 | 0,495 | 0,948 | 0,764 | 0,263 | 1915,446 |
| | 5000 | 4,252 | 9,184 | 0,693 | 0,437 | 0,914 | 0,730 | 0,307 | 2616,087 |
| Higgs | 1500 | 2,011 | **7,640** | 0,524 | 0,549 | 0,913 | 0,475 | 0,426 | **263,109** |
| | 2500 | 2,013 | 8,237 | 0,526 | **0,550** | 0,960 | **0,477** | 0,426 | 337,272 |
| | 5000 | **2,007** | 9,904 | **0,526** | 0,536 | **0,992** | 0,433 | **0,380** | 436,454 |
| Susy | 1500 | 2,006 | **7,081** | **0,622** | 0,674 | **1,000** | **0,516** | **0,272** | **212,349** |
| | 2500 | **2,004** | 7,454 | 0,604 | **0,692** | **1,000** | 0,486 | 0,277 | 257,945 |
| | 5000 | 2,005 | 7,821 | 0,596 | 0,681 | **1,000** | 0,487 | 0,296 | 315,184 |

En la Tabla IV se presentan los resultados medios obtenidos por SE2P a lo largo de los diferentes flujos de datos analizados. Es importante remarcar que, debido al funcionamiento de SE2P, estos resultados medios se han obtenido usando $n_b - 1$ bloques, siendo $n_b$ el número total de bloques analizados. A continuación se muestra un análisis de cada una de las diferentes medidas de calidad analizadas:

- Número de reglas y variables. En general el número de reglas obtenido es muy bajo. Se destaca que el número de instancias por bloque no influye significativamente en estos resultados. Sin embargo, se puede observar que el número de variables es en algunos casos elevado, llevando el modelo extraído a una nivel de complejidad mayor. Sin embargo, en líneas generales, el modelo extraído es simple.
- Atipicidad. Esta medida, que mide el interés de las reglas extraídas, nos indica que en general las reglas obtenidas son interesantes, obteniéndose mejores resultados en bloques de 1500 instancias.
- Confianza. En esta medida se puede observar una amplia variabilidad en los resultados, donde el mejor resultado se obtiene con tamaños de bloque de 2500 en tres de los cinco conjuntos analizados. No obstante, el nivel de confianza es aceptable en los diferentes tamaños de bloque.
- GR. En esta medida se representa el porcentaje de patrones extraídos que son patrones emergentes. Como se puede observar, en general se extraen reglas emergentes y que por tanto poseen altas capacidades discriminativas, independientemente del tamaño de bloque escogido.
- Balance TPR-FPR. En estos resultados se puede observar que la generalidad de las reglas medidas como TPR es bastante elevada. Sin embargo, a pesar de que los niveles de FPR son elevados, la diferencia entre ambas métricas nos permite poder decir que los resultados obtenidos poseen, en general, un buen balance entre la generalidad y la precisión. También es importante destacar que los resultados son mejores a menor tamaño de bloque.
- Tiempo de ejecución. En cuento al tiempo de ejecución del algoritmo se puede observar que, obviamente, el tiempo medio de procesamiento de un bloque es menor en función del tamaño de bloque. Además, se destaca que el tiempo de ejecución es lo suficientemente rápido para el procesamiento de flujos de datos en donde se pueda procesar un bloque de datos cada segundo, lo cual es bastante aceptable.

## V. Conclusiones

En este trabajo se ha presentado un estudio preliminar de un primer enfoque la extracción de patrones emergentes para minería de flujo de datos. Este algoritmo se basa en el procesamiento de las instancias por bloques de datos de tamaño fijo en donde se ejecuta un algoritmo evolutivo multiobjetivo por cada bloque con el objetivo de extraer patrones emergentes que describan las características discriminativas de las diferentes clases del problema.

Los resultados del estudio realizado demuestran unos resultados muy prometedores y abren una línea de investigación donde se necesita implementar nuevas técnicas para mejorar los resultados actuales y los tiempos de ejecución a fin de poder procesas flujos de datos más veloces.

## Agradecimientos

Este trabajo ha sido subvencionado por el Ministerío de Economía y Competitividad bajo el proyecto TIN2015-68454-R y el contrato predoctoral FPI referencia BES-2016-077738 asociado al mismo (Fondos FEDER).

## Referencias


[1] A. Fernández, S. Río, V. López, A. Bawakid, M. del Jesus, J. Benítez, and F. Herrera, "Big Data with Cloud Computing: An Insight on the Computing Environment, MapReduce and Programming Frameworks," *WIREs Data Mining and Knowledge Discovery*, vol. 5, no. 4, pp. 380–409, 2014.

[2] T. Kraska, "Finding the Needle in the Big Data Systems Haystack," *IEEE Internet Computing*, vol. 17, no. 1, pp. 84–86, 2013.

[3] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2014.

[4] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," in *Big Data Analysis: New Algorithms for a New Society*. Springer, 2016, pp. 91–114.

[5] J. Gama, *Knowledge discovery from data streams*. CRC Press, 2010.

[6] A. Bifet, "Adaptive learning and mining for data streams and frequent patterns," Ph.D. dissertation, Universitat Politècnica de Catalunya, 2009.

[7] D. Han, C. Giraud-Carrier, and S. Li, "Efficient mining of high-speed uncertain data streams," *Applied Intelligence*, vol. 43, no. 4, pp. 773–785, 2015.

[8] G. Z. Dong and J. Y. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," in *Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press, 1999, pp. 43–52.

[9] A. M. García-Vico, C. J. Carmona, D. Martín, M. García-Borroto, and M. J. del Jesus, "An overview of emerging pattern mining in supervised descriptive rule discovery: Taxonomy, empirical study, trends and prospects," *WIREs: Data Mining and Knowledge Discovery*, vol. 8, no. 1, 2018.

[10] P. Kralj-Novak, N. Lavrac, and G. I. Webb, "Supervised Descriptive Rule Discovery: A Unifying Survey of Constrast Set, Emerging Pateern and Subgroup Mining," *Journal of Machine Learning Research*, vol. 10, pp. 377–403, 2009.

[11] R. Sherhod, V. J. Gillet, T. Hanser, P. N. Judson, and J. D. Vessey, "Toxicological knowledge discovery by mining emerging patterns from toxicity data," *Journal of Chemical Information and Modeling*, vol. 5, no. S-1, p. 9, 2013.

[12] A. Lepailleur, G. Poezevara, and R. Bureau, "Automated detection of structural alerts (chemical fragments) in (eco) toxicology," *Computational and structural biotechnology journal*, vol. 5, no. 6, pp. 1–8, 2013.

[13] M. Piao, H. G. Lee, G. Y. Sohn, G. Pok, and K. H. Ryu, "Emerging patterns based methodology for prediction of patients with myocardial ischemia," in *Proc. of the 6th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2009, pp. 174–178.

[14] G. Tzanis, I. Kavakiotis, and I. P. Vlahavas, "Polya-iep: A data mining method for the effective prediction of polyadenylation sites," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 398–12 408, 2011.

[15] P. W. Angriyasa, Z. Rustam, and W. Sadewo, "Non-invasive intracranial pressure classification using strong jumping emerging patterns," in *Proc. of the 2011 International Conference on Advanced Computer Science and Information System (ICACSIS)*. IEEE, 2011, pp. 377–380.

[16] Y. Yu, K. Yan, X. Zhu, and G. Wang, "Detecting of PIU Behaviors Based on Discovered Generators and Emerging Patterns from Computer-Mediated Interaction Events," in *Proc. of the 15th International Conference on Web-Age Information Management*, ser. LNCS, vol. 8485. Elsevier, 2014, pp. 277–293.

[17] G. Li, R. Law, H. Q. Vu, J. Rong, and X. R. Zhao, "Identifying emerging hotel preferences using emerging pattern mining technique," *Tourism management*, vol. 46, pp. 311–321, 2015.

[18] F. Herrera, "Genetic fuzzy systems: taxomony, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, pp. 27–46, 2008.

[19] A. M. García-Vico, J. Montes, J. Aguilera, C. J. Carmona, and M. J. del Jesus, "Analysing Concentrating Photovoltaics Technology through the use of Emerging Pattern Mining," in *Proc. of the 11th International Conference on Soft Computing Models in Industrial and Environmental Applications*. Springer, 2016, pp. 1–8.

[20] A. M. García-Vico, C. J. Carmona, P. González, and M. J. del Jesus, "Moea-efep: Multi-objective evolutionary algorithm for extracting fuzzy emerging patterns," *IEEE Transactions on Fuzzy Systems*, In Press.

[21] C. J. Carmona, M. J. del Jesus, and F. Herrera, "A Unifying Analysis for the Supervised Descriptive Rule Discovery via the Weighted Relative Accuracy," *Knowledge-Based Systems*, vol. 139, pp. 89–100, 2018.

[22] L. Wang, H. Zhao, G. Dong, and J. Li, "On the complexity of finding emerging patterns," in *Proc. of the 28th Annual International Computer Software and Applications Conference*, vol. 2, 2004, pp. 126–129.

[23] G. Z. Dong, J. Y. Li, and X. Zhang, "Discovering jumping emerging patterns and experiments on real datasets," in *Proc. on International Database Conference Heterogeneous and Internet Databases*, 1999, pp. 155–168.

[24] H. Fan and K. Ramamohanarao, "Noise Tolerant Classification by Chi Emerging Patterns," in *Proc. of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, ser. LNCS, vol. 3056. Springer, 2004, pp. 201–206.

[25] K. Ramamohanarao and H. Fan, "Patterns Based Classifiers," *World Wide Web*, vol. 10, no. 1, pp. 71–83, 2007.

[26] M. M. Gaber, "Advances in data stream mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 79–85, 2012.

[27] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, 2014.

[28] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Wozniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[29] R. M. Vallim and R. F. De Mello, "Proposal of a new stability concept to detect changes in unsupervised data streams," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7350–7360, 2014.

[30] J. Shan, J. Luo, G. Ni, Z. Wu, and W. Duan, "Cvs: fast cardinality estimation for large-scale data streams over sliding windows," *Neurocomputing*, vol. 194, pp. 107–116, 2016.

[31] B. Krawczyk, L. L. Minku, J. a. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[32] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[33] A. Wald, *Sequential analysis*. Courier Corporation, 1973.

[34] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761 – 1776, 2011.

[35] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery: an overview," in *Advances in knowledge discovery and data mining*. AAAI/MIT Press, 1996, pp. 1–34.

[36] D. Gamberger and N. Lavrac, "Expert-Guided Subgroup Discovery: Methodology and Application," *Journal Artificial Intelligence Research*, vol. 17, pp. 501–527, 2002.

[37] W. Kloesgen, "Explora: A Multipattern and Multistrategy Discovery Assistant," in *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, 1996, pp. 249–271.

[38] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[39] A. García-Vico, C. J. Carmona, and M. J. del Jesus, "Análisis de diferentes tipos de reglas en sistemas difusos evolutivos para minería de patrones emergentes," in *Proc. of the XII Spanish Conference on Metaheuristics, Evolutive and Bioinspired Algorithms (MAEB 2017)*, 2017, p. 876–885.

[40] B. L. Miller and D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," *Complex System*, vol. 9, pp. 193–212, 1995.

[41] J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, 1975.

[42] C. J. Carmona, P. González, M. J. del Jesus, and F. Herrera, "NMEEF-SD: Non-dominated Multi-objective Evolutionary algorithm for Extracting Fuzzy rules in Subgroup Discovery," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 958–970, 2010.

[43] K. S. Leung, Y. Leung, L. So, and K. F. Yam, "Rule Learning in Expert Systems Using Genetic Algorithm: 1, Concepts," in *Proc. of the 2nd International Conference on Fuzzy Logic and Neural Networks*, K. Jizuka, Ed., 1992, pp. 201–204.

[44] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010. [Online]. Available: https://moa.cms.waikato.ac.nz/