Algoritmo semisupervisado de agrupamiento que combina SUBCLU y el agrupamiento basado en restricciones, para la detección de grupos en conjuntos de alta dimensionalidad

Semisupervised clustering algorithm combining SUBCLU and constrained clustering for detecting groups in high dimensional datasets

Luis Alexander Calvo-Valverde[1], Alonso Vallejos-Peña[2]

Fecha de recepción: 7 de octubre de 2017
Fecha de aprobación: 3 de febrero de 2018

Calvo-Valverde, L; Vallejos-Peña, A. Algoritmo semisupervisado de agrupamiento que combina SUBCLU y el agrupamiento basado en restricciones, para la detección de grupos en conjuntos de alta dimensionalidad. *Tecnología en Marcha*. Vol. 31-3. Julio-Setiembre 2018. Pág 74-85.

DOI: 10.18845/tm.v31i3.3904

1 Doctor en Ciencias Naturales para el Desarrollo (DOCINADE), máster en Computación del Instituto Tecnológico de Costa Rica, miembro del Programa Multidisciplinar eScience. Costa Rica. Correo: lcalvo@itcr.ac.cr
2 Máster en Computación, Instituto Tecnológico de Costa Rica. Costa Rica. Correo: alonvalle17@gmail.com

Palabras clave

Minería de datos; subespacios; SUBCLU; algoritmo de agrupamiento; agrupamiento por restricciones.

Resumen

Los datos de alta dimensionalidad plantean un desafío para los algoritmos de agrupamiento tradicionales, ya que las medidas de similitud convencionales utilizadas por estos no son significativas cuando se aplican sobre el espacio completo de datos, por lo que afectan la calidad de los grupos. Ante esto, los algoritmos de agrupamiento de subespacios han sido propuestos como alternativa para encontrar todos los grupos en todos los espacios del conjunto de datos [1].

Al detectar grupos en espacios de menor dimensionalidad, cada grupo detectado puede pertenecer a diferentes subespacios del conjunto de datos original [2]. Consecuentemente, atributos que el usuario considere de interés pueden ser excluidos en algunos o todos los grupos, perdiendo información importante y reduciendo el valor del resultado para los analistas.

En este proyecto, se propone un nuevo método que combina el algoritmo SUBCLU [3] y el algoritmo de agrupamiento por restricciones [4], el cual permite al usuario identificar variables como atributos de interés con base en conocimiento previo del dominio, esto con el objeto de dirigir la detección de grupos hacia espacios que incluyan estos atributos y, por ende, generar grupos más significativos.

Keywords

Data mining; subspaces; SUBCLU; clustering; clustering by constraint.

Abstract

High dimensional data poses a challenge to traditional clustering algorithms, where the similarity measures are not meaningful, affecting the quality of the groups. As a result, subspace clustering algorithms have been proposed as an alternative, aiming to find all groups in all spaces of the dataset [1].

By detecting groups on lower dimensional spaces, each group may belong to different subspaces of the original dataset [2]. Therefore, attributes the user considers of interest may be excluded in some or all groups, decreasing the value of the result for the data analysts.

In this project, a new algorithm is proposed, that combines SUBCLU [3] and the clustering algorithms by constraint [4], which allows the users to identify variables as attributes of interest based on prior knowledge of domain, targeting direct group detection toward spaces that include user's attributes of interest, and thereafter, generating more meaningful groups.

Introducción

El análisis de agrupamientos divide los datos en grupos que son significativos, con base únicamente en los datos que describen los objetos, es decir, los diferentes atributos del conjunto de datos. La finalidad de este tipo de análisis es que los objetos dentro de los grupos generados sean lo más similares posible entre sí, y lo más distintos posible de los objetos de otros grupos [15].

Comúnmente, en datos de alta dimensionalidad, muchos atributos son irrelevantes y pueden ocultar grupos existentes entre datos ruidosos, dificultando la capacidad de los algoritmos de agrupamiento de encontrar grupos relevantes [6].

Técnicas de reducción de dimensionalidad como Principal Component Analysis (PCA), filtros de alta correlación y filtros de baja varianza son utilizados comúnmente para eliminar atributos irrelevantes en los datos [7]. Aunque útiles para otras tareas de minería de datos [8], las técnicas de reducción no son muy prácticas para buscar grupos entre datos de alta dimensionalidad, pues generan un subespacio único de los datos, y los grupos pueden ocultarse en distintos subespacios [2]. Ante esta deficiencia, surgen los algoritmos de agrupamiento de subespacios.

Los algoritmos de agrupamiento de subespacios son capaces de detectar grupos en diferentes espacios de menor dimensionalidad que el conjunto de datos original [9]. Al encontrar grupos en distintos subespacios, cada grupo detectado puede poseer diversos atributos del espacio original. Consecuentemente, atributos de interés pueden ser excluidos en algunos o todos los grupos identificados.

La exclusión de atributos de interés para el usuario, puede ocasionar un decremento en el valor de los grupos. La capacidad de utilizar conocimiento previo del dominio en la generación de los subespacios permitiría guiar la detección hacia aquellos que incluyan los atributos de valor. Sin embargo, a pesar de la existencia de múltiples algoritmos para otorgar distintos pesos a los atributos [10][11][12], ninguno permite guiar la detección de agrupamientos en espacios que incluyan atributos de interés previamente identificados por el usuario [13].

Entre los trabajos más relevantes para este proyecto, en [3] se propone SUBCLU, algoritmo que adopta la definición de grupos conectados por densidad, propuesto en DBSCAN. La ventaja principal de este método es la detección de agrupamientos de distintas formas y tamaños (grupos de forma irregular).

En el área de agrupamiento semisupervisado, Kailing *et al* [3] proponen el algoritmo de agrupamiento por restricciones, que permite a los usuarios especificar qué instancias pueden o no pueden agruparse entre sí. En este trabajo, logran extender con éxito el algoritmo COBWEB, y subsecuentemente, en [4] el K-means. Estos algoritmos, sin embargo, no son aplicables a datos de alta dimensionalidad.

Este proyecto se enfoca en mejorar los resultados en datos de alta dimensionalidad, al dar prioridad a grupos que incluyan atributos de interés para el usuario. Para esto se combinan el algoritmo de agrupamiento de subespacios SUBCLU [3] y el algoritmo de agrupamiento por restricciones.

Metodología

Para este proyecto se diseñó una serie de experimentos que permitieron determinar el impacto de la inclusión de algoritmos de restricciones en el algoritmo SUBCLU. A continuación se indican los detalles que considerar en el diseño de experimentos.

Hipótesis

Mediante la combinación del algoritmo de agrupamiento por restricciones y el algoritmo de agrupamiento de subespacios SUBCLU es posible guiar la detección de agrupamientos hacia espacios que incluyan atributos de interés definidos por el usuario.

Variables de respuesta

Las variables consideradas pertinentes para evaluar los experimentos corresponden a las métricas detalladas a continuación:

Variables internas

1. Cohesión de los grupos: Mide qué tan compacto es un grupo, o qué tan cercanos son todos los objetos dentro de un grupo [5].

$$WSS = \sum_{i} \sum_{x \varepsilon Ci} ||x - m_i||^2$$

2. Separación de los grupos: Mide qué tan separados se encuentran los grupos entre sí [5].

$$BSS = \sum_{i} |C_i||m - m_i||^2$$

3. Índice de silueta: Mide qué tan similar es un objeto a su propio grupo y comparado a los demás grupos. Los valores varían de -1 a 1, donde un valor alto indica que el objeto está bien ubicado en su propio grupo. Si la mayoría de los objetos tienen un valor alto, entonces el resultado se puede considerar inadecuado [14]. Esta métrica es un cálculo que evalúa los agrupamientos con base en las medidas de cohesión y separación.

$$S(i) = \frac{b\,(i) - a(i)}{\max\{a(i), b(i)\}}$$

Variables externas

Para estudiar los grupos detectados por el algoritmo extendido se realizó un estudio comparativo de los grupos generados por el algoritmo SUBCLU original y el algoritmo propuesto en este trabajo, los resultados se representaron de forma tabular. Entre las métricas que analizar en la evaluación integral se tomaron en cuenta la cantidad de subespacios encontrados, la cantidad de agrupamientos y la cantidad de agrupamientos con el atributo de interés.

Conjunto de datos

El conjunto de datos seleccionado para los experimentos fue USCENSUS1990, un conjunto de datos del censo de Estados Unidos de 1990, almacenado y compartido por la Universidad de California (UCI). El conjunto de datos es público, posee un total de 3 100 000 objetos y 68 atributos categóricos con gran variedad de rangos de posibles valores. Cabe destacar que el conjunto de datos ya había sido limpiado, discretizado, y que fue una muestra aleatoria del total del censo.

Para los experimentos, se utilizaron muestras de 10 000 objetos y 10 variables (las mismas variables en todas las muestras). Los atributos fueron seleccionados de acuerdo a su relevancia, la cantidad de los valores que podían tomar y la distribución de los datos. Para el análisis de datos se utilizó la herramienta R-studio, herramienta para el lenguaje R, que permite el análisis estadístico y gráfico de los datos.

Propuesta

Este proyecto surgió con el propósito de crear un método que permitiera realizar análisis de agrupamiento en conjuntos de alta dimensionalidad, y fuera capaz de detectar grupos en subespacios, con atributos de interés.

Para esto, los investigadores propusieron extender la funcionalidad del algoritmo de agrupamiento SUBCLU [3], mediante la inclusión de teoría de agrupamiento por restricciones [4]. Se escogió modificar este algoritmo, ya que es un método basado en densidades, lo que permite la detección de grupos de formas irregulares (no elípticas); además, es un algoritmo popular, con múltiples implementaciones de código abierto.

Esta solución permitió guiar la poda de subespacios hacia aquellos que tuvieran atributos de una lista previamente dada por el usuario como parámetro de ejecución del algoritmo, esto con la finalidad de mejorar la interpretabilidad de los resultados al reducir la cantidad de grupos mediante ese filtro. Se encontró que la calidad de los grupos se mantuvo estadísticamente igual en términos de índice de cohesión, separación y silueta.

Para este proyecto, se modificó el método de generación de subespacios del algoritmo SUBCLU, así: en el momento de la generación de los subespacios, el algoritmo detecta si los espacios propuestos poseen el atributo de interés; de no poseerlo el subespacio se corta y el algoritmo DBSCAN no se ejecuta para este ni para cualquier subespacio generado a partir de este.

La lógica del algoritmo de agrupación por restricciones se incluyó en el método GenerateCandidateSubspaces(S_k), el cambio propuesto en este trabajo se encuentra resaltado.

```
GenerateCandidateSubspaces(S_k)
  CandS_{k+1} := ∅
  for each s_1 ∈ S_k
    for each s_2 ∈ S_k
      if (s_1 and s_2 differ in exactely one attribute)
        CandS_{k+1} := CandS_{k+1} ∪ {s_1 ∪ s_2}
      end if
    end for
  end for
  // Pruning of irrelevant candidate subspaces
  for each cand ∈ CandS_{k+1}
    for each k-element s ⊂ cand
      if (s ∉ S_k and validSubspace( s, attributeOfInterest))
        CandS_{k+1} = CandSa_{k+1} \ {cand}
      end if
    end for
  end for
end
```

Figura 1. Pseudocódigo del algoritmo SUBCLU-R

La figura 1 muestra como la lógica del algoritmo de agrupamiento por restricciones se implementa en el paso de generación de subespacios, donde toma la forma de filtro, podando los subespacios irrelevantes. Este paso se ejecuta k veces (donde k = número de dimensiones), primero para generar subespacios de 1 dimensión; luego, de 2 dimensiones, y así sucesivamente.

El desempeño del algoritmo propuesto se comparó con el del algoritmo SUBCLU, utilizando las mediciones definidas en la sección 2.2.

Resultados

A continuación, se presentan los resultados obtenidos en las diferentes etapas del experimento.

El tamaño de la muestra utilizada para la evaluación de los algoritmos fue de 10 000 objetos y 10 atributos. También se evaluaron tamaños de muestras mayores y conjuntos con mayor cantidad de dimensiones; sin embargo, ELKI no aprovecha el paralelismo durante su ejecución, por lo que se descartaron pruebas de mayor tamaño de muestra o dimensiones, ya que los tiempos proyectados de ejecución aumentaban a más de 3 semanas por experimento.

El *framework* ELKI genera un archivo de extensión “txt” por cada agrupamiento detectado, con detalle de los objetos pertenecientes a cada grupo. En total, 1729 archivos fueron generados como resultado de la ejecución de ambos algoritmos.

El conjunto de archivos generados por ambos algoritmos ocupaba un espacio en disco superior a los 900 mb, por lo que se decidió implementar scripts en PySpark (Python para Spark) sobre los archivos en un sistema HDFS de Hadoop para aprovechar el procesamiento en paralelo de estos.

En general, la evaluación y comparación de resultados de agrupamiento es una tarea compleja, pues no se tiene una verdad base o se conocen los resultados de antemano. Por otro lado, existe gran cantidad de mediciones de la calidad interna que reflejan diferentes aspectos de los agrupamientos; sin embargo, el uso de estas no está estandarizado ni es universal, y se utilizan en cada investigación o propuesta de algoritmo a conveniencia [15]. Es por esto que se decidió evaluar con métricas externas que reflejaran lo que se consideró importante de los agrupamientos detectados, y con el conjunto de métricas internas que son ampliamente utilizadas y están referenciadas en la literatura [9].

Métricas externas

En esta sección se presenta un análisis de calidad externa de los grupos generados por los algoritmos SUBCLU y SUBCLU-R. En el cuadro 1 se muestran los hallazgos más importantes de los grupos generados por ambos algoritmos.

Cabe destacar que las métricas externas se mantuvieron idénticas en todos los experimentos de los algoritmos; es decir, los 6 experimentos de SUBCLU dieron los mismos resultados en cantidad de grupos generados, grupos con atributos de interés y el resto de métricas incluidas en el cuadro 1.

El cuadro 1 describe características de los agrupamientos de ambos algoritmos en términos de relevancia e interpretabilidad para el usuario. Por ejemplo, es más fácil analizar 10 grupos, de los cuales todos incluyen el atributo de interés, que analizar 50 grupos de los cuales no se sabe cuántos incluyan el mismo atributo.

Los resultados de los experimentos generaron un total de 1130 grupos para el algoritmo SUBCLU y 599 para SUBCLU-R, de los cuales 595 y 589 grupos, respectivamente, incluían el

atributo de interés. Esto significa que de los grupos que SUBCLU detectó, un 52,65% incluían la dimensión deseada, mientras que de los que SUBCLU-R detectó, un 98,33%, determinación que fue posible gracias a la implementación de reglas fuertes en la poda de los subespacios. La cantidad de grupos detectados por el algoritmo SUBCLU-R es mucho menor al algoritmo original; sin embargo, el porcentaje de 98,33% demuestra que es capaz de excluir eficazmente aquellos grupos que no sean de interés para el usuario.

Cuadro 1. Comparación entre métricas externas de los experimentos

	SUBCLU	SUBCLU-R
Total de grupos generados	1130	599
Grupos que incluyen el atributo de interés	595	590
Subespacios únicos	1023	521
Subespacios únicos que incluyen el atributo de interés	413	513
Grupos que no incluyen el atributo de interés	535	9
Grupos en común	549	549
Grupos en común con el atributo de interés	539	539
Grupos en común sin el atributo de interés	9	9
Grupos detectados por un algoritmo y no detectados por el otro	581	50
Grupos que incluyen el atributo de interés detectado por un algoritmo y no detectado por el otro	56	51

En las pruebas iniciales se detectó que si la poda de subespacios iniciaba a partir del espacio de 2 dimensiones, el resultado era mucho mejor en cuanto a la cantidad de subespacios detectados e igualaba los generados por el algoritmo original (siempre y cuando el atributo de interés se encontrara al inicio del conjunto de datos). Esto explica el 1,77% de grupos del algoritmo propuesto que no incluyen el atributo de interés.

De los 1130 grupos que SUBCLU detectó, 1023 eran subespacios únicos, mientras que SUBCLU-R encontró 521 de 599; esto representa un 90,53% y un 88,45%, respectivamente. Ante estos valores tan altos, se realizó un análisis de tamaño de los grupos; se encontró que para ambos algoritmos el tamaño medio de los grupos era de 7470 objetos. Tanto el tamaño de los grupos como el porcentaje tan alto de subespacios únicos son fuertes indicativos de que la densidad dada por los parámetros combinados de *eps* y minPts fue muy baja. En DBSCAN, un valor muy alto para este parámetro genera grupos excesivamente grandes (con la mayoría de los objetos); por transitividad, en SUBCLU esto se traduce a pocos grupos por cada subespacio.

En total, ambos algoritmos coincidieron en 539 grupos; es decir, un 89,98% de los grupos detectados por SUBCLU-R son grupos que SUBCLU encontró; esto es un buen indicador de que los cambios realizados al algoritmo no afectaron el funcionamiento del algoritmo original, y a su vez un indicador de la eficacia del algoritmo para la realización de la poda de subespacios.

Ambos algoritmos encontraron grupos con el atributo de interés que el otro no detectó, SUBCLU encontró 56 grupos nuevos, mientras que el SUBCLU restringido por reglas detectó 51

agrupamientos. El análisis de los grupos demostró que los 56 grupos detectados por SUBCLU eran variaciones de los 51 agrupamientos detectados por SUBCLU-R; es decir, contenían los mismos atributos, pero los grupos fueron generados por caminos distintos. Por ejemplo, el grupo X de SUBCLU fue conformado por las columnas *a, b, c,…, F*, mientras que el grupo Y del subespacio del algoritmo propuesto era *c, a, b,…., f,* donde *c* era el atributo de interés. La razón de esta diferencia entre los subespacios es la sensibilidad de SUBCLU-R a la posición de la columna dentro del conjunto de datos. Debido a la construcción de los subespacios de abajo hacia arriba, SUBCLU generara múltiples combinaciones del mismo subespacio, mientras que SUBCLU-R solo tomará en cuenta aquellos que tienen el atributo en la base del subespacio (primera o segunda posición en el conjunto de datos).

El efecto de los grupos incluidos o excluidos, en la calidad de los agrupamientos, puede ser entendido de forma más integral al observar las métricas de calidad internas; estas se presentan en la sección 4.3.

Métricas internas

Como se puede observar en el cuadro 2, los resultados son muy similares para ambos algoritmos en las 3 mediciones propuestas; sin embargo, sí se observó una gran caída en tiempo de ejecución del algoritmo SUBCLU-R con respecto al algoritmo original.

Cuadro 2. Promedio de mediciones internas de los experimentos

Parámetro	Función de distancia	SUBCLU	SUBCLU R
Cohesión	Euclideana	4,113	3,8832
Cohesión	Manhattan	2,9312	2,8813
Subtotal promedio		3,5221	3,38225
Separación	Euclideana	17,7541	17,721
Separación	Manhattan	17,5425	17,5101
Subtotal promedio		17,6483	17,6155
Silueta	Euclideana	1,0065	1,0061
Silueta	Manhattan	1,0055	1,0058
Subtotal promedio		1,006	1,006
Tiempo de ejecución (horas)	Euclideana	31:05	36:56
Tiempo de ejecución (horas)	Manhattan	29:45	36:11
Subtotal promedio		30:41	36:34

Se observaron mejoras leves de los valores de cohesión para SUBCLU-R. El excluir los grupos que no contenían la dimensión de interés mejoró la calidad interna de los grupos resultantes. Dado el alto porcentaje de coincidencias de grupos entre ambos algoritmos, presentado en la sección 4.2, se puede inferir que la calidad de los grupos mejoró no por haber encontrado grupos nuevos de mayor calidad, sino porque se excluyeron grupos de menor calidad debido

a no tener el atributo de interés. Esta hipótesis se refuerza al analizar el atributo de interés seleccionado para la ejecución del experimento de SUBCLU-R (*edad*), pues es un parámetro que en términos de análisis de población tiene mucha inferencia en los agrupamientos, por lo que es muy probable que los grupos generados que no incluían el atributo *edad* tuvieran menor cohesión.

La calidad de los grupos en términos de separación fue levemente inferior para el algoritmo propuesto, aunque la diferencia fue solamente de 0,18%. El análisis de los grupos muestra que esta diferencia en separación se debe a que todos los grupos tienen un mayor porcentaje de atributos en común (la mayoría ahora tiene como mínimo el atributo de interés), en este caso todos los grupos tienen al menos un atributo en común (*edad*), por lo que la separación entre los grupos tenderá a ser menor, pues los subespacios se encontrarán menos dispersos.

Al observar los resultados de cohesión y separación del algoritmo expuesto en este trabajo, y analizar el efecto de la restricción en los agrupamientos, se especula que al comparar SUBCLU y SUBCLU-R la cohesión mejore y la separación empeore. El grado de cambio de un resultado al otro dependerá de distintas variables, como la cantidad de atributos del conjunto de datos, la cantidad de atributos de interés, la relevancia del atributo en los grupos naturales de los datos o la cantidad de veces en las que el atributo de interés aparezca en los distintos agrupamientos. En cuanto al índice silueta, no se puede predecir cómo cambiará la métrica entre ambos algoritmos, pues el índice no se encuentra atado a los valores promedio de cohesión o separación de los grupos, sino a los valores individuales de cada grupo.

Hay una caída significativa de desempeño en el algoritmo SUBCLU-R. Aunque la cantidad de subespacios y grupos que analizar en cada iteración es menor que los generados por el algoritmo original, el tiempo de pre-evaluación de los grupos para comprobar la existencia del atributo de interés termina representando un aumento significativo de tiempo de ejecución.

Conclusiones

En este trabajo de investigación, se estudiaron los algoritmos de agrupamiento SUBCLU y el algoritmo de agrupamiento por restricciones, y la posibilidad de combinarlos para guiar la generación de espacios hacia aquellos que contuvieran los atributos de interés, y consecuentemente, facilitar la detección de agrupamientos de mayor valor para el usuario. Con este fin se realizó un análisis de distribución de datos utilizando R Studio, para la selección de los atributos a utilizar para el análisis; se modificó la implementación del algoritmo SUBCLU del *framework* ELKI con la lógica del algoritmo de agrupamiento por restricciones, utilizada por el algoritmo SUBCLU durante la evaluación de los subespacios generados; se ejecutó un experimento con el algoritmo SUBCLU original y el algoritmo SUBCLU modificado utilizando el conjunto de datos descrito en la sección 4.1; se escribieron *scripts* en Spark para analizar los archivos-texto generados por ambos experimentos, para obtener las mediciones internas y externas de los agrupamientos; se compararon los resultados de las mediciones de las variables de calidad internas y externas de los agrupamientos obtenidos por ambos algoritmos, y se hizo un análisis estadístico de los resultados que permitió aceptar la hipótesis.

En otras palabras, fue posible guiar la detección de subespacios hacia aquellos que incluyen atributos de interés para el usuario, especificado como parámetro de ejecución.

Pese a que el algoritmo original ofreció un mejor desempeño, el algoritmo modificado mantuvo valores muy similares de calidad interna, e incluso para algunos atributos representó una leve mejora. Esto demuestra que, aunque el algoritmo propuesto excluye gran cantidad de grupos e incluso subespacios completos, la calidad general de los agrupamientos se mantiene.

A partir de los resultados y su posterior análisis, se obtuvieron las siguientes conclusiones:

- Es posible extender el algoritmo SUBCLU mediante el algoritmo de agrupamiento por restricciones para guiar la poda de subespacios.
- La cantidad de agrupamientos detectados con atributo de interés por el algoritmo propuesto en este trabajo corresponde a $N - d$, donde N es el número de agrupamientos y d la cantidad de atributos del conjunto de datos.
- La modificación propuesta se basa en restricciones fuertes; es decir, la poda tomará en cuenta únicamente aquellos subespacios que contengan el atributo o los atributos de interés. La única excepción a la poda son los subespacios de una dimensión; estos no se podan, pues se reducirían drásticamente las posibles combinaciones de subespacios generados.
- La posición del atributo dentro del conjunto de datos (número de columna) influye en la generación de los subespacios dada la naturaleza *de abajo hacia arriba* del algoritmo SUBCLU. La cantidad de subespacios generados es mayor si el atributo de interés (columna) se coloca al inicio del conjunto de datos. Esta observación se realizó desde etapas tempranas de los experimentos; durante la ejecución se notó que la cantidad de los subespacios generados era mucho menor cuando la columna no se encontraba en la primera posición del conjunto de datos. Esto se debe a que el algoritmo SUBCLU genera los subespacios de abajo hacia arriba (subespacios de 1 dimensión hacia subespacios de más dimensiones), por lo que, si el subespacio base no tiene el atributo de interés, la cantidad de combinaciones generadas decrece notablemente.
- La cantidad de agrupamientos realizados por el algoritmo modificado es mucho menor que la realizada por el algoritmo original. Esto se debe a que a pesar de que el algoritmo poda más subespacios que el algoritmo SUBCLU original, el criterio de calidad interno del algoritmo SUBCLU para la detección de los grupos en los subespacios se mantiene.
- El tiempo de ejecución aumentó exponencialmente por cada atributo adicional debido al pobre desempeño del *framework* ELKI y el pobre uso de paralelismo, razón por la que se concluye que la actual implementación en ELKI no es apta para pruebas de muchas dimensiones, a menos que se cuente con un sistema que pueda correr de forma ininterrumpida por varios días.

Trabajo futuro

Debido a los problemas de desempeño con conjuntos grandes de datos multidimensionales , se sugiere implementar el algoritmo en un sistema con mayores capacidades de aprovechar el paralelismo para realizar pruebas con conjuntos de datos de mayor tamaño y dimensionalidad. Se recomienda el uso de Spark Apache, *framework* de código libre de computación distribuida, para manejo y análisis de conjuntos de datos grandes (*big data*). Este *framework* fue utilizado en este proyecto para el análisis de resultados de los grupos, y demostró gran capacidad de escalabilidad al permitir crear *clusters* con computadoras de escritorio e, incluso, máquinas virtuales.

En este trabajo se concluyó que es posible añadir restricciones de subespacios al algoritmo SUBCLU. La cantidad de agrupamientos generados no se puede anticipar, sin embargo, sí se conoce de antemano que la cantidad de grupos con el atributo de interés es $N - d$. La generación de los subespacios y agrupamientos que se realizan por el algoritmo propuesto es por restricciones fuertes (a excepción de los d subespacios de una dimensión), y aunque las mediciones de las variables internas y externas demuestran que la calidad general de los

agrupamientos se mantiene, podrían estarse excluyendo agrupamientos de gran calidad con cohesión superior. Se propone modificar el algoritmo de restricciones fuertes a restricciones basadas en pesos, donde, por medio un nuevo parámetro se le pueda otorgar un peso al atributo de interés para guiar la poda de los subespacios. Esto otorgaría mayor flexibilidad al algoritmo actual, pues permitiría la priorización de atributos sin que se excluyan grupos de gran calidad que no incluyan el atributo.

Finalmente, este trabajo se centró en la capacidad de crear agrupamientos en razón de su mayor interés para el usuario, aunque se concentró en SUBCLU, dados los resultados, intuitivamente se puede predecir que aplicar la misma lógica de restricciones para la generación, selección o evaluación de subespacios podría producir resultados similares, por lo que un trabajo de investigación similar podría llevarse a cabo para otros algoritmos de agrupamiento de subespacios.

Agradecimientos

El autor Calvo-Valverde agradece al DOCINADE y al Instituto Tecnológico de Costa Rica, pues fue en el marco de su investigación doctoral donde se generó el tema de la investigación aquí reseñada. El autor Vallejos Peña agradece a la Maestría en Computación del Instituto Tecnológico de Costa Rica y al profesor Calvo-Valverde por la excelente formación profesional recibida.

Referencias

[1] G. Chen, X. Ma, D. Yang, S. Tang and M. Shuai, “Mining representative subspace clusters in high-dimensional data,” 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, 2009, pp. 490-494. doi: 10.1109/FSKD.2009.463

[2] H. P. Kriegel, P. Kroger, and A. Zimek, “*Clustering high dimensional data: A survery on subspace clustering, pattern-based clustering, and correlation clustering*”, ACM Transactions Knowledge Discovery Data, New York, 2009, pp. 1:1--1:58. doi: 10.1145/1497577.1497578

[3] P. Kröger, H.P. Kriegel and K. Kailing. “*Density-connected subspace clustering for high-dimensional data*”. Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004, doi: 10.1137/1.9781611972740.23..

[4] K. Wagstaff and C. Cardie, “Clustering with instance-level constraints”, Proceedings of the Seventeenth International Conference on Machine Learning San Francisco, Morgan Kaufmann Publishers Inc. 2000, pp. 1103—1110.

[5] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Boston: Pearson Addison Wesley2005.

[6] L. Parsons, E. Haque, and H. Liu, “*Subspace clustering for high dimensional data: A review*” vol. 6, no. 1, pp. 90-105 ACM Sigkdd Explorations Newsletter. 2004.

[7] Witten, I. H., Eibe, F., Hall., M. A., Data Mining, 3rd Edition, Morgan Kaufmann Publishers, United States, 2011.

[8] L. v. d. Maaten and E. Postma, “Dimensionality reduction: A comparative reduction”, Tilburg Centre for Creative Computing, Tilburg, 2009.

[9] L. Chen, Q. Jiang and S. Wang, “*Cluster validation for subspace clustering on high dimensional data*,” APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, Macao, 2008, pp. 225-228.

doi: 10.1109/APCCAS.2008.4746001.

[10] W. DeSarbo, J. Carroll, L. Clark, and P. Green, “*Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables*”, Psychometrika 49: 57. https://doi.org/10.1007/BF02294206, 1984.

[11] Z. Huang, M. N. H. Rong, and Z. Li, “*Automated variable weighting in k-means type clustering*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 657-668, May 2005.

doi: 10.1109/TPAMI.2005.95, 2005.

[12] K. Kailing, H.-P. Kriegel, P. Kröger, and S. Wanka, "*Ranking interesting subspaces for clustering high dimensional data*," in PKDD, 2003, pp. 241-252..

[13] K. Sim, V. Gopalkrishnan, A. Zimek et al. "*Data Mining Knowledge Discovery*", 2013, 26: 332. https://doi.org/10.1007/s10618-012-0258-x.

[14] P. J. Rousseeuw,"*Silhouettes: A graphical aid to the interpretation and validation of cluster analysis*", Journal of Computational and Applied Mathematics. Volume 20, November 1987, Pages 53-65., 1987.

[15] E. Muller, S. Gunneman, I. Assent, and T. Seidl, "*Evaluating clustering in subspace projections of high dimensional data*", Proc. VLDB Endow. 2, 1, 1270-1281. August 2009, DOI=http://dx.doi.org/10.14778/1687627.1687770