PROPOSED RESEARCH:

The maximum length of this file is **10 pages** (letter size, Verdana size 10 is suggested)**.** For an adequate evaluation of your proposal merits,this file must include the following aspects: Proposal description, Hypothesis,Goals, Methodology, Work Plan, Work in progress and Available Resources. Make sure to describe the relevance of the proposal topic in relation to the state of the art in the field. Keep in mind the Bases del Concurso FONDECYT de Postdoctorado 2017 and Application Instructions.

1 Introducción

Como consecuencia del crecimiento explosivo de la WEB, la integración de motores de búsqueda como Google en computadoras personales y dispositivos móviles, y del uso de la redes sociales, la tarea de agrupamiento automático de texto, e.g. Tweets o documentos en un motor de búsqueda, resulta cada vez de mayor importancia debido a la necesidad de revelar las categorías subyacentes en grandes volúmenes de datos. La generación de grandes cantidades de documentos sobrepasa hoy en día las capacidades de computadores personales e incluso de equipos de cómputo de alto desempeño. A modo de ejemplo, se estima que la cantidad de páginas WEB que motores de búsqueda populares como Google o Yahoo indexan es del orden de las decenas de billón. Es por lo tanto de gran interés poder desarrollar técnicas capaces de automáticamente organizar, clasificar y resumir colecciones de documentos que se encuentren distribuidas en múltiples máquinas de una red, y que a su vez hagan uso eficiente del hardware moderno, en particular del paralelismo existente en las arquitectura multi-núcleo (multi-core). Problemas reales como *Collection Selection* para colecciones distribuídas, en donde para una consulta dada se debe identificar la colección más adecuada dentro de la cual se buscaran documentos relevantes Crestani and Markov (2013), los desafíos relacionados con la escalabilidad y eficiencia en los métodos de *Knowledge Discovery* se han vuelto de gran importancia. Los algoritmos tradicionales operan usualmente con conjuntos de datos cargados completamente en memoria principal. De ahí que puedan realizar operaciones de cálculo de distancias entre pares de documentos de manera muy rápida, visitando cada dato muchas veces. Cuando el tamaño de la colección es mucho mayor a la cantidad de memoria RAM disponible, ya sea por la cantidad de documentos o también por el tamaño de cada uno, este proceso es impracticable debido a restricciones de espacio o capacidad de cómputo.

A menudo, el texto se estructura en colecciones digitales de documentos cuya extensión (cantidad de caracteres) es variable, e.g. las páginas WEB o el contenido generado por los usuarios de redes sociales como Twitter. Para permitir el procesamiento de estas colecciones, primero se extrae el conjunto de palabras que aparece en ella y se ordena lexicográficamente; a este conjunto ordenado se le denomina vocabulario. Luego, el contenido de cada documento es representado algebraicamente por un vector, donde cada componente corresponderá a una de las palabras del vocabulario y cuyos valores estarán dados por las frecuencias de ocurrencia de cada palabra respectiva en el documento. Como consecuencia de la riqueza léxica del lenguaje, la cardinalidad del vocabulario es en general muy grande en comparación con los tamaños usados habitualmente por los algoritmos tradicionales de reconocimiento de patrones, i.e. partiendo del orden de los miles hasta los millones de palabras. Debido a esto, la tarea de agrupamiento automático de documentos tiene un alto costo de cómputo (tiempo de uso del procesador) y de almacenamiento (cantidad de memoria RAM y disco usado). Si a lo anterior se agrega que las cantidades de documentos en una colección puede superar los millones, entonces tanto las técnicas tradicionales de procesamiento, agrupamiento, así como también las capacidades computacionales actuales de una sola máquina resultan insuficientes o en el mejor de los casos los tiempos de respuesta son excesivos.

Existe tres enfoques exitosos para la construcción de algoritmos de agrupamiento sobre grandes volúmenes de datos. El primero consiste en introducir restricciones en la cantidad de accesos a un documento (una sola pasada), el segundo en usar el paralelismo disponible en las arquitecturas multi-core actuales o usar GPUs (tarjetas gráficas) y por último, particionar el conjunto de datos en mútlples máquinas para su procesamiento distribuido.

Esta última línea de trabajo es promisoria dado que permite aprovechar las capacidades locales de computadores que no necesariamente deben ser servidores de gran escala con múltiples procesadores. Dentro de este enfoque existen dos contextos respecto del origen de los datos que deben ser diferenciados. Por una parte, existen problemas donde la estrategia usada para hacer frente al gran volúmen de datos consiste en distribuirlo entre distintos computadores, lo que conlleva un traslado de datos durante la ejecución del algoritmo Nagwani (2015). Por otra parte, existe otro contexto en el que los datos se encuentran naturalmente distribuídos, e.g. colecciones documentales locales a distintas zonas geográficas, y donde además no es posible centralizarlos, debido a costos de transmisión o por motivos de privacidad Jagannathan et al. (2005); Liu et al. (2012).

2 Definición del Problema y Estado del Arte

Sea $X = \{X_1, X_2, \ldots, X_p\}$ una colección de documentos donde $\bigcup_{i=1}^{p} X_i = X$ y $\forall i \neq j \in [1, \ldots, p], X_i \cap X_j = \emptyset$. Esta colección se encuentra dividida en p máquinas distinas, y a su vez, cada subconjunto X_i se encuentra compuesto por n_i vectores en $\mathbb{R}^d$. Adicionalmente, los nodos o máquinas pueden compartir el mismo conjunto de atributos sobre los cuales se representan los datos, denominándose en este caso entorno homogéneo. El caso contrario, i.e. todos los nodos tienen los mismos datos pero representados sobre distintos conjuntos de atributos, se denomina entorno heterogéneo. La tarea de Clustering distribuído

consiste en obtener un conjunto de k particiones $C_1, C_2 \ldots, C_k$ donde $\bigcup_{i=1}^{k} C_i = X$ y además $\forall i \neq j \in [1, \ldots, k], C_i \cap C_j = \emptyset$, tal que cada una de ellas represente una o una parte de una categoría temática de la colección completa y a su vez todas las categorías esten representadas en los grupos o clusters C_i. Expresado de una manera distinta, esto significa que cada cluster contiene documentos más similares entre sí según su contenido que los que están en otros clusters. La similaridad entre vectores respresentates de documentos es usualmente medida mediante una función $S : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$ como por ejemplo la denominada similaridad del coseno:

$$S(x_a, y_b) = \frac{< x_a, x_b >}{\|x_a||x_b|}$$

, donde $< u, v >$ con $u, v \in \mathbb{R}^d$ denota el producto interno entre dos vectores y $|u|$ es la norma euclideana del vector.

En pocas palabras, esta tarea consiste en encontrar una estructura de grupos compactos respecto a la similaridad entre sus miembros y homogéneos respecto de su contenido, de acuerdo a alguna medida de similitud como puede ser la del coseno.

El modo general de operación de las técnicas de clustering sobre colecciones de datos distribuídas consiste de cuatro etapas: Inicialmente el conjunto de datos se encuentra particionado en varios nodos. Luego cada nodo genera un modelo de agrupamiento sobre su respectivo subconjunto local. Estos modelos son transmitidos a un nodo central (e.g. puede ser un conjunto de representantes identificados en cada grupo), el cual combina los modelos locales en una solución global. Opcionalmente, el modelo global puede ser transmitido a los otros nodos para que estos refinen sus modelos locales y se genere un nuevo modelo global depurado siguiendo las mismas etapas ya mencionadas. Este esquema se representa gráficamente en la figura 1.

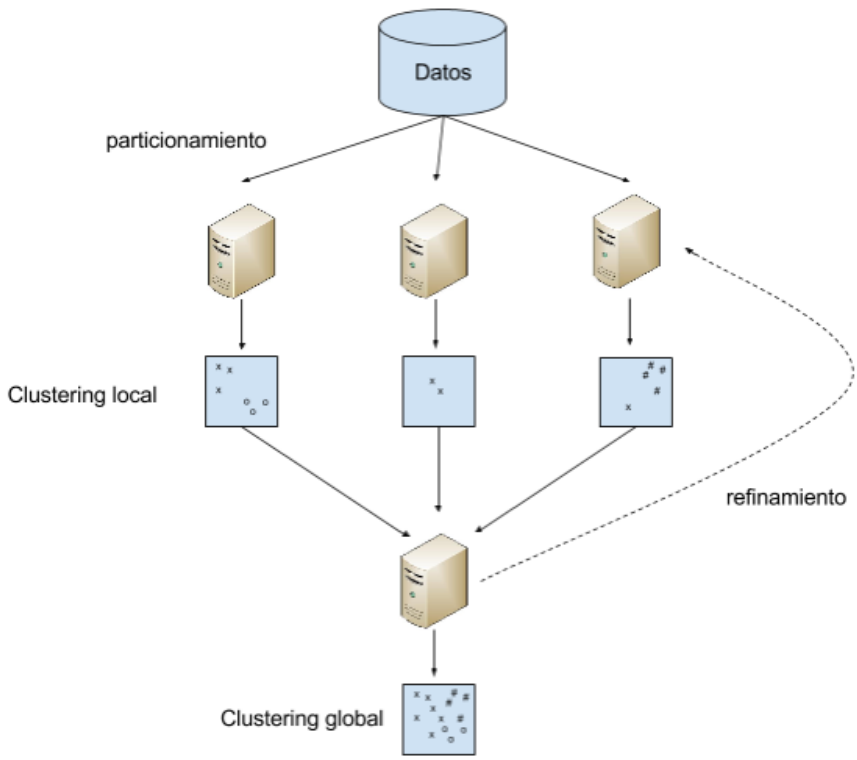

Figure 1: Esquema general del funcionamiento de las técnicas de clustering sobre datos distribuídos.

2.1 Enfoques paralelos y distribuídos para el agrupamiento de datos masivos y de alta dimensionalidad

Hasta donde sabemos, gran parte de los esfuerzos en la literatura para la construcción de técnicas de agrupamiento capaces de operar en entornos donde los datos están distribuídos se ha enfocado en datos con datos cuya dimensionalidad es relativamente baja (menos 100 atributos) en comparación con las colecciones documentales (más de 10^4 atributos). Sin embargo, a continuación detallamos los principales avances en el área de agrupamiento sobre datos distribuídos, destacando aquellas contribuciones dirigidas a los datos con alta dimensionalidad en sus representaciones computacionales.

1. Principales contribuciones en algoritmos paralelos Xu et al. (2002) y Dhillon and Modha (1999) proponen extensiones paralelas para los algoritmos DBSCAN y K-Means respectivamente. Otro enfoque escalable basado en memoria secundaria consiste en diseñar algoritmos capaces de operar dentro del marco de Mapreduce. En este contexto, es posible destacar las contribuciones de Das et al. (2007) que proponen una implementación del algoritmo EM y también de

Ene et al. (2011) que consideran el problema de las K-Medianas en Mapreduce[1]. Otro enfoque paralelo para K-Means se presenta en Bahmani et al. (2012) y se denomina K-Means. También destaca, por los volúmenes de documentos procesados, la técnica EM-Tree propuesta por De Vries et al. (2015) (cientos de millones de páginas WEB).

2. Enfoques que consideran alta dimensionalidad En Kargupta et al. (2001) los autores proponen un método para obtener componentes principales sobre datos heterogéneos distribuídos. Aprovechando este primer avance, también proponen un método de clustering que opera sobre conjuntos de datos heterogéneos y de alta dimensionalidad. Una vez obtenidas las componentes principales globales, se proyectan los datos locales de cada nodo y se aplica una técnica tradicional de clustering sobre estos. Finalmente, un nodo centraliza los clusters identificados y los combina para obtener el agrupamiento global. Varios años más tarde, Liang et al. (2013) presentan otro algoritmo de computo de componentes principales (PCA) sobre datos distribuídos. Para esto, cada nodo calcula PCA sobre sus datos locales y envía una fracción de estos, los que serán usados por un nodo central para estimar los componentes principales globales. Estos componentes globales son re-enviados a cada nodo. Luego, los datos locales de cada nodo serán proyectados, y estas proyecciones serán usadas para calcular un coreset mediante un algoritmo distribuído. Este coreset global construído sobre los datos proyectados será usado para la obtener la solución del clustering global.

   Li et al. (2003) proponen un algoritmo denominado D-CoFD para datos de alta dimensionalidad distribuídos tanto homogéneos como heterogéneos.

3. Enfoques basados en densidad Januzaj et al. (2003) proponen una técnica de clustering por densidad para datos distribuídos homogéneos. Los nodos construyen modelos locales y transmiten un conjunto de representates de cada cluster encontrado a un nodo central. En el nodo central se realiza un nuevo clustering sobre los representates, modelo que es retransmitido a los nodos locales para su actualización. Klusch et al. (2003) proponen una técnica de clustering sobre conjuntos de datos distribuidos homogéneos basada en estimadores de densidad. Januzaj et al. (2004) Proponen una versión de DBSCAN escalable y capaz de operar sobre colecciones distribuidas. Primero se seleccionan los mejores representantes locales dependiendo de que tantos datos representan en cada nodo y se envian a un nodo central. Este nodo agrupa estos representantes y los envía a cada nodo para que estos asocien sus datos locales a aquellos representantes más cercanos.

4. Enfoques basados en modelos paramétricos Merugu and Ghosh (2003) Proponen un método de clustering sobre datos distribuídos que combina múltiples modelos paramétricos generados por varios nodos sobre subconjuntos de datos. El foco de este trabajo recae también sobre el cuidado de la privacidad de los datos. Kriegel et al. (2005) proponen una técnica de clustering sobre datos distribuidos basada en un modelo paramétrico. Localmente, se identifican modelos de mezcla de gaussianas usando el algoritmo EM. Posteriormente, se mezclan estos modelos gaussianos locales para generar un modelo global.

5. Enfoques basados en representantes Forman and Zhang (2000) extienden esquemas paralelos basados en centroides para soportar la identificación de grupos sobre datos distribuídos. Específicamente extienden K-Means, K-Harmonic-Means y EM. Qi et al. (2008) proponen una técnica para la construcción de K-Medianas aproximadas sobre datos distribuidos en el escenario de streaming en que los datos se reciben de manera continua. Balcan et al. (2013) presentan técnicas de clustering basadas en centroides para datos distribuídos. Para esto proponen un método de construcción de coresets sobre datos distribuidos para K-Medias y K-Medianas. Naldi and Campello (2014) utilizan algoritmos evolutivos para solucionar dos problemas de K-Medias sobre datos distribuidos: La selección de prototipos iniciales y la identificación de la cantidad de clusters.

6. Enfoques que consideran agrupamiento jerárquico Johnson and Kargupta (2000) proponen una adaptación a un algoritmo de jerárquico para manejar conjuntos de datos distribuídos y heterogéneos. Este método supone que los datos son particionados verticalmente, por lo tanto todos los nodos disponen del mismo conjunto de observaciones, pero representados mediante diferentes conjuntos de atributos. Primero se generan dendrogramas en cada nodo, para luego ser transmitidos a un nodo central, el cual combina estos modelos locales en un modelo global. Jin et al. (2015) proponen un algoritmo de clustering jerárquico para datos distribuidos que además opera en modo incremental, i.e. incorpora nuevos datos recibidos sin requerir de re-construir nuevamente el modelo. Para ello reformulan el problema de clustering jerárquico como un problema de construcción del Minimum Spanning Tree en un grafo. Para esto proponen una técnica de combinación de múltiples Minimum Spanning Trees obtenidos sobre subgrafos disjuntos del grafo original. Este proceso de mezcla itera hasta que solamente quede un Minimum Spanning Tree, que será equivalente al agrupamiento jerárquico buscado originalmente.

3 Hipótesis

Un enfoque distribuido del tipo maestro/esclavo de algoritmos de agrupamiento (clustering) locales y paralelos de una pasada, obtendrán resultados comparables a un esquema centralizado, en términos de medidas de desempeño de pureza y entropía,

[1]Hadoop MapReduce es un software que permite escribir aplicaciones que procesen grandes cantidades de datos (terabytes) en paralelo sobre grandes Clusters de computadores.

para problemas donde los datos son altamente dimensionales, masivos y naturalmente distribuidos provenientes de bases de datos documentales.

4 Objetivos

El objetivo general de este trabajo consiste en desarrollar nuevas técnicas paralelas y distribuidas de Clustering para grandes volúmenes de documentos. En específico, este trabajo comprende los siguientes objetivos:

- Desarrollar un esquema paralelo agrupamiento de documentos que realice sólo una pasada sobre la colección documental, i.e. no utilice memoria secundaria para procesar la colección de documentos.
- Extender el esquema paralelo propuesto para entornos donde la colección de documentos se encuentra distribuida en varias máquinas.
- Validar los modelos propuestos tanto con datos de benchmark extraídos de sitios especializados, como con conjuntos de documentos reales (Noticias y textos de redes sociales).
- Divulgar los resultados alcanzados en esta investigación en publicaciones registradas en el catalogo ISI, específicamente una publicación aceptada y otra enviada.

5 Metodología

Para alcanzar las metas de esta propuesta, distinguimos las siguientes actividades generales y específicas:

1. Estudio y Discusión de la Literatura relevante a la propuesta En el desarrollo de esta investigación se realizará una revisión completa y constante de la literatura relacionada con:
   - Problemas teóricos y experimentales de agrupamiento de una pasada y paralelo sobre datos distribuidos.
   - Revisión de trabajos de Clustering, tanto paralelos y distribuidos,
   - Revisión de trabajos relacionados con estrategias de Hashing para estimación de vecindarios ya sea centralizados o distribuidos,
   - Estudio de algoritmos de clustering para datos masivos y
   - Revisión de trabajos relacionados con Computación Distribuída y Paralela.
   - Se realizará un Seminario sobre técnicas de clustering y aplicaciones sobre problemas específicos de aprendizaje distribuido, Big Data, métodos de ensamblado, con colegas y estudiantes. Este seminario incluirá discusiones con alumnos de pre y post grado.
   - Participación en conferencias nacionales e internacionales relacionadas con máquinas de aprendizaje, sistemas inteligentes distribuidos y descubrimiento de conocimiento desde grandes bases de datos.
   - Se mantendrá contacto con especialistas internacionales de temas de investigación relacionados, ya sea tanto del medio nacional, latinoamericano, como de Europa y América del norte.
2. Diseño de los algoritmos y modelo propuestos. Basado en la hipótesis de la propuesta reconocemos los siguientes pasos en la formulación y diseño de los modelos de la propuesta:
   - Modelamiento de ambientes donde existan datos distribuidos altamente dimensionales.
   - Construcción de los modelos propuestos: Desarrollo de un enfoque de procesamiento distribuido para algoritmos de Clustering de una sola pasada capaces de operar en escenarios donde los conjuntos de datos son representados por vectores altamente dimensionales y se encuentran naturalmente distribuidos.
   - Estudio de como resumir o sintetizar volúmenes masivos de datos con el objetivo de generar representaciones computacionales económicas (e.g. coresets o minwise signatures).
   - Descripción de las propiedades teóricas de los modelos: Durante esta fase nos enfocaremos en la generación de conocimiento centralizado a partir de modelos locales de Clustering.
3. Implementación y Optimización de los Algoritmos Propuestos Los algoritmos propuestos serán implementados en un nuevo lenguaje llamado JULIA (http://www.julialang.org). Desarrollaremos una estrategia compuesta de módulos capaces de ser evaluados e intercambiados independientemente. Durante esta fase de desarrollo, identificamos los siguientes pasos (los cuales pueden ser iterados): a) Construcción de una plataforma general para construir la arquitectura de los prototipos, b) Implementación de diversos escenarios distribuidos, c) Implementación de algoritmos del estado del arte con fines comparativos, d) Implementación de las técnicas de validación, e) Elaboración de soporte para clusters y tecnologías de Cloud Computing tales como MPI, OpenStack y Amazon EC2, con el fin de simular el problema en un ambiente real.

4. Diseño de los experimentos y validación de los algoritmos propuestos. Para validar los modelos propuestos, tenemos planificado usar tanto datos "benchmark" conocidos en el área, como datos provenientes de problemas reales de interés regional. Los datos de "benchmark" serán recolectados de sitios web de acceso publico del área de Máquinas de Aprendizaje. En la mayoría de los casos, los conjuntos de datos tienen asociada a cada documento una etiqueta que indica su verdadera clase. A las medidas de evaluación que comparan las etiquetas asignadas por el método de clustering con las verdaderas se les denomina medidas externas. Las medidas externas que usaremos con el fin de validar la propuesta son: Rand Score, Mutual Information, Homogeneity, Completeness y V-Measure. Para aquellos conjuntos de datos que no cuenten con etiquetas verdaderas, se usará la medida interna Silhouette Rousseeuw (1987), la cual cuantifica que tan bien diferenciados están los grupos encontrados por el método bajo evaluación. La etapa de validación será realizada mediante la corrida de 20 experimentos con particionamientos aleatorios para cada uno de los conjuntos de datos utilizados. Los métodos propuestos serán comparados con modelos del estado del arte en al menos 10 conjuntos de datos, tanto reales como sintéticos obtenidos de dos fuentes:

    - UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) el cual es el repositorio más extenso de datos usados en Machine Learning,
    - Datos provenientes de bases de datos documentales como Tipster y 20-Newsgroups.

6 Plan de Trabajo

Las etapas fueron descritas en el punto 5 y son las siguientes:

- *Etapa 1*: Estudio y discusión de la literatura relevante para la propuesta.
- *Etapa 2*: Diseño de los modelos y algoritmos propuestos:
    1. Modelamiento de ambientes con patrones distribuidos.
    2. Construcción de los modelos propuestos.
    3. Descripción teórica de las capacidades y propiedades de los modelos.
- *Etapa 3*: Implementación y optimización de los algoritmos propuestos.
- *Etapa 4*: Diseño de experimentos y estrategia de validación de los algoritmos propuestos.
    1. Generación de conjuntos de datos sintéticos.
    2. Recolección de datos reales y sintéticos.
    3. Procedimiento de validación.
    4. Análisis comparativo.
- *Etapa 5*: Diseminación de los resultados obtenidos en este proyecto.

Durante el año 2017 definiremos el modelo de clustering distribuido desde el punto de vista de la extracción de representantes desde cada fuente que permitan una posterior identificación de los grupos subyacentes a toda la colección (Etapa 1). Concentraremos esfuerzos en el estudio y construcción de técnicas basadas en Coresets y en distancias derivadas de vecinos más cercanos compartidos (Etapa 2 y 3). Por último, evaluaremos empíricamente y compararemos los algoritmos propuestos (Etapa 4).

Durante el año 2018 concentraremos nuestros esfuerzos en el estudio teórico de propiedades de los algoritmos que permitan justificar su desempeño e identificar escenarios menos favorables para su operación. Este análisis facilitará la comprensión de la contribución de los algoritmos propuestos para su posterior diseminación (Etapa 5).

7 Trabajo en Progreso

Una enfoque inicial de exploración para el diseño y construcción de algoritmos de clustering de colecciones documentales distribuidas consiste en una adaptación del algoritmo de Ertöz et al. (2003) para un contexto distribuído. En este esquema se ataca primero la alta dimensionalidad de los vectores mediante medidas de distancia basadas en la cantidad de vecinos que comparten dos puntos cualquiera. Por otra parte, la gran cantidad de datos y el ruído se ataca mediante la selección de representantes, denominados *Core-points*. Para esto se utilizan dos parámetros, **Eps** y **MinPts**. Luego, un punto será representante solamente si comparte más de **Eps** vecinos con más de **MinPts** puntos. Finalmente, se etiquetan con el mismo número de grupo aquellos *Core-points* que se compartan más de **Eps** puntos, y luego el resto de los puntos recibe la etiqueta de su *Core-point* más cercano en término de vecinos cercanos compartidos. En el caso de que el *Core-point* más cercano esté a distancia menor que **Eps**, se identifica como ruído.

El esquema anterior fue pensado en un escenario centralizado en donde es posible calcular las distancias entre todos los pares de puntos. Para un conjunto de datos con n puntos, cada uno representado por un vector en $\mathbb{R}^d$, el costo de calcular estas distancias es $O(n^2 * d)$ y de almacenarlas es $O(n^2)$. Al considerar grandes volúmenes de datos representados además por vectores altamente dimensionales, resulta claro que el desempeño del método difícilmente escale.

El trabajo que actualmente estamos realizando considera que la colección se encuentra particionada aleatoriamente en un conjunto de nodos o máquinas. En cada uno de estos nodos se usa el esquema de Ertöz et al. (2003) para identificar *Core-points* y etiquetar los datos locales. Tomando algunas ideas propuestas para la construcción de core-sets distribuídos por Balcan et al. (2013), se realiza en cada máquina una selección de *Core-points* aleatoria con pesos inversamente proporcionales a la cantidad de puntos con la etiqueta del *Core-point* y luego se transmiten los puntos seleccionados aun nodo central. Es importante mencionar que, a diferencia del trabajo en curso que se describe en esta sección, las contribuciones en esta línea no consideran en general datos de alta dimensionalidad (vectores con miles de atributos) ni tampoco grupos que pueden tener formas no esféricas. De esta manera, serán seleccionados con mayor probabilidad aquellos *Core-points* ubicados en grupos más pequeños, buscando así representar a todos los grupos independientemente de que tan pequeños sean. La cantidad de puntos seleccionados es un parámetro expresado en términos de porcentaje y es precisamente este mecanismo el que permite atacar los problemas generados por los grandes volúmenes de datos. Así en el nodo central se realizará nuevamente un clustering usando la medida de distancia basada en vecinos más cercanos compartidos, pero únicamente sobre una fracción de todos los *Core-points*. El agrupamiento final contendrá los grupos de la colección completa, lo que corresponderá al resumen esperado de la colección.

Actualmente, hemos realizado experimentos sobre datos sintéticos con formas esfericas y no esfericas, obteniendo resultados interesantes. Para poder realizar afirmaciones concluyentes, aplicaremos el método sobre colecciones documentales reales.

8 Recursos disponibles

Los recursos disponibles para este proyecto en la Escuela de Ingeniería Informática de la Pontificia Universidad Católica de Valparaíso son:

- Biblioteca de la Universidad y biblioteca personal del investigador patrocinante.
- Salón de seminarios.
- Oficina, Internet, Teléfono, etc.
- Servidor Dell R730, de 20 núcleos y 128 GB de RAM.

Existen varias licencias de software gratuitas en internet, como por ejemplo: Julia, Python, Java, R, Latex, etc.

References

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. (2012). Scalable K-Means ++. Proceedings of the VLDB Endowment (PVLDB), 5:622–633.

Balcan, M. F., Ehrlich, S., and Liang, Y. (2013). Distributed k -Means and k -Median Clustering on General Topologies. Advances in Neural Information Processing Systems 26 (NIPS 2013), pages 1–9.

Crestani, F., and Markov, I. (2013). Distributed Information Retrieval and Applications. 35th European Conference on IR Research, 865–868.

Das, A., Datar, M., Garg, A., and Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web, pages 271–280. ACM.

De Vries, C. M., De Vine, L., Geva, S., and Nayak, R. (2015). Parallel streaming signature em-tree: A clustering algorithm for web scale applications. In Proceedings of the 24th International Conference on World Wide Web, pages 216–226. ACM.

Dhillon, I. S. and Modha, D. S. (1999). A data-clustering algorithm on distributed memory multiprocessors. LargeScale Parallel Data Mining, 1759(802):245–260.

Ene, A., Im, S., and Moseley, B. (2011). Fast Clustering using MapReduce. Kdd, 681–689.

Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. Proceedings of the SIAM International Conference on Data Mining, 47–58.

Forman, G. and Zhang, B. (2000). Distributed data clustering can be efficient and exact. ACM SIGKDD Explorations Newsletter, 2(2):34–38.

Han, J., Pei, J., and Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

Jagannathan, G., and Wright, R. N. (2005). Privacy-preserving Distributed K-means Clustering over Arbitrarily Partitioned Data. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 593–599.

Januzaj, E., Kriegel, H.-P., and Pfeifle, M. (2003). Towards Effective and Efficient Distributed Clustering. Workshop on Clustering Large Data Sets, pages 49–58.

Januzaj, E., Kriegel, H.-P., and Pfeifle, M. (2004). Scalable Density-Based Distributed Clustering. pages 231–244.

Jin, C., Chen, Z., Hendrix, W., Agrawal, A., and Choudhary, A. (2015). Incremental, Distributed Single-linkage Hierarchical Clustering Algorithm Using Mapreduce. Proceedings of the Symposium on High Performance Computing, pages 83–92.

Johnson, E. and Kargupta, H. (2000). Collective, hierarchical clustering from distributed, heterogeneous data. Lecture Notes in Computer Science, 1759:221–244.

Kargupta, H., Huang, W., Sivakumar, K., and Johnson, E. (2001). Distributed clustering using collective principal component analysis. Knowledge and Information Systems, 3(4):422–448.

Klusch, M., Lodi, S., and Moro, G. (2003). Distributed clustering based on sampling local density estimates. IJCAI International Joint Conference on Artificial Intelligence, pages 485–490.

Kriegel, H.-p., Kr, P., Pryakhin, A., and Schubert, M. (2005). Effective and Efficient Distributed Model-based Clustering.

Li, T., Zhu, S., and Ogihara, M. (2003). Algorithms for Clustering High Dimensional and Distributed Data. Intelligent Data Analysis Journal, 7(February):1–36.

Liang, Y., Balcan, M.-f., and Kanchanapally, V. (2013). Distributed PCA and k-Means Clustering. The Big Learning Workshop in NIPS 2013, pages 1–8.

Liu, J., Huang, J. Z., Luo, J., and Xiong, L. (2012). Privacy Preserving Distributed DBSCAN Clustering. Proceedings of the 2012 Joint EDBT/ICDT Workshops, 177–185.

Merugu, S. and Ghosh, J. (2003). Privacy-preserving Distributed Clustering using Generative Models. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), pages 0–7.

Nagwani, N. K. (2015). Summarizing large text collection using topic modeling and clustering based on MapReduce framework. Journal of Big Data, 2:1–18.

Naldi, M. C. and Campello, R. J. G. B. (2014). Evolutionary k-means for distributed data sets. Neurocomputing, 127:30–42.

Qi, Z., Jinze, L., and Wei, W. (2008). Approximate clustering on distributed data streams. Proceedings - International Conference on Data Engineering, 00:1131–1139.

Rousseeuw Peter J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20:53–65.

Xu, X., Jäger, J., and Kriegel, H. (2002). A fast parallel clustering algorithm for large spatial databases. High Performance Data Mining, 290:263–290.