Informe telematica
Proyecto 4: Big Data

Christian Londoño Cañas
Universidad EAFIT
201510112010

Geralin Stefania Fernandez
Universidad EAFIT
201510033010

***Abstract*—Grandes problemas de la computación se ven limitados a los tiempos requeridos para solucionarlos, conociendo que actualmente la mayoría de los ordenadores y todos los centros de super computo del mundo tienen múltiples procesadores, comenzar a paralelizar los problemas es una opción muy llamativa para acortar tiempos, en este documento presentaremos diferencias entre una solución serial vs una paralela analizando tiempos de solución**

Key words: K-Means, Spark, Tf-idf, cluster

I. Introducción

Debido a la gran expansión que ha tenido la información en los últimos años, el análisis de esta con los métodos convencionales se ha vuelto inservible, debido a los gigantescos tiempos de espera que conllevaría hacer este proceso, para este tipo de problemas se propone la computación paralela, en este documento nos enfocaremos en los clusters, mas específicamente en un sistema de separación de archivos por similitud. Comenzaremos implantando un método convencional y presentaremos sus problemas en términos de eficiencia, pasando a una solución recursiva que mejorara los tiempos de ejecución una cantidad considerable.

Noviembre 23, 2017

II. Marco teórico

La minera de texto (text mining), es una de las técnicas de análisis de textos que ha permitido implementar una serie de aplicaciones muy novedosas hoy en da. Buscadores en la web (Google, Facebook, Amazon, Spotify, Netflix, entre otros), sistemas de recomendación, procesamiento natural del lenguaje, son algunas de las aplicaciones. Las técnicas de agrupamiento de documentos (clustering) permiten relacionar un documento con otros parecidos de acuerdo con alguna métrica de similar dad. Esto es muy usado en diferentes aplicaciones como: Clasificación de nuevos documentos entrantes al dataset, búsqueda y recuperación de documentos, ya que cuando se encuentra un documento seleccionado de acuerdo al criterio de búsqueda, el contar con un grupo de documentos relacionados, permite ofrecerle al usuario otros documentos que potencialmente son de interés para l. Algoritmos de agrupamiento patronal han sido reconocido como más adecuado en comparación con el jerárquico esquemas de agrupación para procesar grandes conjuntos de datos utilizando este método como respuesta a una problemática con la que nos enfrentamos claramente en la actualidad. un volumen de datos muy grande que crean desafíos de organización efectiva de textos. En la creación de estos algoritmos se parte siempre con un vector de palabras, este es sencillamente cada palabra del documento almacenada en un arreglo, además se debe tener cuenta que en el idioma ingles existen al menos 550 palabras consideradas ”STOPWORDS” las cuales no dan relevancia al texto, por ejemplo ”a,the,do” estas pueden estar 1000 veces entre todos los textos pero no aportan ninguna característica diferenciadora la manera en la que se busca una similitud entre documentos en esta práctica está dada por el peso de una palabra el cual se asigna bajo unos criterios específicos como la distancia euclidiana y la similitud del coseno.

1) Medidas de similitud: En general, las medidas de similitud-distancia mapean la distancia o similitud entre la descripción simbólica de dos objetos en un solo valor numérico, que depende de dos factores: las propiedades de los dos objetos y la medida misma.

2) Metricas: Métricas ya que no todas las medidas para la distancia pueden ser consideradas métricas estas tienen que cumplir 4 requisitos Sean x, y los dos objetos en un conjunto y $d(x, y)$ sea la distancia entre x, y.

- La distancia entre dos puntos cualesquiera debe ser no negativa, es decir, $d(x, y) = 0$.
- La distancia entre dos objetos debe ser cero si y solo si los dos objetos son idénticos, es decir $d(x, y) = 0$ si y solo si $x = y$.
- La distancia debe ser simétrica, es decir, la distancia desde x a y es lo mismo que la distancia de y a x, así que $d(x, y) = d(y, x)$.
- La medida debe satisfacer la desigualdad del triángulo que es $d(x, z) < d(x, y) + d(y, z)$.

A. Distancia Euclidiana

Este mide la distancia ordinaria entre dos puntos y puede se puede medir fácilmente con una regla en dos o tres dimensiones espacio, que es ampliamente utilizada en problemas de agrupación además de que es el predeterminado para la medida de distancia utilizada con el algoritmo K-means (1).

$$D_E(\vec{t_a}, \vec{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{\frac{1}{2}} \quad (1)$$

donde el trmino establecido es $T = t1, ..., tm$. Como se mencionamos anteriormente, utilizamos el valor de $tfidf$ como ponderaciones de terminos, es decir wt, $a = tfidf(da, t)$.

B. Cosine Similarity

Representando como vectores de términos los documentos, la similitud corresponde a la correlación entre los vectores. Esto se cuantifica como el coseno del Angulo entre vectores, es decir, la llamada similitud coseno. Dados dos documentos ta y tb, su similitud coseno es (2):

$$SIM_C(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a}.\vec{t_b}}{|\vec{t_a}| \times |\vec{t_b}|} \tag{2}$$

donde ta y tb son vectores m dimensiones en el conjunto de términos $T = t1, ..., tm$. Cada dimensión representa un termino con su peso en el documento, que no es negativo. Como resultado, la similitud del coseno es no negativa y limitada entre $[0, 1]$. Algo muy importante es el hecho de que Una propiedad importante de la similitud del coseno es su independencia de la longitud del documento.

C. Jaccard Coefficient

El coeficiente de Jaccard, que a veces se conoce como el Coeficiente de Taminoto, mide la similitud como la intersección dividido por la unión de los objetos. Para el documento de texto, el coeficiente de Jaccard compara la suma de peso de los términos compartidos al peso de la suma de los términos que están presentes en cualquiera de los dos documentos, pero no son los términos compartidos.

$$SIM_J(\vec{t_a}, \vec{t_b}) = \frac{\vec{t_a}.\vec{t_b}}{|\vec{t_a}|^2 + |\vec{t_b}|^2 - \vec{t_a}.\vec{t_b}} \tag{3}$$

El coeficiente de Jaccard es una medida de similitud y rangos entre 0 y 1. Es 1 cuando el $ta = tb$ y 0 cuando $taytb$ son disjuntos, donde 1 significa que los dos objetos son lo mismo y 0 significa que son completamente diferentes

D. Pearson Correlation Coefficient

El coeficiente de correlación de Pearson es otra medida de la medida en que dos vectores están relacionados. Hay diferentes formas de la formula del coeficiente de correlación de Pearson. Dado el termino establece $T = t1, ..., tm$ una forma comúnmente utilizada es:

$$SIM_P(\vec{t_a}, \vec{t_b}) = \frac{m \sum_{t=1}^{m} wt_a - TF_a \times TF_b}{\sqrt{[m \sum_{t=1}^{m} {w_{t,a}}^2 - {TF_a}^2][m \sum_{t=1}^{m} {w_{t,b}}^2 - {TF_b}^2]}} \tag{4}$$

Donde:

$$TF_a = \sum_{t=1}^{m} w_{t,a} \tag{5}$$

Y:

$$TF_b = \sum_{t=1}^{m} w_{t,b} \tag{6}$$

Esta es tambien una medida de similitud. Sin embargo, a diferencia del otras medidas, va de +1 a -1 y es 1 cuando $ta = tb$.

E. Averaged Kullback-Leibler Divergence

La similitud de dos documentos se mide como la distancia entre el dos distribuciones de probabilidad correspondientes. The KullbackLeibler divergencia (divergencia KL), también llamada relativa entropía, es una medida ampliamente aplicada para evaluar las diferencias entre dos distribuciones de probabilidad. Dadas dos distribuciones P y Q, la divergencia KL de distribución P, a la distribución Q se define como:

$$D_{KL}(P||Q) = Plog(\frac{P}{Q}) \tag{7}$$

En el escenario del documento, la divergencia entre dos distribuciones de palabras es:

$$D_{KL}(\vec{t_a}||\vec{t_b}) = \sum_{t=1}^{m} w_{t,a} \times log(\frac{w_{t,a}}{w_{t,b}}) \tag{8}$$

Sin embargo, a diferencia de las medidas anteriores, la divergencia KL no es simétrica, es decir. $DKL(PkQ) = DKL(QkP)$. Por lo tanto, no es una verdadera métrica. Como resultado usamos el KL promediado divergencia en cambio, que se define como:

$$D_{AvgKL}(P||Q) = \pi_1 D_{KL}(P||M) + \pi_2 D_{KL}(Q||M) \tag{9}$$

donde $1 = P/(P + Q)$,2 $= Q/(P + Q)$ y $M = 1P + 2Q$ Para los documentos, la divergencia promedio de KL se puede calcular con la siguiente fórmula:

$$D_{AvgKL}(\vec{t_a}||\vec{t_b}) = \sum_{t=1}^{m} (\pi_1 \times D(w_{t,a}||w_t) + \pi_2 \times D(w_{t,b}||w_t)) \tag{10}$$

Donde:

$$\pi_1 = \frac{w_{t,a}}{w_{t,a} + w_{t,b}}, \pi_2 = \frac{w_{t,b}}{w_{t,a} + w_{t,b}} \tag{11}$$

Y:

$$w_t = \pi_1 \times w_{t,a} + \pi_2 \times w_{t,b} \tag{12}$$

La ponderación promedio entre dos vectores asegura la simetría, es decir, la divergencia del documento i al documento j es lo mismo que la divergencia del documento j al documento i. La divergencia KL promediada se ha aplicado recientemente agrupar documentos de texto, como en la familia de Algoritmos de agrupamiento de cuellos de botella.

F. TFIDF

TF-IDF son las siglas en inglés de "Term frequency – Inverse document frequency lo que al traducirse a el español es "Frecuencia de términos – Frecuencia inversa del documento", y el peso de tf-idf es un peso que a menudo se utiliza en la recuperación de información y minería de texto. Este es una medida estadística utilizada para evaluar la importancia de una palabra para un documento en una colección. La importancia depende completamente del número de veces que aparezca una palabra en un documento, y esto se compensa con la frecuencia de la palabra en el mismo. K-means y sus variaciones son ampliamente

utilizadas en el campo ya que estas son elementos centrales en buscadores y todo tipo de aplicaciones en minería de datos.

La suma de Tf-idr es una de las funciones de clasificación más sencillas, esta se calcula sumando el tf-idr para cada termino de consulta; además de que existen un montón de funciones de clasificación que son variantes de este. La suma de Tf-idr es una de las funciones de clasificación más sencillas, esta se calcula sumando el tf-idr para cada termino de consulta; además de que existen un montón de funciones de clasificación que son variantes de este.

Este método puede ser utilizado con éxito para el filtrado de palabras vacías en los diferentes campos temáticos, como el resumen o la clasificación de texto.

El campo en el que se emplea tf-idf son el campo de información y minería de texto ya que estos usan elementos tales como las bibliotecas digitales y que esta directamente relacionado con los buscadores que utilizan variaciones de este algoritmo en sus procesos como induración, posicionamiento y la muestra de contenido especifico a los usuarios.

TF: este es la frecuencia en la que un termino determinado aparece en un documento, así que en pocas palabras este es el numero de veces en el que una palabra aparece en un texto.

"TF = Nº Total de la KW en el documento / Nº Total de palabras en el documento"

IDF: Este es el encargado de disminuir el peso de los términos que se repiten mucho en los documentos, dándole asi mayor valor a las que tienen una menor frecuencia.

"IDF = Nº Total de documentos / Nº de documentos con la KW"

Las formulas siguientes son las utilizadas para calcular la relevancia de un documento al compararse con los demás documentos que compartes palabras claves. Para que los resultados obtenidos sean realmente útiles se debe calcular para todas las palabras relevantes en un texto, además que entre más grande sea la base de datos usada para el cálculo más precisos son los resultados obtenidos.

$$TF(i) = \frac{log_2(Freg(i,j)) + 1}{log_2(L)} \tag{13}$$

$$IDF_t = log(1 + \frac{N_D}{f_t}) \tag{14}$$

Una desventaja importante a la hora de usar TF-IDF es que hay que considerar que este no contempla la posibilidad de que los términos evaluados aparezcan agrupados, que se apliquen normas de lexema o que se estén usando sinónimos.

G. K-MEANS

"K-means (MacQueen, 1967) es uno de los algoritmos de aprendizaje sin supervisión más simples que resuelven el conocido problema de agrupamiento. El procedimiento sigue una forma simple y fácil de clasificar un conjunto de datos dado a través de un cierto número de clusters (supongamos k clusters) fijados a priori."

La agrupación en K-means se utiliza cuando se tienen datos sin categoría o grupos definidos, su objetivo es encontrar conjuntos de datos, que cumplan con el numero de grupos definidos por K. Este es un algoritmo iterativo para asignar cada punto de datos a los correspondientes grupos de k cumpliendo con las características proporcionadas. Los datos son agrupados según la similitud de las características. Los resultados que arroja el algoritmo de K-means son:

- Los centroides de los clusters K, los cuales se pueden utilizar para hallar nuevos clusters
- Etiquetas para los datos de entrenamiento (cada punto de datos se asigna a un solo grupo)

La función objetivo de este algoritmo es

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i{}^j - c_j \right\|^2$$

(15)

En el que $\left\| x_i{}^j - c_j \right\|^2$ es una medida de distancia escogida entre un punto de datos y el centro del clúster, este indica la distancia que se encuentra en los puntos de datos y sus respectivos custres. El algoritmo se compone de los siguientes pasos:

- Paso de asignación de datos: En este paso cada centroide define uno de los clusters, Y luego cada punto de datos se asigna a su centroide más cercano, llendo en función de la distancia asignada.
- Paso de actualización del centroide: Aquí los centroides se recalcular. Esto se hace tomando la media de todos los puntos de datos asignados al clúster de ese centroide. Se repiten los pasos hasta que los centroides ya no se muevan. Produciendo una separación de los objetos en grupos a partir de los cuales se puede calcular la métrica que se va a minimizar.

Se debe tomar en cuenta: Se garantiza que este algoritmo converge a un resultado. Aunque se puede demostrar que el procedimiento siempre terminará, el algoritmo de k-medias no necesariamente encuentra la configuración más óptima, que corresponde al mínimo de la función objetivo global. El algoritmo también es significativamente sensible a los centros de clúster iniciales seleccionados aleatoriamente. El algoritmo k-means se puede ejecutar varias veces para reducir este efecto.

H. Analisis

1) Extraer: Obtenemos la información del hdfs y la almacenamos en tipo de dato similar a los mapas o un PairRDD, en donde guardamos la ruta del archivo y el contenido de este. Los datos bases usados para esta extracion estan en: *hdfs:///user/clondo46/datasets/gutenber*
Aunque durante el desarrollo se realizaron pruebas con otros datasets

2) Transformar: En esta fasa primeramente se implemento el TF guardandolo en un RDD, con los datos obtenidos anteriormente luego el IDF y finalmente se hizo el K-Means Con spark es posible desarrolar esto facilmente, Spark es una tecnologia desarrollada para el procesamiento de big data que usa memoria para mejorar el rendimiento.

3) Cargar: Se carga la información obtenida a una carpeta en el HDFS

I. Implementación

```
import sys

from pyspark import SparkContext
from pyspark.mllib.feature import HashingTF, IDF
from pyspark.mllib.clustering import KMeans
```

Fig. 1. Librerias

Librerias usadas para este proyecto resaltar las de pyspark que son las que nos permitiran realizar grandes procesos en paralelo y desarrollar funciones explicadas posteriormente En

```
dirs = "hdfs:///user/clondo46/datasets/gutenberg"
k = 5
maxIters = 20
sc = SparkContext(appName="Proyecto04")
```

Fig. 2. Librerias

dirs tenemos los archivos que usaremos del HDFS
K son los Clusters usados
MaxIters es el control para el K-means, este se encarga para que en caso de que no se estabilizen los cluster termina cuando se cumpla esa condición
SC Es el contexto de spark, necesario para realizar todas las operaciones Documentos es un PairRDD que contiene la

```
documentos = sc.wholeTextFiles(dirs)
nombreDocumentos = documentos.keys().collect()
docs = documentos.values().map(lambda doc: doc.split(" "))
```

Fig. 3. Librerias

ruta de los archivos como su contenido NombreDocumentos contiene la ruta de los archivos Docs es el contenido de los archivos En estos dos ultimas partes del codigo esta todo el

```
hashingTF = HashingTF()
tf = hashingTF.transform(docs)
idf = IDF().fit(tf)
tfidf = idf.transform(tf)
```

Fig. 4. Librerias

```
clusters = KMeans.train(tfidf,k,maxIterations=maxIters)
clustersid = clusters.predict(tfidf).collect()
diccionario = dict(zip(nombreDocumentos, clustersid))
d = sc.parallelize(diccionario.items())
d.coalesce(1).saveAsTextFile("hdfs:///user/clondo46/gut5")
```

Fig. 5. Librerias

algoritmo en donde se calcula el tfidf y luego se hace uso del K-Means

III. Conclusiones

- Spark permite solucionar grandes problemas matematicos de una manera sencilla y elegante debido a su amplia variedad de librerias
- Existen diferentes formas para hacer Clustering, conocer las herramientas que tienes permiten un desarrollo mas agil

References

[1] A. Huang, *Similarity measures for text document clustering*,The University of Waikato, Hamilton, New Zealand.

[2] A. Trevino,*Introduction to K-means clustering*, www.datascience.com/blog/k-means-clustering, 2016.

[3] J. Lopez,*Qué es el TF-IDF y qué relación tiene con el SEO*, http://www.iebschool.com/blog/que-es-el-tfidf-relacion-seo-sem/, 2017.