A COMPARASION OF EXPERIMENTS WITH THE BISECTING-SPHERICAL K-MEANS CLUSTERING AND SVD ALGORITHMS

Análisis del Artículo

Antonio Carlos Alcalde Aragonés,
Rocío Carratalá Sáez,
Mihaita Alexandru Lupoiu

18 Enero 2016

Librerías de Altas Prestaciones Para Problemas Algebraicos Dispersos (LAPPAD)

TRABAJO PROPUESTO

1. Contexto
2. Preproceso
3. IR con SVD
4. Clustering Algorithms
5. Caso de estudio
6. Conclusiones

CONTEXTO

CONTEXTO

Recuperación de la información (IR) es la actividad de obtener información de recursos pertinentes a una necesidad de información desde una colección de recursos de información.

CONTEXTO

Para poder extraer de forma efectiva los documentos relevantes con las estrategias IR, los documentos son transformados a una representación lógica de los mismos.

Cada estrategia de recuperación incorpora un modelo específico para sus propósitos de representación de los documentos.

Los usados en los experimentos de este trabajo usan modelos algebraicos que usan el espacio vectorial.

PREPROCESO

Stemming es un método para reducir una palabra a su raíz o (en inglés) a un stem.

Hay algunos algoritmos de *stemming* que ayudan en sistemas de recuperación de información.

Por ejemplo una consulta sobre ”bibliotecas” también encuentra documentos en los que solo aparezca ”bibliotecario” porque el *stem* de las dos palabras es el mismo (”bibliotec”).

PREPROCESO

Los algoritmos seleccionados para las pruebas han sido Paice, Porter y Lovins.

Porter es considerado un algoritmo de *stemming* ligero mientras que Paice y Lovins son considerados algoritmos de *stemming* pesados.

Por lo tanto no se deberían de comparar en cuanto a exactitud, porque están diseñados para tareas distintas, pero sí en cuanto a rendimiento.

PREPROCESO

	Porter	Lovins	Paice
Nº Input terms	49528	49656	49656
Nº. output terms	41283	35029	33297
Reduction	16.65%	29.45%	32.94%

PREPROCESO

Se han incluido al preproceso un conjunto de heurísticas sencillas para eliminar términos que solo aparecen en un conjunto de documentos pequeño.

PREPROCESO

Nº. Docs.	% Docs	Number Terms	Size Reduction	Number Non Zeros
0	0 %	12476	0 %	81082
1	0.24 %	6545	47.54 %	75151
2	0.47 %	4803	61.50 %	71667
3	0.71 %	3903	68.72 %	68967
4	0.94 %	3321	73.38 %	66639
5	1.18 %	2913	76.65 %	64599

El criterio para comparar la reducción heuristica ha sido la media del ratio precisión-exhaustividad:

$$\overline{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

Precisión es la fracción de instancias recuperadas que son relevantes.

Exhaustividad es la fracción de instancias relevantes que han sido recuperadas.

PREPROCESO

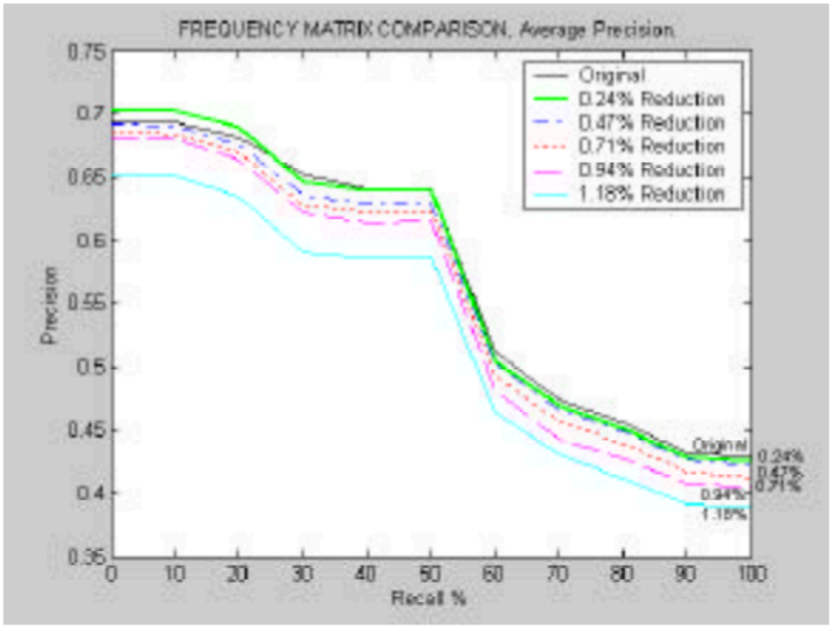


Figure 2. Frequency Matrix Average Precision Comparison.

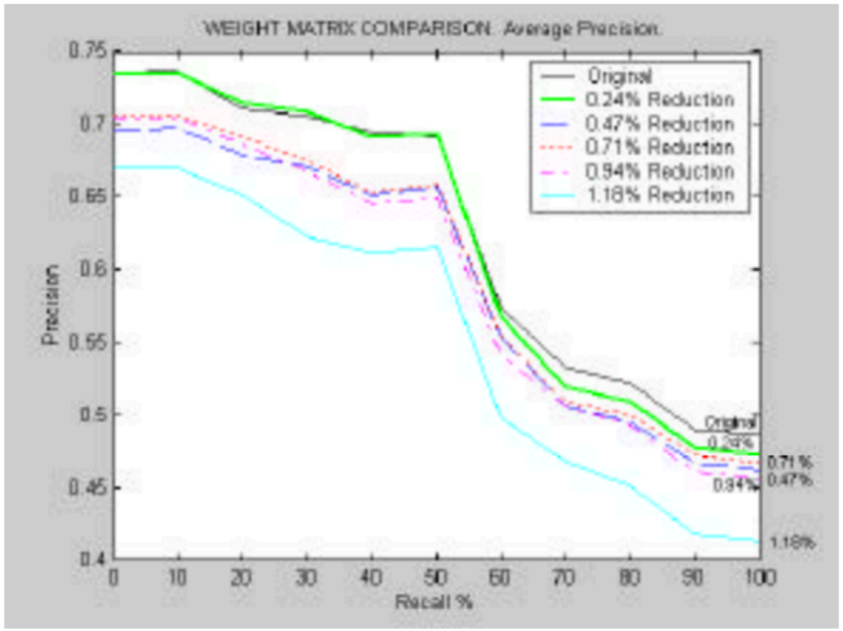


Figure 3. Weight Matrix Average Precision Comparison.

IR CON SVD

IR CON SVD

En álgebra lineal, la **descomposición en valores singulares** (SVD) de una matriz real o compleja es una factorización de la misma con muchas aplicaciones en estadística y otras disciplinas.

IR CON SVD

En recuperación de la información se realiza sobre una matriz para determinar patrones en las relaciones entre los términos y los conceptos contenidos en el texto.

IR CON SVD

$$M = U\Sigma V^T = \sum_{i=1}^{m} u_i \sigma_i v_i^T$$

IR CON SVD

Es suficiente e incluso mejor calcular solo parte del espectro de los valores singulares de la matriz.

IR CON SVD

$$M \approx M_p = U_p \Sigma_p V_p^T = \sum_{i=1}^{p} u_i \sigma_i v_i^T$$

IR CON SVD

Existen distintos métodos para resolver el problema de SVD completo, parcial o cuando la matriz es dispersa.

Para matrices dispersas se usarán métodos iterativos como el de Arnoldi o Lanczos.

Las librerías LAPACK Y ARPACK implementan SVD de forma eficiente tanto para matrices dispersas como para matrices densas.

A la hora de evaluar las consultas de SVD se utilizan 3 matrices que son U, $\sum$, V en lugar de una solo como en el caso del modelo de espacio vectorial.

$$\cos\theta_j = \frac{m_j^T q}{\|m_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} m_{ij} q_i}{\sqrt{\sum_{i=1}^{m} m_{ij}^2} \sqrt{\sum_{i=1}^{m} q_i^2}} \quad j = 1 \ldots n$$

M_j es el vector del documento y q es la columna de la palabra y n el numero de documentos.

Como de normal ambos términos están normalizados $|m_j| = |q| = 1$ la ecuación se puede simplificarse de la siguiente manera:

$$\cos\theta_j = m_j^T q = \sum_{i=1}^{m} m_{ij} q_i$$

M_j es el vector del documento y q es la columna de la palabra y n el numero de documentos.

$$\cos\theta_j = \frac{m_j^T q}{\|m_j\|_2 \|q\|_2} = \frac{\left(M_p e_j\right)^T q}{\|M_p e_j\|_2} =$$

$$= \frac{\left(U_p \Sigma_p V_p^T e_j\right)^T q}{\|U_p \Sigma_p V_p^T e_j\|_2} =$$

$$= \frac{\left(e_j^T V_p \Sigma_p\right)^T \left(U_p^T q\right)}{\|\Sigma_p V_p^T e_j\|_2} = \frac{s_j^T \left(U_p^T q\right)}{\|s_j\|_2}$$

CLUSTERING ALGORITHMS

El Spherical k-means clustering algorithm intenta encontrar los k clusteres disjuntos tries to find k disjoint clusters $\pi_{j_{j-1}}^{k}$ de la matriz M de tal manera que se intenta maximizar la función:

$$f\left(\{\pi_j\}_{j=1}^{k}\right) = \sum_{j=1}^{k} \sum_{m \in \pi_j} m^t c_j$$

$$t_j = \frac{1}{n_j} \sum_{m \in \pi_j} m, \; c_j = \frac{t_j}{\|t_j\|},$$

Encontrar la solución optima en un cluster es un problema NP completo.

Algorithm 1. Spherical k-means algorithm

Step 1. Calculate k initial clusters with the bisection technique described in algorithm 2, $\{\pi^{(0)}{}_j\}_{j=1}^k$ and its concept vectors $\{c^{(0)}{}_j\}_{j=1}^k$
Initialise $t=0$.

Step 2. Calculate the new partition $\{\pi^{(t+1)}{}_j\}_{j=1}^k$ induced by the concept vector $\{c^{(t)}{}_j\}_{j=1}^k$:

$$\pi_j^{(t+1)} = \left\{ m \in \{m_j\}_{i=1}^n : m^T c_j^{(t)} > m^T c_l^{(t)}, \; 1 \le l \le n, l \ne j \right\}, \quad 1 \le j \le k$$

Step 3. Calculate the concept vectors associated to the new clusters $\{c^{(t+1)}{}_j\}_{j=1}^k$, using expression (3)

Step 4. When the stopping criteria is fulfilled, store $\{\pi^{(t+1)}{}_j\}_{j=1}^k$ and $\{c^{(t+1)}{}_j\}_{j=1}^k$. In other case increment $t=t+1$ and go to step 2

El criterio de parada utilizado es el siguiente:

$$\frac{\left| f(\{\pi^{(t)}{}_j\}_{j=1}^k) - f(\{\pi^{(t+1)}{}_j\}_{j=1}^k) \right|}{\left| f(\{\pi^{(t+1)}{}_j\}_{j=1}^k) \right|} \leq \varepsilon$$

Donde el error es $2x10^{-2}$

Algorithm 2. Iterative Bisecting method for the determination of k-initial clusters

Step 1.
Select a sparse vector c_l, where $c_l \in \Re^n$ with *nnz(M)/(n*m)* density, where *nnz* is the number of non-zero elements of matrix *M*, *m* is the number of terms and *n* is the number of documents.
Select a maximum number of iterations, *maxiter*, and the convergence criteria, *tol*.
Initialise *iter*=0

Step 2. Divide M into two sub-clusters M_α and M_β according to:

$$m_i \in M_\alpha \quad si \quad m_i^T c_1 \geq \alpha$$

$$m_i \in M_\beta \quad si \quad m_i^T c_1 < \alpha$$

Step 3. Calculate the concept vector of M_α, given c_l define as indicated in expression (3)

Step 4. Stop when the stopping criteria has been fulfilled or when it has been achieved the maximum number of iterations defined by *maxiter* and take M_α, c_α,$=c_l$ and M_β . In other case increment *iter=iter+1* and goto step 2

CASO DE ESTUDIO

CASO DE ESTUDIO

425 documentos con una media de 546 palabras y 53 líneas por documento pertenecientes al Time Magazine, año 1963

Los contenidos eran especialmente políticos y enmarcados en la Guerra Fría.

Palabras frecuentes: WORLD, AFRICAN, NASSER, U.S, U.N, POLITICAL, CHINA, REGIME, NATO, SAID, COMMUNIST, EUROPE, NUCLEAR, GERMANY, KHRUSHCHEV, GAULLE, PRESIDENT, SOVIET, MOSCOW

COMPARATIVA: FRECUENCIA VS. MATRICES DE PESOS

Curva de la precisión media obtenida de las matrices de frecuencia (Mf) y de las de peso (Mw), ambas con una reducción óptima.

Claramente la precisión de la matriz de pesos es superior.

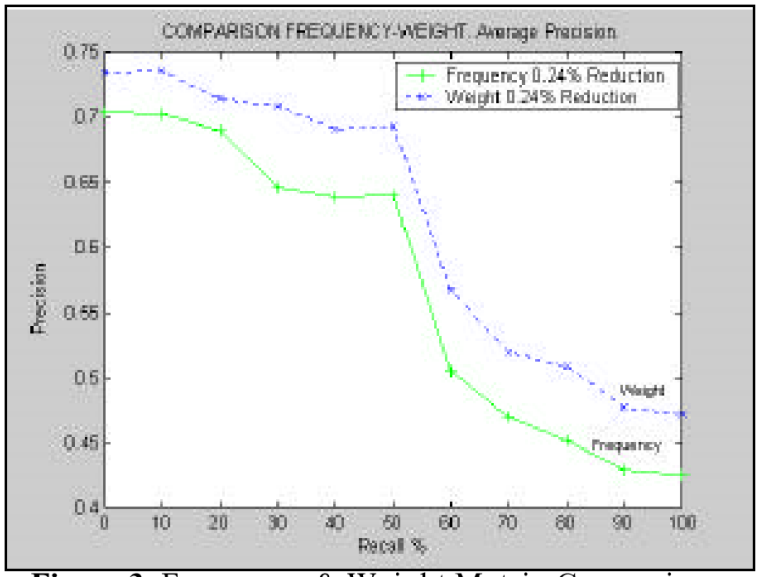
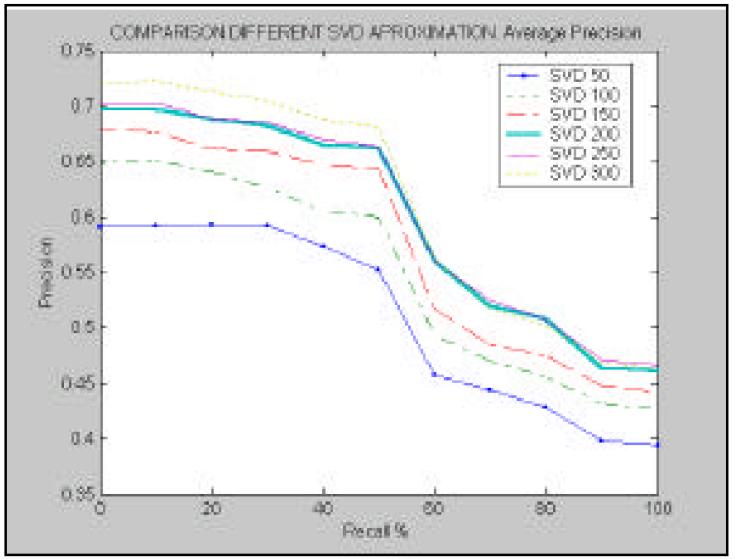


Figure 3. Frequency & Weight Matrix Comparison.

COMPARATIVA: SVD VS. MATRICES DE PESOS (I)

Conforme aumenta el rango de aproximación (cantidad de valores singulares calculados), la precisión aumenta.



No obstante, conviene tomar una cantidad no demasiado elevada para no necesitar grandes cantidades de memoria y cómputo.

COMPARATIVA: SVD VS. MATRICES DE PESOS (II)

Así pues, tomando 200 valores singulares (Msvd), la comparativa queda:

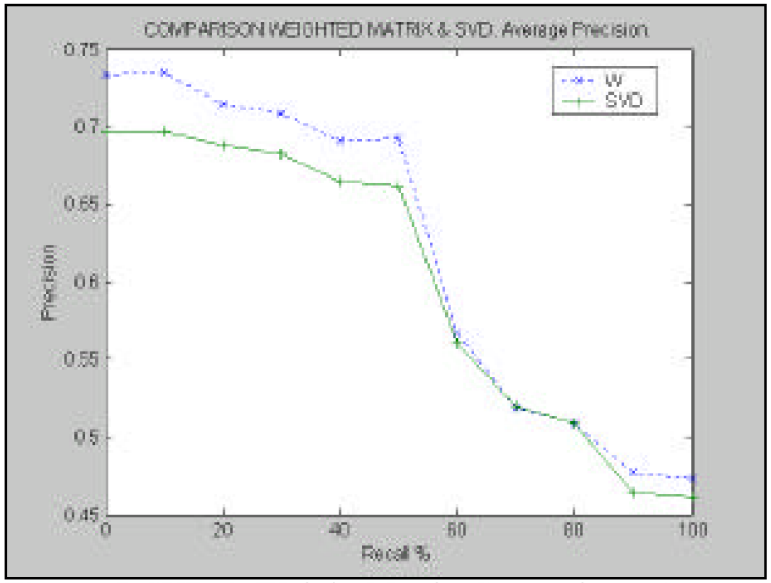


Figure 5. SVD & Weight Matrix Comparison. Average Precision.

Aunque la curva de Mw parece ofrecer mejores prestaciones, como son muy cercanas conviene estudiar esto con más detalle.

COMPARATIVA: SVD VS. MATRICES DE PESOS (III)

Utilizando como criterio de comparación la R-Precisión tenemos:

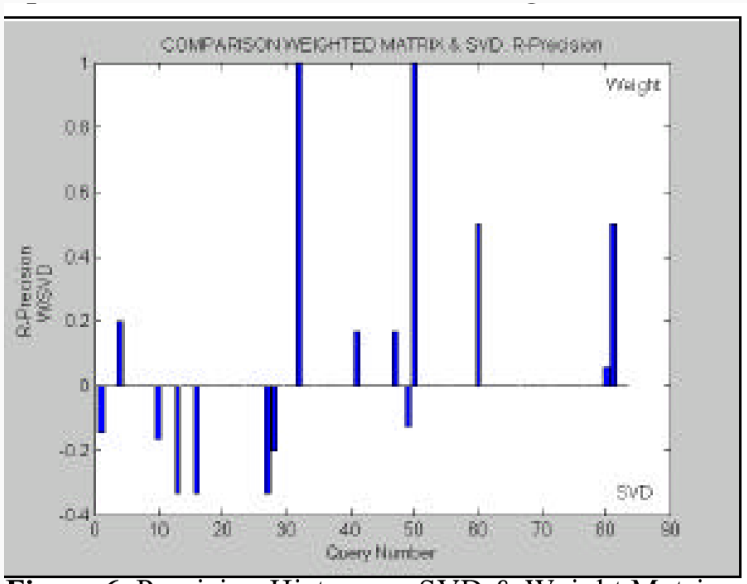


Figure 6. Precision Histogram SVD & Weight Matrix

Analizando varias de las *queries* estudiadas con ambos procesos (85% del total), pueden obtenerse resultados interesantes y se observa que con casos de estudio pequeños, los resultados son similares, pero con SVD puede reducirse el almacenamiento necesario.

COMPARATIVA: SVD VS. BISECTING-SPHERICAL K-MEANS CLUSTERING ALGORITHMS

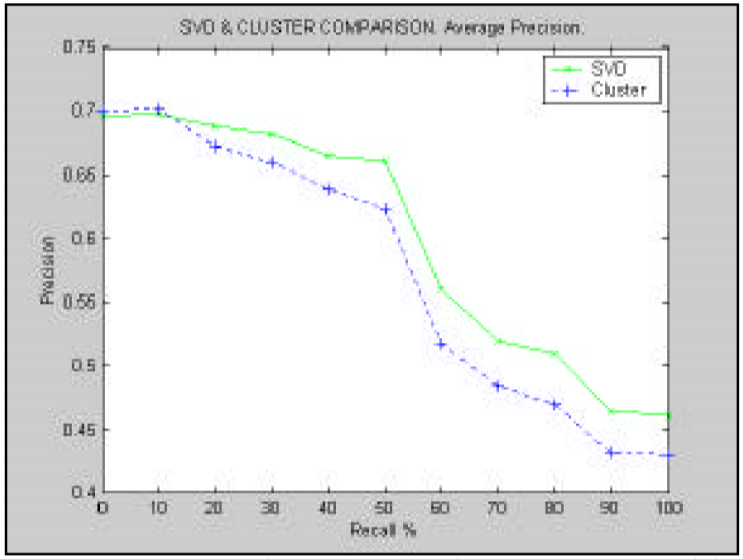


Figure 12. SVD & Cluster Comparison. Average Precision.

Las precisiones son similares, pero hay que destacar que los algoritmos de Clustering son más rápidos y requieren menor tiempo de cómputo.

CONCLUSIONES

CONCLUSIONES

- Aunque las matrices de peso reducidas parecen presentar mejores resultados, no es factible aplicarlas en todos los casos, pues necesitan una gran cantidad de datos para sus operaciones y, en casos reales, los conjuntos de datos sobre los que trabajar son enormes.
- Se recomienda utilizar pues, métodos basados en aproximaciones de menor rango, como SVD o Clustering.
- Los métodos de Clustering dependen mucho de la selección inicial de parámetros, por lo que debe hacerse correctamente, no obstante, es cierto que las particiones obtenidas con distintos parámetros son similares.
- La principal ventaja de los métodos de Clustering es que ofrecen respuestas en un tiempo menor porque no necesitan comparar todos los datos sino los grupos (clusters) formados.
- Utilizando SVD se aborda la información semántica de la colección de datos y se reduce el almacenamiento necesario para matrices de gran tamaño.

¡Gracias!

¿Alguna pregunta?