Algoritmos para Big Data:

Grafos y PageRank

Sergio García Prado

E.T.S. Ingeniería Informática, UVa

Índice General

Introducción

Big Data

Los algoritmos para **Big Data** son aquellos que se encargan de resolver problemas sobre conjuntos de datos de tamaño masivo.

Problema de Accesos a Memoria

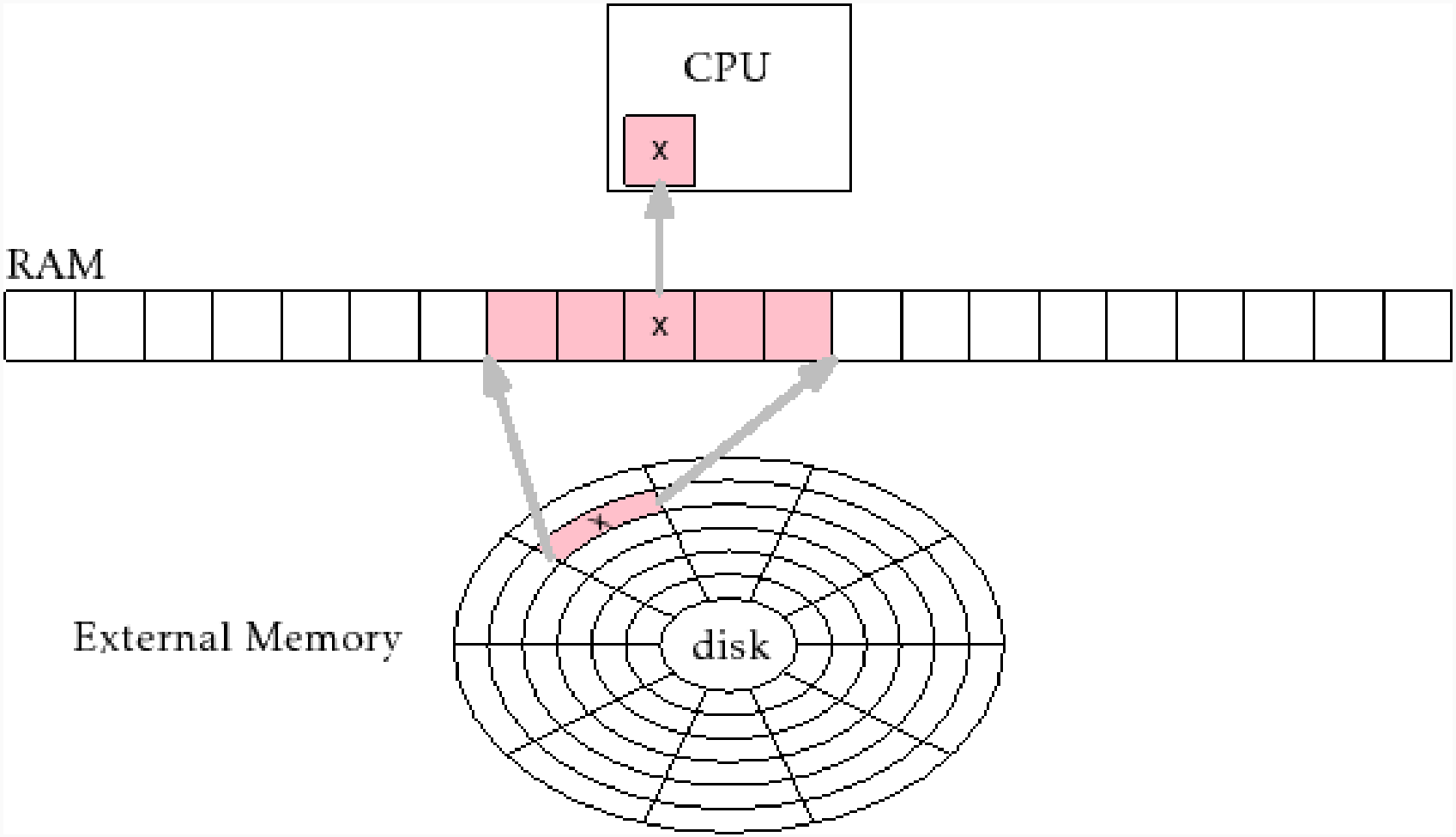


Figura 1

Soluciones a la complejidad del Big Data

- Algoritmos para Streaming
- Técnicas de Reducción de la Dimensionalidad
- Paralelización a Gran Escala
- ...

Algoritmos para Streaming

Modelo en Streaming

- Serie Temporal:
  $1, 5, 3, -4, 2, -3, 5, ...$
- Caja Registradora (Cash-Register):
  $(2, +1), (3, +4), (1, +3), (2, +3), (4, +5), ...$
- Molinete (Turnstile):
  $(2, -1), (3, -4), (1, +3), (2, -3), (4, +5), ...$

Algoritmos para Streaming

Los **Algoritmos para Streaming** son aquellos que procesan la entrada de manera secuencial, teniendo en cuenta únicamente el elemento actual, junto con una estimación de los procesados anteriormente, utilizando un orden sublineal $o(n)$ en espacio respecto del rango de posibles valores en la entrada.

Algoritmos para Streaming

$$F_k = \sum_{i=1}^{n} m_i^k \tag{1}$$

- Algoritmo de Morris [Mor78]: F_1
- Algoritmo de Flajolet-Martin [FM85]: F_0
- Estimación de Momentos de Frecuencia [AMS96]: $F_k,\ k \in N^*$

Algoritmo de Flajolet-Martin

Figura 2

Estrategias de Sumarización

Estrategias de Sumarización

- Muestreo Aleatorio
- Histogramas
- Wavelets
- Sketches

Sketches

- Bloom Filter [Blo70]
- Count-Min Sketch [CM05]
- Count Sketch [CCFC02]
- AMS Sketch [AMS96]
- Hyper-LogLog [FFGM07]
- L_p-Samplers [JST11]

Count-Min Sketch

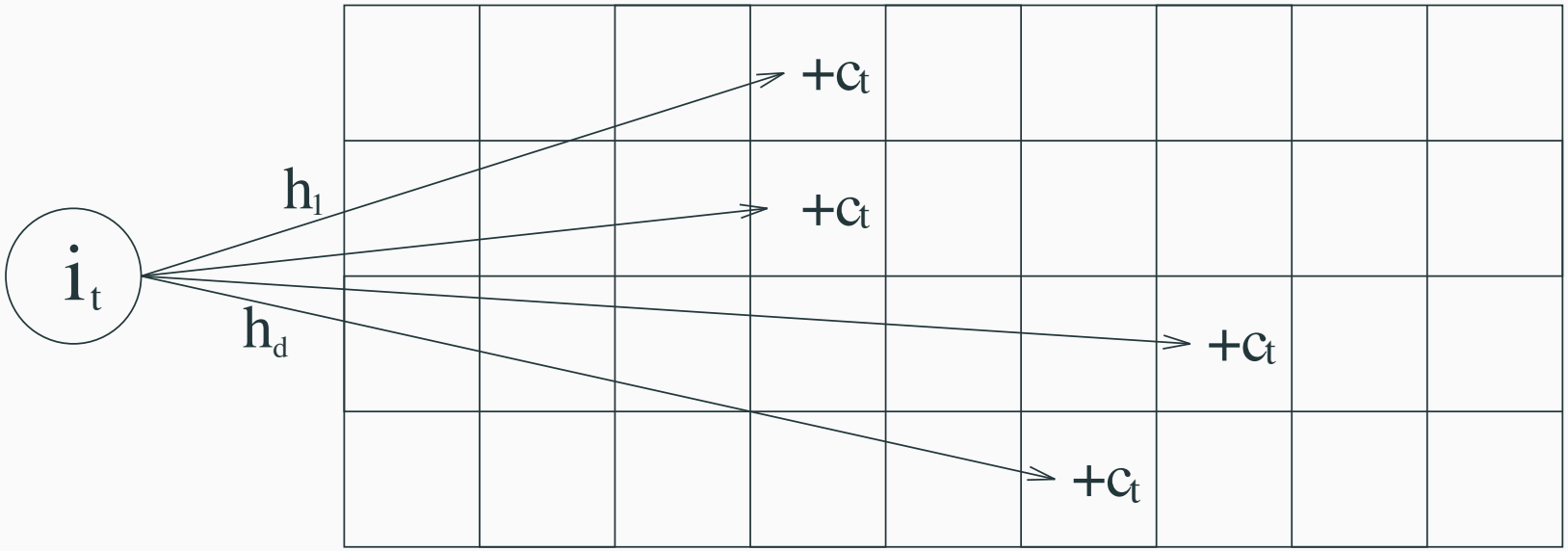


Figura 3

Algoritmos aplicados a Grafos

Algoritmos aplicados a Grafos

Sea $G = (V, E)$ un grafo formado por $n = |V|$ vértices y $m = |E|$ aristas, de tal manera que $e_i = (v_{i_1}, v_{i_2}) \in E$ y $\{v_{i_1}, v_{i_2}\} \in V$

Sobre el **Modelo en Semi-Streaming** se procesa un grafo a través del stream de aristas, en un espacio poli-logarítmico respecto del cardinal de vértices utilizando un número reducido de pasadas sobre el stream.

Spanners y Sparsifiers

- α-Spanner:

$$\forall v_{i_1}, v_{i_2} \in V,\ d_G(v_{i_1}, v_{i_2}) \leq d_H(v_{i_1}, v_{i_2}) \leq \alpha \cdot d_G(v_{i_1}, v_{i_2}) \quad (2)$$

- $(1+\epsilon)$-Sparsifier:

$$\forall A \in V\ (1-\epsilon)\lambda_A(G) \leq \lambda_A(H) \leq (1+\epsilon)\lambda_A(G) \quad (3)$$

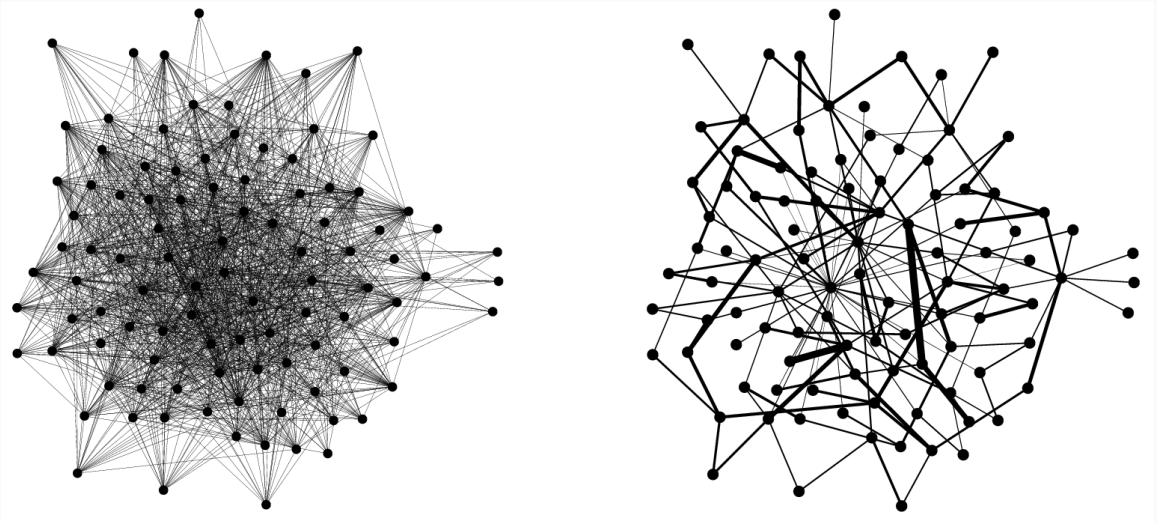


Figura 4

Problemas sobre Grafos

- Verificación de Grafo Bipartito [FKM+05]
- Conteo de Triángulos [BYKS02]
- Árbol Recubridor Mínimo [AGM12]
- Componentes Conectados [AGM12]
- Paseos Aleatorios [SGP11]
- ...

Algoritmo PageRank

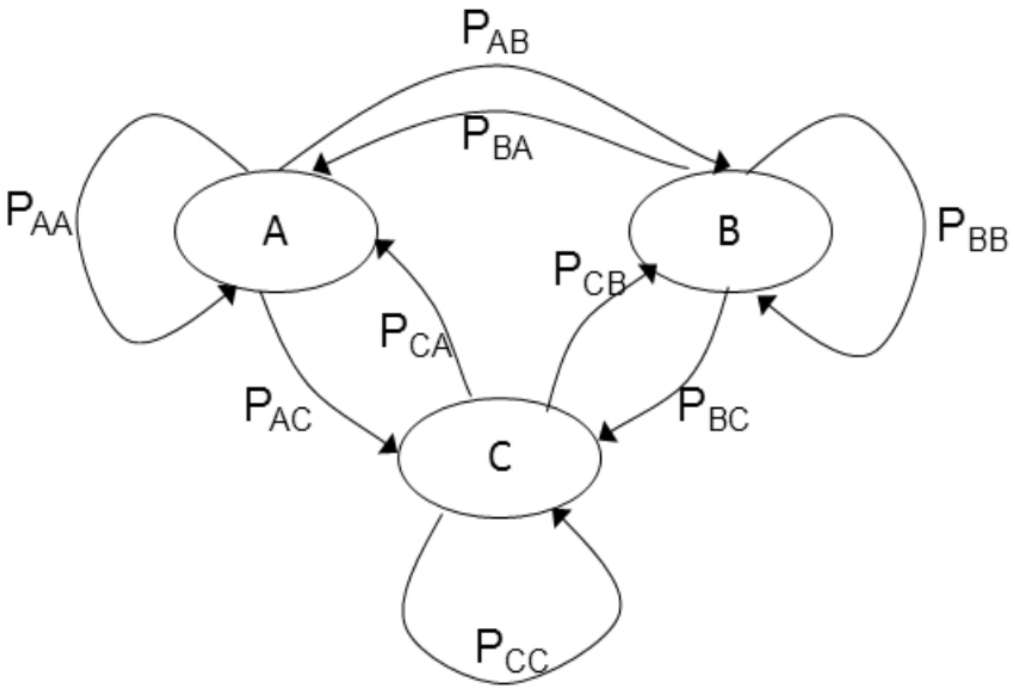
Algoritmo PageRank

Ranking de los vértices de un grafo basado únicamente en la estructura de relaciones generada por las aristas del grafo.

Los vértices incidentes con vértices populares tendrán mayor popularidad.

Distribución estacionaria de la **Cadena de Markov** subyacente.

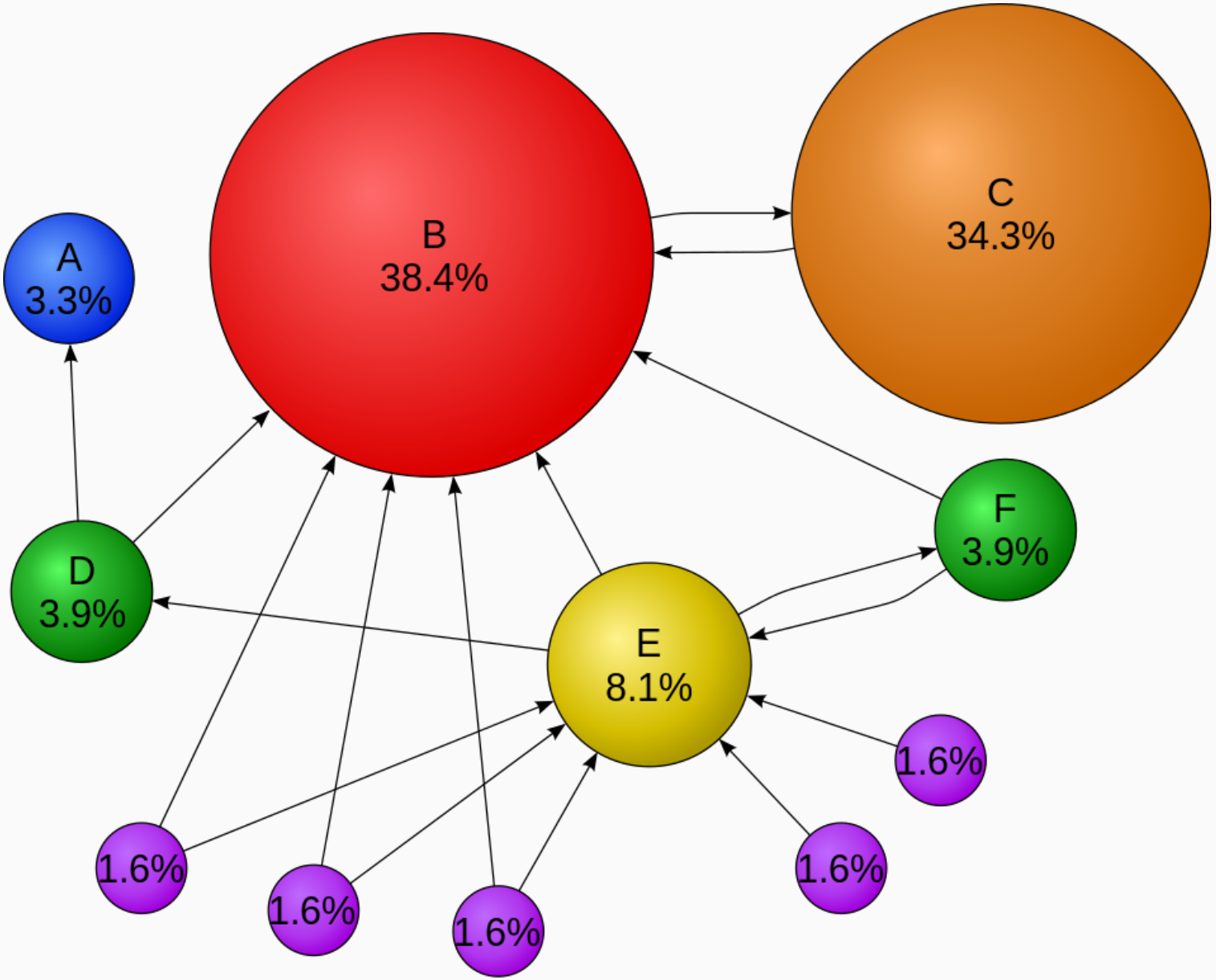
Cadena de Markov



$$\mathbf{P}=\begin{bmatrix} p_{AA} & p_{AB} & p_{AC} \\ p_{BA} & p_{BB} & p_{BC} \\ p_{CA} & p_{CB} & p_{CC} \end{bmatrix}$$

Figura 5

PageRank



PageRank sobre grafos reales

Vértices Sumidero no cumplen la propiedad de Cadena de Markov.

Saltos Aleatorios como solución al problema de los vértices sumidero, siguiendo una distribución uniforme.

$$T'_{ij} = \begin{cases} \beta * \frac{A_{ij}}{d^-(i)} + (1-\beta) * p_i & \text{if } d^-(i) \neq 0 \\ p_i & \text{otherwise} \end{cases} \quad \forall i, j \in [1, n] \tag{4}$$

¿Cómo calcularlo?

- Algebraica:

$$\pi = \left(\boldsymbol{I} - \beta * \frac{A}{d^{-}} \right)^{-1} * (1 - \beta) * p \tag{5}$$

- Iterativa:

$$\pi(t) = \pi(t-1) * T' \tag{6}$$

- Paseos Aleatorios

Algoritmos Similares

- HITS [Kle99]
- SALSA [LM01]
- SimRank [JW02]

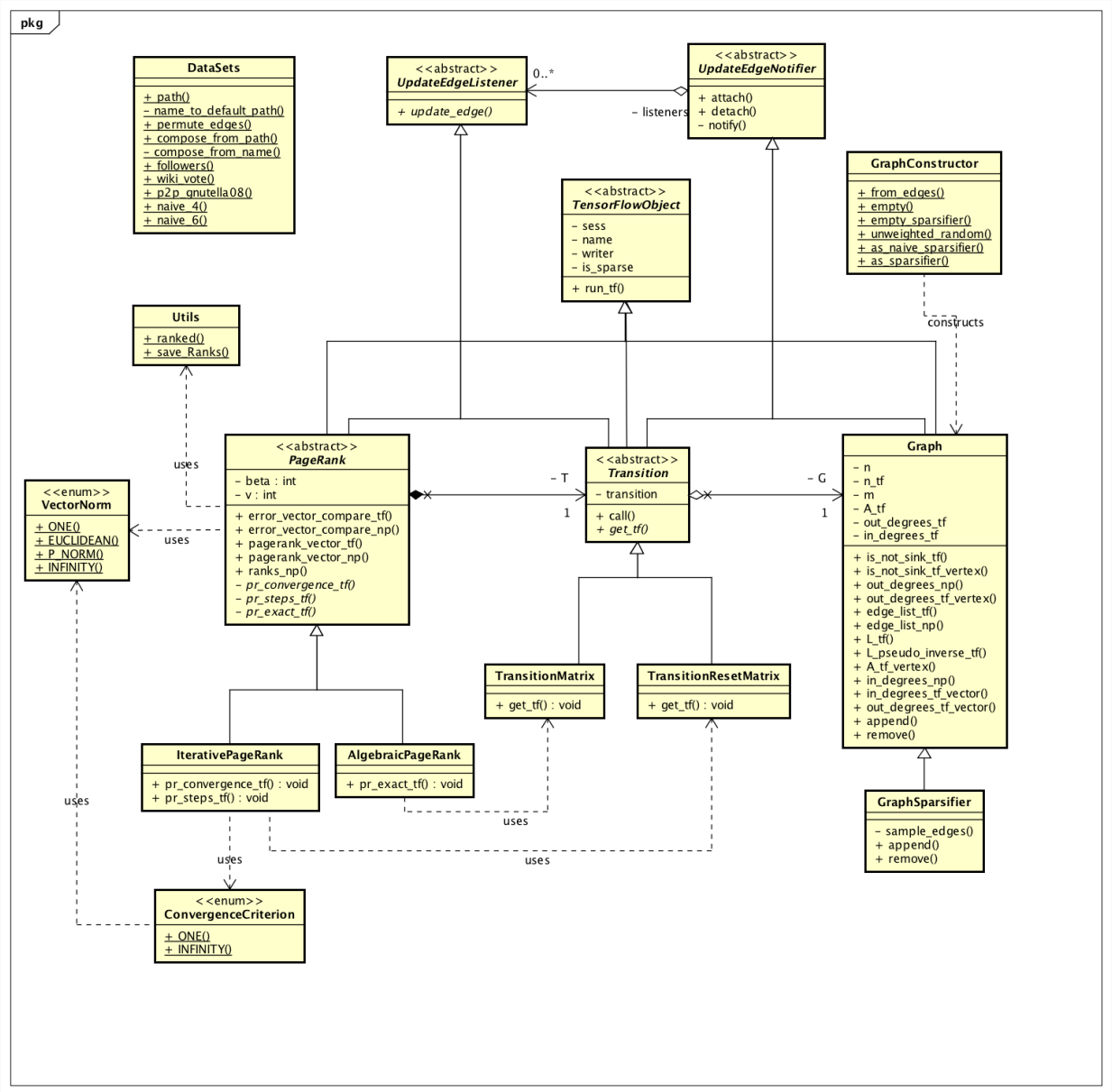
Implementación

Implementación

La implementación realizada consiste en una biblioteca de grafos (**tf_G**) utilizando como base la plataforma de cálculo matemático intesivo **TensorFlow** [AAB+16].

tf_G se encuentra en una fase muy temprana, formando únicamente el conjunto de métodos para el cálculo del **PageRank**.

Diagrama de Clases



Trabajo completo en:

https://github.com/garciparedes/tf_G

¿Preguntas?

Referencias i

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al.
Tensorflow: Large-scale machine learning on heterogeneous distributed systems.
arXiv preprint arXiv:1603.04467, 2016.

Kook Jin Ahn, Sudipto Guha, and Andrew McGregor.
Analyzing graph structure via linear measurements.
In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 459–467. Society for Industrial and Applied Mathematics, 2012.

Referencias ii

Noga Alon, Yossi Matias, and Mario Szegedy.
The space complexity of approximating the frequency moments.
In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

Burton H Bloom.
Space/time trade-offs in hash coding with allowable errors.
Communications of the ACM, 13(7):422–426, 1970.

Referencias iii

Ziv Bar-Yossef, Ravi Kumar, and D Sivakumar. **Reductions in streaming algorithms, with an application to counting triangles in graphs.** In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 623–632. Society for Industrial and Applied Mathematics, 2002.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. **Finding frequent items in data streams.** In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

Referencias iv

Graham Cormode and Shan Muthukrishnan.
An improved data stream summary: the count-min sketch and its applications.
Journal of Algorithms, 55(1):58–75, 2005.

Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier.
Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm.
In *Analysis of Algorithms 2007 (AofA07)*, pages 127–146, 2007.

Referencias v

Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang.
On graph problems in a semi-streaming model.
Theoretical Computer Science, 348(2-3):207–216, 2005.

Philippe Flajolet and G Nigel Martin.
Probabilistic counting algorithms for data base applications.
Journal of computer and system sciences, 31(2):182–209, 1985.

Referencias vi


Hossein Jowhari, Mert Sağlam, and Gábor Tardos.
Tight bounds for lp samplers, finding duplicates in streams, and related problems.
In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 49–58. ACM, 2011.


Glen Jeh and Jennifer Widom.
Simrank: a measure of structural-context similarity.
In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

Referencias vii


Jon M Kleinberg.
Authoritative sources in a hyperlinked environment.
Journal of the ACM (JACM), 46(5):604–632, 1999.


Ronny Lempel and Shlomo Moran.
Salsa: the stochastic approach for link-structure analysis.
ACM Transactions on Information Systems (TOIS), 19(2):131–160, 2001.


Robert Morris.
Counting large numbers of events in small registers.
Communications of the ACM, 21(10):840–842, 1978.

Referencias viii

Atish Das Sarma, Sreenivas Gollapudi, and Rina Panigrahy.
Estimating pagerank on graph streams.
Journal of the ACM (JACM), 58(3):13, 2011.