Drakkar: a graph based All-Nearest Neighbour search algorithm for bibliographic coupling

Bart Thijs

KU Leuven, FEB, ECOOM; Leuven; Belgium
Bart.thijs@kuleuven.be

Abstract

Drakkar is a novel algorithm for the creation of bibliographic coupling graphs in huge document spaces. The algorithm approaches this as an All-Nearest Neighbour search problem and starts from a bipartite graph constituted by the citing publications and the cited references and the directed citations connecting them. The approach is inspired by dimensionality reduction techniques like Random Projection and Locality Sensitive Hashing which use global random functions for dimension or feature selection. The proposed algorithm enables the use of local selection functions at the level of the individual nodes. For the particular case of bibliographic coupling the selection functions are based on the boat-shaped information distribution associated with the indegree of the cited references. This distribution resembles the typical symmetrical shape of a Viking ship (called 'Drakkar' in Dutch, hence the name). An experiment with several different random functions reveals that focussing on the end of the distribution related to the references with low indegree results in a graph with accurate strong links but many false negative while the other end of the distribution can detect most links but underestimates the strength of the link. The algorithm is implemented in GraphX, the library for distributed graph processing within Spark. It is using Pregel's messaging framework.

Keywords: Nearest Neighbour Search, Bibliographic Coupling, GraphX, Pregel, Bulk Synchronous Parallel.

Introduction

An important challenge for large scale application of bibliographic coupling (BC) in global clustering exercises or large domain studies is the computational and storage resources required for the creation of such BC-networks. Depending on the chosen representation of the underlying data, different strategies for the calculation of the cosine similarities can be applied.

In a relational database one could store citing publication-citing references pairs and use a query that joins such a table with itself on the cited references. An aggregate function can then count the number of joint references for each publication-publication pair.

Alternatively, data can be stored in a (sparse) matrix representation of a document-feature space where the cosine similarity is based on the dot product of this matrix and its transposed. But without any optimisation this calculation would take up to $O(n^2m)$-time for n documents and m features (cited references). Using dimensionality reduction tech-

niques like PCA or SVD could reduce the computational complexity by lowering the m-factor but not without an additional cost as these techniques imply a matrix decomposition which would require substantial computation time even when using an iterative implementation. Based on the Johnson–Lindenstrauss lemma [1], Random Project reduces the high dimensional space to a subspace with much lower features while preserving the distance between documents. However, such a dimensionality reduction does not eliminate the n-by-n document comparison and implies new projections whenever new documents that extend the feature space are added.

The first problem of the n-by-n comparison can successfully be solved by the application of Locality Sensitive Hashing which is a common technique applied in record linkage problems (eg. [2]). LSH uses several random hashing functions for mapping with high probability documents with a great similarity into the same buckets (see [3] or [4]). Documents that often co-occur in these buckets have a high likelihood to be similar and the number of pairwise cosine calculation can thus be drastically reduced by limiting it to those document pair with high co-occurrence. The cosine similarity can be approximated based on the number of co-occurrences in buckets.

The application of LSH for bibliographic coupling comes with two main drawbacks that have some substantial consequences on its applicability. LSH does not solve the issues that are confronted when extending the document-feature space. New hashing functions have to be created and all documents have to be assigned to new buckets in order to be able to calculate the similarity with documents in the prior set and the newly added ones. The second drawback is related to the existence of false positives. Given the extremely sparse nature of bibliographic coupling it is quite likely that the set of hashing functions only selects those features that are absent in a large set of papers which do not share any reference. Consequently, these papers are all assigned to the same buckets despite the distance between them. This can only be solved by increasing the number of hashing functions, by increasing the dimensionality of the functions or by avoiding the approximation of the cosine similarity by actual calculating this value. Each of these solutions come with a substantial computational cost.

This paper takes an alternative approach by exploiting the properties of a graph representation of the underlying citation data. The creation of a bibliographic coupled network can be considered as an all-nearest neighbour search (ANNS) problem in a huge feature space represented as a bipartite graph.

Message passing

The proposed algorithm is based on the Pregel messaging framework developed by Malewicz [5] at Google. This framework builds on the Bulk Synchronous Parallel model [6] by implementing a sequence of supersteps. These supersteps start with the parallel calculation of vertex properties either based on existing properties of the vertex or based on the incoming messages from the previous superstep. In a second step, messages are sent to neighbouring vertices containing calculated properties. The last step in the superstep is the aggregation of the incoming messages at the receiving vertices. Typical Pregel based programs run an iteration of a superstep until some prior defined stopping criterion is reached.

Given the bipartite nature of the graph underlying our bibliographic coupling ANNS problem it is impossible to apply an iteration of a single superstep multiple times. Therefore, this algorithm consists of three distinct supersteps.

Superstep I.

Step 1. Publication and references calculate their degree, thus the number of outgoing or incoming edges or links.

Step 2. Each publication sends a message containing its identifier and out-degree to cited reference across all the outgoing edges. (Dashed line in figure 1)

Step 3. Each reference collects the received messages into an ordered list.

Superstep II.

Step 1. Each reference decides if it will send out the list and to which of the citing publications. If a reference decides not to send it becomes inactive

Step 2. Each active reference sends messages across incoming links. Each message contains a list with the identifiers and properties of those publications that appear after the identifier of the recipient in the ordered list. (Dotted line in figure 1)

Step 3. Each publication collects the incoming messages

Superstep III.

Step 1. Each publication calculates the occurrence of each identifier in the joined set of messages. A Salton cosine similarity is now calculated based on the number of joint references, the out-degree of the current publication and the out-degree of the other publication being part of the message.

Step 2. Each publication can now send a message to its bibliographically coupled neighbour without actual edges being present. (Dash-dotted line in figure 1)

Step 3. Publications receive incoming messages and weighted edges are created and no further calculations are needed.

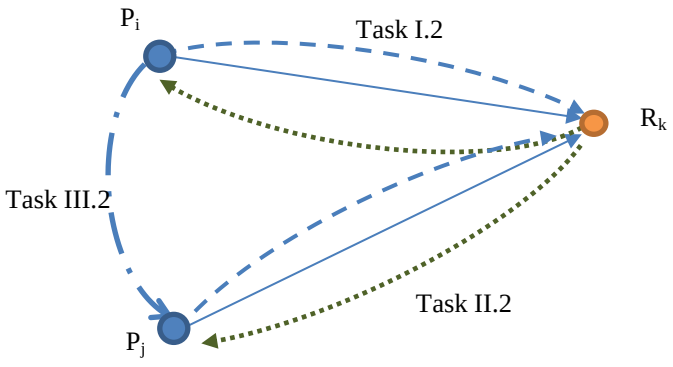

Figure 1. Schematic overview of message passing between two publications and one joint reference.

Several advantages are associated to this approach. Steps are performed in a sequential order and results are stored. Tasks within each step are suitable for distributed execution as they run independent from each other. References sending out messages in step II.2 rely solely on the information already gathered by each individual reference. Consequently, each task can be performed in parallel.

But most important for this algorithm is the ability that this framework provides to define any function to be applied at the individual reference for the selection of publications receiving messages with the identifiers of their neighbour publications. This selection function could be completely randomized and thus be analogous to the selection of dimensions in a LSH procedure. But it also allows for more complex functions either deterministic or probabilistic for the selection of active references. In a state where the individual references have no or very limited information about the actual topology of the graph, it is the in-degree of each reference that is the most obvious parameter for the selecting function.

Experimental setup

For a valid testing of the different selection scenarios, I use the amount of information that is being sent after the application of the selection function in step 2. This amount can be calculated based on the indegree of the reference and the total distribution of indegrees across the network. The next section introduces the required definitions and formulas for the calculation of the amount of passed information.

At first, a bipartite graph G is defined by the sets of publications *P*, cited references *R* and edges *E*, with *p*, *r* and *e* their respective cardinality.

$$G = (P, R, E)$$
$$p = |P| \; and \; r = |R| \; and \; e = |E|$$

The number of outgoing and incoming links is calculated as the out- and indegree of publication and reference.

$$outdeg_i = outdegree \; of \; publication \; P_i \in P$$
$$indeg_j = indegree \; of \; reference \; R_j \in R$$
$$\sum_{i=1}^{p} outdeg_i = \sum_{j=1}^{r} indeg_j = e$$

The indegree for the references in the graph ranges from 1 to some highest value n_{max}. References not cited by any publication are not included in the graph. Each reference can be assigned to a set of references with the same indegree.

$$n = 1..n_{max} : \; \forall R_j \in R: 1 \leq indeg_j \leq n_{max} \; \wedge \exists R_j \in R: indeg_j = n_{max}$$
$$\forall R_j \in R: \; R_j \; \in D_{(n)} \Leftrightarrow indeg_j = n$$

The amount of information to be sent by all the references in a set of same indegree *n* is equal to the product of cardinality of this set and the number of possible 2-

combinations in a set of size *n*. One unit of information is the pair of the identifier of the citing publication and its outdegree as it is send out at step I.2. References with a degree of 1 will not send out any information as it is not possible to make any 2-combination in a set of size 1

$$I_{(n)} = \left|D_{(n)}\right| \frac{n(n-1)}{2}$$
$$I_{(1)} = 0$$

The total information in the publication-reference network is equal to the sum of information over each of the indegree sets.

$$I_{tot} = \sum_{n=2}^{n_{max}} I_{(n)}$$

It is not only possible to calculate the total amount of transmitted information but also to sum over a range of indegree values.

$$I_{(n..m)} = \sum_{i=n}^{m} I_{(i)}$$

The range can be chosen with such boundaries that it accounts for a given share of the total information. The upper bound for the range of indegrees accounting for up to 25% of the total information can be defined as follows:

$$n = n_{25\%} \Leftrightarrow \left|D_{(n)}\right| > 0 \ \wedge \forall m < n_{25\%}: \frac{I_{(2..m)}}{I_{tot}} \leq \frac{I_{(2..n_{25\%})}}{I_{tot}} \leq 0.25$$

The definition of these ranges associated with some share of information provides the mechanism to choose different testing scenarios with equal amount of information contained in the messages being transferred from reference back to publications. These ranges can not only be taken from the lowest end of the indegree distribution, but also from the top, in the middle or a combination of bottom and top end.

As table 1 shows, a combination of these four types with four different levels of shares of information to be transmitted defines the first sixteen scenarios to be tested. The table 1 specifies the indegree ranges. This approach defines deterministic binary functions solely based on the indegree and the relevant range. References do or do not send their compiled list to each of their citing publications;

Table 1. Indegree ranges used for the specified share of transmitted information

	Bottom	Middle	Top	Bottom + Top
20%	$2..n_{20\%}$	$n_{40\%}..n_{60\%}$	$n_{80\%}..n_{max}$	$2..n_{10\%}$ or $n_{90\%}..n_{max}$
40%	$2..n_{40\%}$	$n_{30\%}..n_{70\%}$	$n_{60\%}..n_{max}$	$2..n_{20\%}$ or $n_{80\%}..n_{max}$
50%	$2..n_{50\%}$	$n_{25\%}..n_{75\%}$	$n_{50\%}..n_{max}$	$2..n_{25\%}$ or $n_{75\%}..n_{max}$
66%	$2..n_{66\%}$	$n_{17\%}..n_{83\%}$	$n_{34\%}..n_{max}$	$2..n_{33\%}$ or $n_{67\%}..n_{max}$

However, it is also possible to define probabilistic functions. The first four probabilistic scenarios apply a simple random function to each message to be transmitted. The probability to be transmitted is equal to the given share of information and independent of the indegree at the level of the cited reference and independent of the size of the compiled

list to be sent. Analogous to the deterministic functions the shares are set to 25%, 40%, 50% and 66%

A last series of four 'tailed' scenarios combines a deterministic upper limit threshold with a probabilistic function for those references where the indegree exceeds the threshold. This means that these references randomly select a limited set of citing publications that receive the compiled list. The probability to be selected can be defined as

$$P = \frac{k}{n}$$

where *n* is the indegree of the reference and *k* is equal to

$$k = \begin{cases} n \mid n \leq l \\ \left\lceil \frac{l(l-1)}{2(n-1)} \right\rceil \mid n > l \end{cases}$$

with l being the threshold.

The amount of information to be sent by a set of references with the same indegree *n* can be calculated by substitution of *n* by *k*

$$I_{(n|l)} = \left|D_{(n)}\right| \frac{k(n-1)}{2}$$

and the amount of information transmitted in the network with a given threshold l is then the sum over all the indegree values in the

$$I_{tot|l} = \sum_{n=2}^{n_{max}} I_{(n|l)}$$

These definitions allow us to set the threshold to such a value that only a given share of information is used for the creation of the bibliographic coupling networks. In line with all the previous scenarios, thresholds are set to create tailed scenarios accounting for 20%, 40% 50% and 66% of the total amount of information.

The results from these scenarios are gauged against the original bibliographic network using all the publication-reference links.

Data source and processing

1.39 million Publications of type Article or Review indexed in the 2013 volume of Clarivate Analytics Web of Science (WoS) were used. In WoS, references in these publications get a specific R9-code. References to the same cited work in different publications are labelled with the same R9-code. Consequently, a co-occurrence of R9 –codes in the reference lists of two publications indicates a bibliographic coupling. Both publications and cited references are considered to be nodes in a large network. The reference to a cited document is recorded as a directed edge in the bipartite network. The final dataset

consists of pairs of identifiers where the first refers to the citing publication and the second to the cited reference.

Table 2. Descriptive statistics for the bi-partite network

Number of publications	p	1,391,192
Number of cited references	r	17,248,290
Number of publication-reference pairs	e	49,156,442
Average number of references per publication	p/e	35.34
Average number of citations to references	e/r	2.85

[Data sourced from Clarivate Analytics Web of Science Core]

The processing is done using the Elastic MapReduce service offered by Amazon in their AWS Cloud Compute environment. Several Hadoop clusters running Spark with one master and from five up to ten memory optimized worker instances were created. The bipartite network is processed by using the GraphX library which is the graph computation API within Apache's Spark. This library provides the required methods for the development of a bulk-synchronous messaging system. *mapVertices* and *joinVertices* are the two methods that can be used in the first (calculation) task in each superstep. The *aggregateMessage* method combines second and third task which passes relevant information across existing edges and combines all the incoming messages using the provided function.

Results

The analysis started with the calculation of distribution of the indegree and the amount of information to be transmitted at step II.2 associated with each value of indegree found in the graph. As mentioned before, the unit of information to be sent is the pair of identifier and outdegree of the citing publication (a pair of a *Long* and *Int* values in the Spark implementation). Figure 2 plots this amount of information in a logarithmic scale for the obtained indegree values. The highest indegree value found was 5927 and occurred only once. The horizontal axis is not truly interval scaled but merely ordinal as only those indegree values that occur in the dataset are included. Consequently, the figure shows a steep increase of information near the end of the distribution for those values that are only observed once.

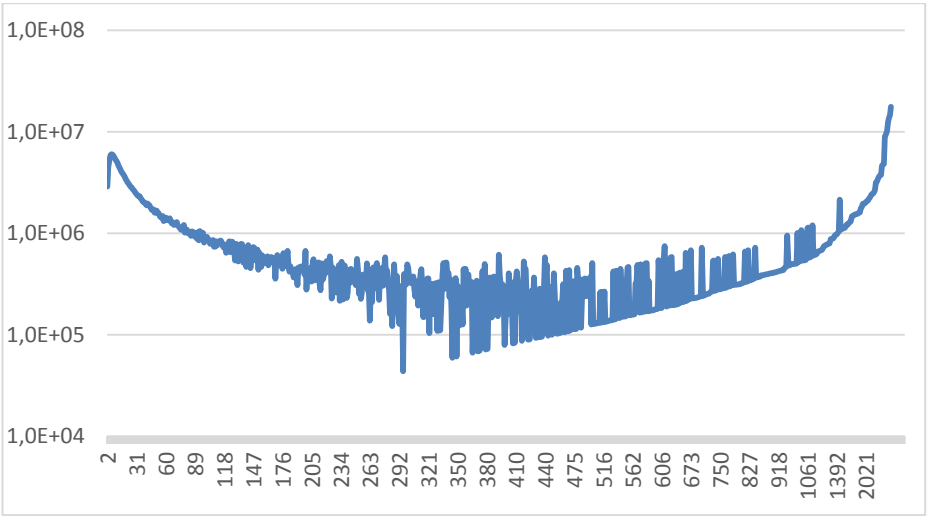

Figure 2. Amount of information to be transmitted by each value of observed indegree (x-axis: observed indegree; y-axis: amount of information)
[Data sourced from Clarivate Analytics Web of Science Core Collection]

The particular shape of the figure resembles the typical design of Viking ships (drakkar called in Dutch, hence the name of the algorithm) with symmetrical ends and justifies the selection of a given amount of information from both sides of the distribution. The thresholds of the indegrees are given in table 3 and allow the creation of the intervals required for the definition of the sixteen scenarios as presented in table 1. In the tailed scenarios, the thresholds are respectively set to 20, 93, 193 and 600 to obtain the same shares.

Table 3. Upper thresholds for the selection of the associated amount of information.

	threshold			threshold
$n_{10\%}$	13		$n_{60\%}$	657
$n_{17\%}$	26		$n_{66\%}$	1026
$n_{20\%}$	35		$n_{70\%}$	1260
$n_{25\%}$	53		$n_{75\%}$	1760
$n_{30\%}$	75		$n_{80\%}$	2165
$n_{34\%}$	98		$n_{83\%}$	2725
$n_{40\%}$	158		$n_{90\%}$	4333
$n_{50\%}$	326			

[Data sourced from Clarivate Analytics Web of Science Core Collection]

The first test measures the recall of each scenario. This is calculated by comparing the final number of bibliographic coupling links of each scenario with the selected share of information with the number of bibliographic coupling links when using the complete citation graph. Using all the information present in the original publication-reference network of all 2013 publications resulted in 496 million weighted links between publications. This recall could also be rephrased as the ratio between the density of the biblio-

graphic network after the application of the selection function and the density of the BC network without any selection. The density of the latter network is about 0.05%. Table 4 presents these recall values for each of the twenty-four versions. The columns present the amount of information that is transmitted and the rows refer to the different scenarios being applied for the selection function.

The first observation is that when using a pure random function the recall is almost the same as the selected information share. This can be observed in the sixth row. Next, the recall of the two scenarios that focus on those references with a low indegree ('Bottom' and 'Tailed') is below the value set by the pure random selection and the share of used information. The highest density is obtained when choosing those references with the highest indegree to build the bibliographic coupling network.

The scenarios where either the references located at the centre of the information distributions or at the outer bounds are selected still perform better than the random scenarios. The largest difference between top and bottom can be observed when half the information is used. Cutting the set of references into two subsets associated with an equal amount of information results in a recall of 46.8% for the lower end compared to 57.6% for the upper end. But relatively, when only using 20% of the available information, the use of the most cited references results in an BC-network with a 30% higher recall than based on the least cited references. Based on these observations, it would be justified to say the selection of the top references is a better approach.

Table 4. Recall of each scenario with the associated amount of selected information

	20%	40%	50%	66%
Bottom	18.5%	36.8%	46.8%	64.2%
Between	23.4%	44.8%	54.4%	70.1%
Top	24.2%	45.9%	56.7%	73.8%
Bottom & Top	22.9%	42.5%	52.1%	68.8%
Tailed	17.9%	36.1%	46.0%	63.3%
Random	20.1%	40.0%	50.0%	66.0%

[Data sourced from Clarivate Analytics Web of Science Core Collection]

It should be noted that this messaging based algorithm does not result in false positive BC-links between two publications as messages are only passed along truly existing citation links. When using a binary approach, the precision would then be equal to 100%. However, as Bibliographic Coupling results in a weighted network, we can measure the ability of each scenario to approximate the actual strength or weight of the link. It is in step III.1 that the cosine similarity between publications is calculated. Given the fact that false positives are absent and only false negatives can occur, the weight of the link can never be overestimated. The effect of a selection of different scenarios on the distribution of link weights can be seen in figure 3.

The top line refers to the BC-network without any selection function. None of the other scenarios has a distribution that surpasses this at any weight value. Two clear phenomena can be observed from this graph. First those scenarios that do not include systematically those references with a low indegree underestimate the strong links. The top scenar-

io with 40% of the available information is at the bottom of the graph from a weight of at least 0.1. But also the scenario which selects the references in the middle of the information distribution performs lower. And a pure random function is not much better. But those scenarios that focus on the references with low indegree approximate the distribution of strong links. As expected, adding more information to these scenarios improves the results slightly at the upper end of the weight distribution.

The second observation is that the scenarios selecting the bottom references fail primarily in detecting the lower weighted links. It is in this area that the explanation can be found for the results presented in table 4 with respect to the lower recall of those scenarios.

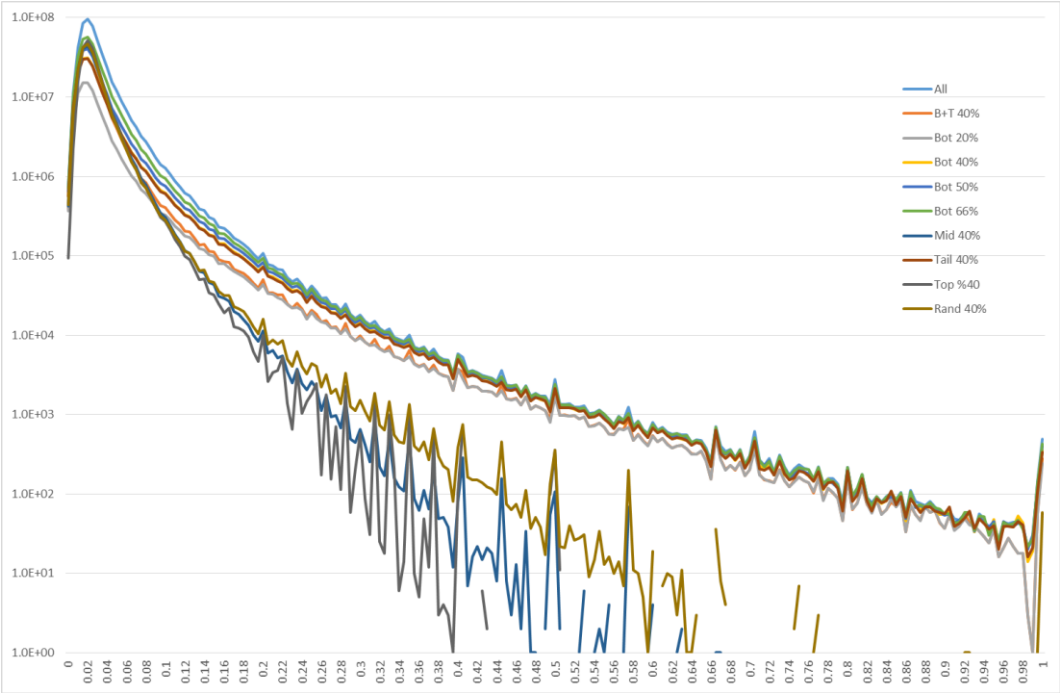

Figure 3. Comparison of distribution of weighted links across different scenarios.
(x-axis: Strength of BC-link; y-axis: count of observed links)
[Data sourced from Clarivate Analytics Web of Science Core Collection]

It seems that closely related documents share references to poorly cited documents while highly cited documents receive citations from a broader range of papers covering multiple topics. These observations have strong implications on the choice of selection function. When the objective of the creation of a large scale bibliographic coupling network in a computationally constrained environment is to find pairs of closely related paper than the selection function should focus on the lesser cited references. Opposed to this, the objective could also be the clustering of the complete network in which case the selection functions can be restricted to the upper end of the indegree distribution.

Conclusions

The application and use of large scale bibliographic coupling networks has been hindered by the computational and storage resources required for the creation of these networks. Alternative networks based on direct citations have been used in large scale analysis. The new graph messaging algorithm proposed in this paper provides an opportunity to produce the large scale networks through the application of different selection functions at the level of individual cited references. The experiments with different functions show that references at the lower or higher end of the indegree distribution play a different role in the citation network. Focussing on the bottom results in a network that approximates most of the strong links but is more likely to ignore the weaker ones. Shifting the focus to the other end creates the inverse effect: a higher recall but worse for the identification of strong links. The choice for a particular set of selection function thus depends on the actual objectives for the creation of these BC-networks. If global clustering is the goal then the upper end of the distribution is the right path while if the objective is only to delineate a set of documents closest related to a particular sample the lower end of the indegree is most relevant. Future research will investigate the applicability of this graph based nearest neighbour search algorithm for lexical similarity between scientific documents.

References

1. Johnson, W. B. & Lindenstrauss, J., (1984). Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics. 26, 189–20
2. Karapiperis , D. & Verykios, V.S. (2016). A fast and efficient Hamming LSH-based scheme for accurate linkage. Knowledge and Information Systems, 49 (3), 861-884.
3. Rajaraman, A. & Ullman, J. (2010). "Mining of Massive Datasets, Ch. 3." URL: http://infolab.stanford.edu/~ullman/mmds.html
4. Ravichandran. D., Pantel. P. & Hovy. E. (2005). Randomized algorithms and nlp: using locality sensitive hash function for high speed noun clustering. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 622-629.
5. Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, I, Leiser, N. & Czajkowski, G. (2010). Pregel: a system for large-scale graph processing. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 135-146
6. Valiant, L.G. (1990). A bridging model for parallel computation, Communications of the ACM, 33 (8), 103-111.