Datamining algorithms

Apriori algorithm and SON algorithm (Association rule algorithm)
Given an item set $I = \{I_1, I_2, \ldots, I_M\}$, a transaction set $T = \{T_1, T_2, \ldots, T_N\}$ each of which is a subset of items, so that $T_n \subset I$, an association rule $X \rightarrow Y$ is then valid when conditional probability of Y given X exceeds a threshold :

$$\Pr(Y \mid X) > threshold$$

where $X, Y \subset I$, and $X \cap Y = \varnothing$. We can extract all the association rules using brute force, the total number of combinations are :

$$M \times (M-1) \times 3^{M-2}$$

which means picking one item into X, picking another item into Y, each of the rest items either belongs to X, Y or $complement(X \cup Y)$. Apriori algorithm is a systematic and slightly more efficient approach than brute force, which extracts all association rules, by early elimination of invalid combinations with the following property :

$$\Pr(YZ \mid X) < P(Y \mid X)$$

Therefore there is no need to validate rule $X \rightarrow Y,Z$ if $X \rightarrow Y$ is invalid. However Apriori algorithm is inefficient when M is large (*10?*). SON algorithm is simply a concurrent version of apriori algorithm. Its major contribution is to reformulate apriori as map reduce.

Pagerank
Pagerank of a page is defined as the weighted average number of reference links to the page from the whole web. Let $PR(x_n)$ be the page rank of a page x_n, then we have :

$$PR(x_n) = \sum_{\substack{x_m \in www \\ NL(x_m) > 0}} PR(x_m) / NL(x_m) \times d + (1-d)$$

where d is damping factor, which is typically *0.85*, $NL(x_m)$ is the number of outward links from page x_m. If there are only N pages in the whole www, all the N pageranks form a system of N linear equations in N unknowns, which must have one solution, unless the system becomes degenerated when damping factor is set to be *1*. When an appropriate damping factor is used, we shouldnt find all pageranks simply by matrix inverse of the system, it is inefficient as the web is enormous, therefore we need an iterative approach : initialize all pageranks as *1*, randomly sample a page and update its pagerank using the above equation, repeat until convergence.