Data Mining

Scritto del 10/07/2017
(SOLUZIONI degli ESERCIZI)

Problema. Sia D un dataset di N transazioni su un insieme I di item. Dato $X \subseteq I$ e detto T_X l'insieme delle transazioni che contengono X, abbiamo definito Closure$(X) = \cap_{t \in T_X} t$ e dimostrato che è un itemset chiuso con lo stesso supporto di X. Inoltre, se X è chiuso vale che Closure$(X) = X$.

1. Dimostrare che se X è chiuso e $Y \subset X$ (Y non necessariamente chiuso), allora Closure$(Y) \subseteq X$
2. Sia X l'itemset chiuso e non vuoto che ha supporto massimo tra tutti gli itemset chiusi e non vuoti. Usando il risultato del punto precedente, dimostrare che per ogni $a \in X$ is ha che Closure$(\{a\}) = X$.
3.

Soluzione.

1. Since $Y \subset X$ we have that $T_X \subseteq T_Y$, hence
$$\text{Closure}(Y) = \cap_{t \in T_Y} t \subseteq \cap_{t \in T_X} t = \text{Closure}(X) = X.$$
2. From the previous point, we know that for each $a \in X$ Closure$(\{a\}) \subseteq X$, hence Supp(Closure$(\{a\})) \geq$ Supp(X). Since Closure$(\{a\})$ is a closed itemset and X was chosen as the closed itemset of maximum support, we must have Supp(Closure$(\{a\})) =$ Supp(X). Therefore X and Closure$(\{a\})$ must coincide, otherwise Closure$(\{a\})$ would not be closed.

□

Problema. Siano $P = \{x_0, x_1, \ldots, x_{N-1}\}$ ed $S = \{y_0, y_1, \ldots, y_{K-1}\}$ due insiemi di punti in uno spazio metrico (M, d), dove $K \leq \sqrt{N}$. Per ogni $y \in S$ si definisca la distanza di y da P come $\min_{x \in P} d(y, x)$. Progettare un algoritmo MapReduce efficiente che calcola la distanza da P per ogni punto di S, determinando il numero di round, lo spazio locale e lo spazio aggregato richiesti dall'algoritmo. Inizialmente, si assuma P rappresentato dalle coppie (i, x_i), con $0 \leq i < N$, e S dalle coppie $(N + i, y_i)$, con $0 \leq i < K$. Per avere punteggio pieno l'algoritmo deve usare spazio locale $o(N)$ e spazio aggregato lineare in N.

Soluzione. The idea is the following. In the first round, we partition the points of P arbitrarily into $\sqrt{N}$ subsets of equal size, namely P_j with $0 \leq j < \sqrt{N}$, and replicate each point of S $\sqrt{N}$ times. Then, for every $0 \leq j < \sqrt{N}$ we gather together P_j and the entire set S and compute the distance $d_{j,y}$ of y from P_j, for every $y \in S$. In the second round, for each $y \in S$ independently we compute minimum distance among the $d_{j,y}$'s, for $0 \leq j < \sqrt{N}$. A detailed specification of the algorithm is as follows:

Round 1

- **Map phase.** Each pair (i, x_i) is mapped into the intermediate pair $(i \bmod \sqrt{N}, (i, x_i))$. The pair $(N + i, y_i)$ is mapped into the $\sqrt{N}$ intermediate pairs $(j, (N + i, y_i))$, for $0 \leq j < \sqrt{N}$.
- **Reduce phase.** For each $0 \leq j < \sqrt{N}$ independently, gather all intermediate pairs with key j. Note that these pairs represent a subset P_j of P (those of kind $(j, (i, x_i))$) and the entire set S. For every $y \in S$, compute the minimum distance $d_{j,y}$ between y and a point of P_j, and produce the pair $(y, d_{j,y})$.

Round 2

- **Map phase.** Identity
- **Reduce phase.** For each $y \in S$ independently, gather all pairs $(y, d_{j,y})$ (where $0 \leq j < \sqrt{N}$) and return the pair $(y, d_{\min})$ where $d_{\min} = \min_{0 \leq j < \sqrt{N}} d_{j,y}$)

- The number of rounds is 2.
- Since each P_j has size $\sqrt{N}$ and $K = |S| \leq \sqrt{N}$, the first round requires local space $\Theta\left(\sqrt{N}\right)$. Also, in the reduce phase of the second round $\sqrt{N}$ pairs $(y, d_{j,y})$ are gathered for each $y \in S$, hence the second round also requires local space $\Theta\left(\sqrt{N}\right)$. Therefore, $M_L = \Theta\left(\sqrt{N}\right) = o(N)$.
- For what concerns the aggregate space, note that points of P are not replicated, while each of the K points of S is replicated $\sqrt{N}$ times. Since $K \leq \sqrt{N}$ we have that $M_A = \Theta\left(N + K\sqrt{N}\right) = \Theta\left(N\right)$

□