18.434: Seminar in Theoretical Computer Science April 30, 2015

The Hungarian Algorithm for Weighted Bipartite Graphs

Alex Grinman *agrinman@mit.edu*

1 Motivation: The Assignment Problem

Suppose there are n trucks that each carry a different product and n possible stores, each willing to buy the n different products at different prices represented by matrix W. *The Assignment Problem*: how can we assign each truck x_i to go to store y_j, given that y_i offers to buy the products from x_i for $\$W_{ij}$ dollars, such that the combined profit is maximized over all possible assignments?

1. General problem: given $X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_n\}$, matrix W where $W_{ij} = weight(x_i, y_j)$ is the weight of assigning x_i to y_i, find the matching assigning each x_i to each y_j such that the total weight is *maximized*.
2. What is the naive algorithm: look through all $n!$ possible assignments, pick highest scoring.
3. Assumption: $\forall i, j \in \{1, \ldots, n\} : W_{ij} \geq 0$.
4. Can think of problem as a Complete Weighted Bipartite Graph $G = (V, E)$:
   - $V = X \cup Y$
   - $E = \{(x_i, y_j)\}_{x_i \in X, y_j \in Y}$.
   - $weight(x_i, y_j) = W_{ij}$
5. An assignment is a perfect matching: problem is reduced to finding the perfect matching with maximum weight.

2 Basis of the Hungarian Algorithm (Kuhn-Munkres)

2.1 Definitions

1. A *labeling* for graph $G = (V, E)$ is a function $l : V \to \mathbb{R}$, such that:
$$\forall (u, v) \in E : l(u) + l(v) \geq weight((u, v))$$
2. An *Equality Subgraph* is a subgraph $G_l = (V, E_l) \subseteq G = (V, E)$, fixed on a labeling l, such that
$$E_l = \{(u, v) \in E : l(u) + l(v) = weight((u, v))$$

2.2 The Kuhn-Munkres Theorem

Theorem 2.1. *Given labeling l, if M is a perfect matching on G_l, then M is maximal-weight matching of G.*

1. Let M' be any perfect matching in G. By definition of a labeling function and since M' is perfect,
$$weight(M') = \sum_{(u,v) \in M'} weight((u, v)) \leq \sum_{(u,v) \in M'} l(u) + l(v) = \sum_{v \in V} l(v)$$
2. This means: $\sum_{v \in V} l(v)$ is an upper bound for any perfect matching M' of G.
3. Now look at the weight of matching M:
$$weight(M) = \sum_{(u,v) \in M} weight((u, v)) = \sum_{(u,v) \in M} l(u) + l(v) = \sum_{v \in V} l(v)$$
4. By equation (1) and (3), we have that for all perfect matchings M' of G:
$$weight(M) \geq weight(M')$$

Key Point: By the Kuhn-Munkres Theorem the problem of finding a maximum weight assignment is reduced to finding the right labeling function and any perfect matching on the corresponding equality subgraph.

3 The Hungarian Algorithm

Algorithm Idea: maintain both a matching M and equality graph G_l, starting with $M = \emptyset$ and a valid l. Continue until M becomes a perfect matching on G_l. Each step: either augment M or improve the labeling $l \to l'$.

3.1 Augment the matching

Given labeling l, $G_l = (V, E_l)$, some matching M on G_l, unmatched $u \in V, u \notin M$.

1. A path is *augmenting* for M on G_l if it alternates between $E_l - M$ and M, and the first and last vertices of the path are un-matched in M. Keep track of an "almost" augmenting path starting at u
2. *If* we can find an unmatched vertex v, then we create augmenting path α from u to v.
3. Flip the matching by replacing the edges in M with the edges in the augmenting path that are in $E_l - M$.
4. Since we start and end unmatched, this increases the size of the matching $\implies |M'| > |M|$

3.2 Improve the labeling

1. $S \subseteq X$ and $T \subseteq Y$ s.t S, T represent the current "almost" augmenting alternating path between the matching M and outside other edges in $E_l - M$.
2. Let $N_l(S)$ be the neighbors to each node in S along E_l. $N_l(S) = \{v | \forall u \in S : (u, v) \in E_l\}$
3. if $N_l(S) = T$ we cannot increase the alternating path and augment, so we must improve the labeling!
4. Compute: $\delta_l = \min_{u \in S, v \notin T}\{l(u) + l(v) - weight((u, v))\}$
5. Improve $l \to l'$:
$$l'(r) = \begin{cases} l(r) - \delta_l & \text{if } r \in S \\ l(r) + \delta_l & \text{if } r \in T \\ l(r) & \text{otherwise.} \end{cases}$$
6. *Claim:* l' is a valid labeling and $E_l \subset E_{l'}$.
7. Proof follows by examining cases for all possibilities of membership of $u \in S, u \notin S, v \in T, v \notin T$.

3.3 The Hungarian Algorithm

1. Start with some matching M, and a valid labeling $l ::= \forall x \in X, y \in Y : l(y) = 0, l(x) = \max_{y' \in Y}(weight(x, y'))$
2. Do the following until M is a perfect matching:
   (a) Look for augmenting path
   (b) If augmenting path does not exist, improve $l \to l'$ and go to step (a).

3.4 Complexity

1. Each Step (a) or (b) increase the size of the matching by 1 edge each round, hence $O(|V| = 2n) = O(n)$ total rounds.
2. Augmenting M: $O(|V|)$ to find the right vertex if one exists, $O(|V|)$ to flip the matching
3. Improving the label: $O(|V|)$ to find δ_l and update the labeling. Improving the labeling can occur $O(|V|)$ times if no augmenting path is found. So total $O(|V|^2)$ in a single round.
4. $O(|V|)$ rounds with work $O(|V|^2)$, total running time is $O(|V|^3) = O(n^3)$.

References

[1] Wikipedia. "Hungarian Algorithm" `http://en.wikipedia.org/wiki/Hungarian_algorithm`.

[2] Dr. Derek Bruff. Department of Mathematics, Harvard University. "The Assignment Problem and the Hungarian Method" `http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf`.

[3] Mordecai Golin. Dept. of Computer Science, Hong Kong UST. "Bipartite Matching & the Hungarian Method" `http://www.cse.ust.hk/~golin/COMP572/Notes/Matching.pdf`.