Hungarian Method

May 22, 2020

Description

The Hungarian method solves the assignment problem. Given n people, n tasks, and an $n \times n$ cost (value) matrix, find the assignment that minimizes (maximizes) the cost (value). Although the problem can also be solved using the matrix directly, it is easier to reason about by using bipartite graphs.

Vocabulary

Definition: A *labelling* is attaching a number to each vertex of a graph: $\ell : V \to \mathbb{R}$.
Definition: A *feasible labelling* is one such that the sum of the labels of every edge are greater than or equal to its weight: $\ell(x) + \ell(y) \geq w(x, y)\ \forall x, y \in V$.
Definition: The *equality graph* is a subset of edges where the sum of the labels are equal to its weight: $G = (V, E_\ell)$, where $E_\ell = \{(x, y) \mid \ell(x) + \ell(y) = w(x, y)\}$.
Definition: A *matching* is a subset of edges such that every vertex is incident to at most one edge.
Definition: A *perfect matching* is a subset of edges such that every vertex is incident to exactly one edge.
Definition: A *maximum weight perfect matching* is a perfect matching with the maximum possible sum of edge weights.
Definition: A vertex is *matched* if it is incident to an edge in a matching. Otherwise it is *free*.
Definition: The *neighborhood* with respect to ℓ of a set S is the union of all the neighbors in E_ℓ of the vertices in S: $N_\ell(S) = \bigcup_{x \in S} \{y \mid (x, y) \in E_\ell\}$.
Definition: An *augmenting path* in E_ℓ is a path that starts and ends with unmatched edges, and alternates in between: $P = e_0 e_1 \dots e_n$, where n is even and $e_i \in M$ if i is odd.

Definition: The *trivial labelling* of a bipartite graph with partitions X and Y is to label each vertex in X with the weight of its largest incident edge, and label the vertices of Y with 0:

$$\ell(v) = \begin{cases} \max_y w(v,y) & \text{if } v \in X \\ 0 & \text{if } v \in Y \end{cases}$$

Theorems

Kuhn-Munkres Theorem: Let ℓ be a feasible labelling and M be a perfect matching in E_ℓ. Then M is a maximum weight perfect matching.

Proof. Let M' be an arbitrary perfect matching in G. Then

$$w(M') = \sum_{(x,y)\in M'} w(x,y) \leq \sum_{(x,y)\in M'} (\ell(x) + \ell(y))$$

Since M' is a perfect matching, every edge is incident to exactly one vertex, so

$$\sum_{(x,y)\in M'} (\ell(x) + \ell(y)) = \sum_{v\in V} (\ell(v))$$

Now, we have

$$w(M) = \sum_{(x,y)\in M} w(x,y) = \sum_{(x,y)\in M} (\ell(x) + \ell(y))$$

since $M \subseteq E_\ell$. Again, since M is a perfect matching,

$$\sum_{(x,y)\in M} (\ell(x) + \ell(y)) = \sum_{v\in V} (\ell(v))$$

Therefore, $w(M') \leq w(M)$, so M is a maximum weight perfect matching. □

This theorem tells us that if we find a perfect matching in an equality graph of G, it is guaranteed to be the optimal perfect matching.

Theorem: Let G be a bipartite graph partitioned into vertex sets X and Y and ℓ be a feasible labelling. Let $S \subseteq X$, $T = N_\ell(S) \neq Y$. Let

$$\alpha = \min_{(x\in S, y\notin T)} (\ell(x) + \ell(y) - w(x,y))$$

Let

$$\ell'(v) = \begin{cases} \ell(v) - \alpha & \text{if } v \in S \\ \ell(v) + \alpha & \text{if } v \in T \\ \ell(v) & \text{otherwise} \end{cases}$$

Then, all of the following are true:

1. ℓ' is a feasible labelling.
2. If $(x, y) \in E_\ell$ and $x \in S, y \in T$, then $(x, y) \in E_{\ell'}$.
3. If $(x, y) \in E_\ell$ and $x \notin S, y \notin T$, then $(x, y) \in E_{\ell'}$.
4. $N_{\ell'}(S) \neq T$.
5. If a matching $M \subseteq E_\ell$, then $M \subseteq E_{\ell'}$.

Proof.

1. There are 4 types of edges (x, y) to consider:
   (a) $x \in S, y \in T$. Then,
   $$\ell'(x) + \ell'(y) = \ell(x) - \alpha + \ell(y) + \alpha = \ell(x) + \ell(y) \geq w(x, y)$$
   (b) $x \notin S, y \notin T$. Then,
   $$\ell'(x) + \ell'(y) = \ell(x) + \ell(y) \geq w(x, y)$$
   (c) $x \in S, y \notin T$. Then,
   $$\ell'(x) + \ell'(y) = \ell(x) - \alpha + \ell(y)$$
   Since α was the minimum value, we know
   $$\begin{aligned} \alpha &\leq \ell(x) + \ell(y) - w(x, y) \\ -\alpha &\geq -\ell(x) - \ell(y) + w(x, y) \\ \ell(x) - \alpha + \ell(y) &\geq w(x, y) \\ \ell'(x) + \ell'(y) &\geq w(x, y) \end{aligned}$$

(d) $x \notin S, y \in T$. Then

$$\ell'(x) + \ell'(y) = \ell(x) + \ell(y) + \alpha \geq \ell(x) + \ell(y) \geq w(x,y)$$

Thus, ℓ' is a feasible labelling.

2. This follows from 1(a) since $\ell(x) + \ell(y) = w(x,y)$.

3. This follows from 1(b) since $\ell(x) + \ell(y) = w(x,y)$.

4. Let $x \in S, y \notin T$ be the vertices that gave us α. Since $N_\ell(S) = T$, we know that $y \notin N_\ell(S)$. From 1(c), using equality instead of inequality, we can see that $\ell'(x) + \ell'(y) = w(x,y)$. Therefore, $(x,y) \in E_{\ell'}$, so $y \in N_{\ell'}(S)$. Since $y \notin T$, $N_{\ell'}(S) \neq T$.

5. If $(x,y) \in M \subseteq E_\ell$,

$\square$

The Algorithm

The idea of the algorithm is to start with an empty matching and the trivial labelling. In each iteration, we add edges to the equality graph until we create an augmenting path that we can "flip" without causing any conflicts. Since the augmenting path starts and ends with unmatched edges, this will increase the size of our matching by one. The algorithm is as follows:

```
def hungarian(G):
  Start with trivial labelling L and empty matching M
  while M is not a perfect matching:
    Choose a free vertex u in X
    S = {u}
    T = {}

    if N_L(S) == T:
      alpha = min([L[x] + L[y] - w(x,y) for x in S, y not in T])
      L[x] -= alpha for x in S
      L[y] += alpha for y in T
```

```
    Choose y not in N_L(S)
    if y is matched to some x in S:
      S.add(x)
      T.add(y)
    else:
      for each edge e in augmenting path u...y:
        if e in M:
          M.remove(e)
        else:
          M.add(e)

  return M
```

Correctness

1. We can always start with trivial labelling and and empty matching

2.