Hungarian algorithm

Quentin Bertrand, Mathurin Massias, Titouan Vayer

Last updated: November 28, 2024

Contents

1 Dual of OT

Recall that the dual formulation of discrete OT is:

$$\max_{\phi \in \mathbb{R}^m, \psi \in \mathbb{R}^n} a^\top \phi + b^\top \psi \quad \text{s.t.} \quad \phi_i + \psi_j \leq C_{ij} \tag{1.1}$$

where a and b have positive entries. The variables ϕ and ψ are called (Kantorovich) potentials.

Two remarks: first since $\sum_i a_i = \sum_i b_i$, the dual objective is invariant if we add α to all entries of ϕ and subtract α to all entries of ψ, as this leaves $\phi_i + \psi_j$ invariant. Thus the solution of the dual problem is never unique; in some algorithm, we will need to handle this explicitly, e.g. by setting the first coordinate of ϕ to 0.

Second of all, for a given ϕ, the best ψ is as large as possible (since $b \geq 0$) while satisfying the constraint $\psi_j \leq c_{ij} - \phi_{ij}$ for all i. The latter cosntraint is is equivalent to $\psi_j \leq \min_i c_{ij} - \phi_i$, and so we are enclined to take $\psi_j = \min_i c_{ij} - \phi_i$: this is the largest value satisfying the feasibility constraint.

If we introduce the infimal convolution $Q_c(-\phi) = \left(\min_i c_{ij} - \phi_i\right)_j$, we can rewrite the dual problem writes in terms of ϕ only by replacing ψ by $Q_c(-\phi)$. This form is the so-called semi-dual. A similar argument shows that one must have $\phi = Q_{c^\top}(-\psi)$, and so ϕ can be replaced by $Q_{c^\top}(-Q_c(-\phi))$.

2 The Optimal assignment problem and the Hungarian algorithm

Resources & background The Wikipedia page explains the algorithm nicely in terms of graphs. Kuhn (the Kuhn of KKT conditions!) coined it the Hungarian algorithm in 1955 because it was based on the works of Hungarian mathematicians König and Egerváry. Also known as the Kuhn-Munkres algorithm, it had been in fact discovered by Jacobi and published posthumously in 1890 (in Latin!).

The notes for this chapter are based on the very clear "Notes de cours sur le Transport Optimal" (Sec. 6.4), in French, by Nathael Gozlan, Paul-Marie Samson and Pierre-André Zitt. We thank them for making this great material publicly available.

Overview The Hungarian algorithm solves the problem of optimal assignment. Like the simplex algorithm, it uses the KKT conditions for optimality, and thus relies on the dual potentials.

For a cost $C \in \mathbb{R}^{n\times n}$ the optimal assignment problem is the optimal transport problem with $a = b = 1_n$:

$$\min_{P\in\mathbb{R}^{n\times n}} \sum_{i,j=1}^{n} C_{ij}P_{ij} \quad \text{s.t.} \quad \sum_{i=1}^{n} P_{ij} = 1,\ \sum_{j=1}^{n} P_{ij} = 1,\ P_{ij} \in \{0,1\} \quad \forall i,j \tag{2.1}$$

Because of the constraints, a feasible P has exactly one nonzero entry per row and per line; this entry is equal to 1: P is a *permutation matrix*.

Exercise 2.1. *Show that the optimal assignment problem is equivalent to finding a perfect matching with minimum total cost in the complete bipartite graph with two sets of n nodes L and R, and edges from every node of L to every node of R (that is the definition of complete bipartite).*

Working with discrete variables is hard and a bruteforce approach would have combinatorial complexity, so we introduce the relaxed version of Problem 2.1:

$$\min_{P\in\mathbb{R}^{n\times n}} \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}P_{ij} \quad \text{s.t.} \quad \sum_{i=1}^{n} P_{ij} = 1, \sum_{j=1}^{n} P_{ij} = 1, P_{ij} \geq 0 \quad \forall i,j \tag{2.2}$$

The main interest of the relaxed version is that it is *convex*; better, it is a linear program, for which we can leverage all the artillery developed in the last 70 years. It is a *relaxation* in the following sense: the feasible set clearly strictly contains (for $n > 1$) the set of permutations, hence the optimal value of Problem 2.2 is lower than that of (2.1).

What do we lose by the relaxation? As a matter of fact, nothing! To prove, we will rely on a very strong theorem, and geometrical objects.

Definition 2.1. *The transport polytope $\{P \in \mathbb{R}^{n\times n} : P \geq 0, P1 = 1_n, P^\top 1 = 1_n\}$ is called the Birkhoff polytope. Its elements are called bistochastic matrices.*

Proposition 2.2 (Birkhoff theorem)**.** *The extremal points of the Birkhoff polytope are the permutation matrices. Recall that x in an extremal point of the convex set C if $x = (y + z)/2$ for $y, z \in C$ implies that $y = z = x$; in layman's terms, x is not on a nontrivial segment contained in C.*

Proposition 2.3 (Fundamental theorem of linear programming)**.** *A minimizer*[1] *of a linear function over a polytope either is an extremal point of said polytope, either lies a face of the polytope, and then this whole face is composed of minimizers.*

Since Problem (2.2) is a linear program, Propositions 2.2 and 2.3 together state that at least a minimizer of it is a permutation matrix. Hence, Problems (2.1) and (2.2) actually have the same optimal value, and share one minimizer – though the relaxed problem may have more minimizers (e.g. in the brutally degenerate case $C = 0$...).

Duality The dual of the relaxed Problem 2.2 is:

$$\max_{\phi,\psi\in\mathbb{R}^n} \sum_{i=1}^{n} \phi_i + \sum_{j=1}^{n} \psi_j \quad \text{s.t.} \quad \phi_i + \psi_j \leq C_{ij} \quad \forall i, j \tag{2.3}$$

Note: this is a formulation of the dual where we have used the stationary condition on P (namely $C_{ij} - \mu_{ij} - \phi_i - \psi_j = 0$) to replace the multiplier μ associated to the constraint $P \geq 0$ by its value in terms of ϕ and ψ.

Exercise 2.2. *Prove that the dual of* (2.2) *is indeed* (2.3)*.*

The KKT conditions for problem (2.2) are (we omit primal stationarity, that is satisfied de facto with our choice of μ):

$$P \geq 0 \tag{2.4}$$

$$\sum_i P_{ij} = 1 \tag{2.5}$$

$$\sum_j P_{ij} = 1 \tag{2.6}$$

$$P_{ij}(C_{ij} - \phi_i - \psi_j) = 0 \qquad \text{(complementary slackness)} \tag{2.7}$$

2.1 The Hungarian algorithm

The Hungarian algorithm proceeds iteratively, in the following fashion: at each iteration it has as variables

[1]the same result hold with maximizers replacing minimizers

- an (unfeasible) primal variable P, called partial assignation: it has at most one 1 per line and per row (but for some rows or column, it has none). A critical property is that the number of assignments in P increases by 1 at each iteration
- dual potentials ϕ and ψ, that are feasible: $\phi_i + \psi_j \leq C_{ij}$ and satisfy complementary slackness: if $P_{ij} \neq 0$, the constraint is tight.

Following the graph matching terminology, we will say that node $i \in L$ is assigned to $j \in R$ if $P_{ij} = 1$; the couple (i, j) is referred to as an edge and we say that the edge is *assigned* if $P_{ij} = 1$, *saturated* if $\phi_i + \psi_j = C_{ij}$.

As done in Figure 1, to understand the algorithm it helps to visualize a (complete) bipartite graph with n nodes on the left L, n nodes on the right R, all possible edges from left to right, with saturated edges in dashed, and assigned edges in red.

Initialization The algorithm first initializes P, ϕ and ψ as follows:

- $P = 0$
- $\phi = Q_c(-0) = (\min_j C_{ij})$ – i.e. the best possible ϕ such that $(\phi, 0)$ is feasible
- $\psi = Q_{c^\top}(-\phi) = \min_i C_{ij} - \phi_i$ – i.e. the best possible feasible ψ given the choice we just made for ϕ.
- for each $i \in L$, if there exists an unassigned $j \in R$ such that edge (i, j) is saturated, we pick the lowest such j and set $P_{ij} = 1$.

Exercise 2.3. *Show that the initialization for* $C = \begin{pmatrix} 1 & 4 & 2 \\ 3 & 5 & 6 \\ 2 & 1 & 5 \end{pmatrix}$ *is:* $\phi = (1, 3, 1)$, $\psi = (0, 0, 1)$ *and the saturated and assigned edges are like in Figure 1*

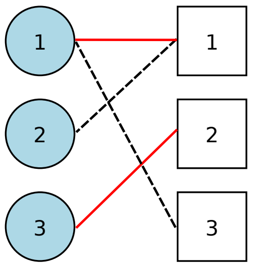

Figure 1: Initialization of assigned (red) and saturated (dashed) edges for the example.

Main loop After the initialization, the main loop repeats the three following steps until convergence:

(M.i) look for an *augmenting path* among saturated edges,

(M.ii) if there is one, *invert* it,

(M.iii) if there is none, increase the number of saturated edges by modifying the dual potentials.

What is an augmenting path? A path is *alternating* if it starts from an unassigned node in the left, where each left-right edge is unassigned, and each right-left edge is assigned. It is *augmenting* if in addition it ends on the right, on a non assigned node; in that case we observe that it must contain k unassigned edges and $k-1$ assigned edges.

Exercise 2.4. *Write down the main loop of the Hungarian algorithm for the matrix C of Exercise 2.3. What is the minimal cost for this assignment problem, and what is the associated assignment? What are the optimal dual potentials? Check that the KKT conditions are satisfied.*

We now explain step (M.iii), that we need to perform when we are unable to find an augmenting path. We are considering an unassigned left node, say u. Consider the set of all maximal alternating paths starting from u; this can be constructed easily by bread-first search. It forms a tree, $\mathcal{T}$. Then, let's write S (resp. T) for the set of left (resp. right) nodes on that tree. Let

$$\delta = \min_{i \in S, j \notin T} \{C_{ij} - \phi_i - \psi_j\} \tag{2.8}$$

Notice that δ is strictly positive: if there was some $i \in S, j \in T$ such that $C_{ij} - \phi_i - \psi_j = 0$, the edge ij would be saturated; we could thus add it to the tree $\mathcal{T}$ (since there must already be an alternated path from u to i).

We then modify ϕ and ψ as follows:

$$\phi_i = \begin{cases} \phi_i + \delta & \text{if } i \in S \\ \phi_i & \text{otherwise} \end{cases} \qquad \text{and} \qquad \psi_j = \begin{cases} \phi_j - \delta & \text{if } j \in T \\ \psi_j & \text{otherwise} \end{cases} \tag{2.9}$$

This change is designed in order to have the following effects:

- ϕ, ψ remains feasible
- at least one edge leaving from S becomes saturated (by choice of the value of δ)
- saturated edges from S to T remain saturated
- assigned edges remain saturated

Let us prove that we maintain feasibility. Let (i, j) an edge in the complete bipartite graph. We have four cases to distinguish:

(i) $i \in S$, $j \in T$: the sum $\phi_i + \psi_j$ remains unchanged (this proves that saturated edges from S to T remain saturated)

(ii) $i \in S, j \notin T$: the sum increases by δ, but by our choice of δ, it remains less than C_{ij}; by choice of δ one edge at least becomes saturated (because we increase the sums until one becomes saturated)

(iii) $i \notin S$, $j \in T$: the sum $\phi_i + \psi_j$ decreases by δ so remains less than C_{ij}; this may desaturate the edge

(iv) $i \notin S$, $j \notin T$: the sum does not change.

If $v \in T$ is assigned to u, then vu is necessarily in the tree, so $u \in S$: assigned edges are thus either fully in the tree, either fully outside, and are thus not desaturated.

After this modification, we look for augmenting paths again; the alternating tree will have one more edge than previously. This means that the subloop we will always be able to end up with an augmenting path, as the alternating tree cannot have an unbounded number of edges.

The algorithm therefore terminates: we always manage to find an augmenting path, and each time we do so, by flipping it we increase the number of assignments by 1, while the latter is never decreased.

It converges to a solution, because we always maintain duality feasability complementary slackness: as soon as we end up satisfying the primal feasability, we have a KKT point.

Exercise 2.5. *Show how to use the Hungarian algorithm to find the closest permutation matrix to a given matrix.*

Exercise 2.6. *Write down the steps of the hungarian algorithm for* $C = \begin{pmatrix} 7 & 9 & 8 & 9 \\ 2 & 8 & 5 & 7 \\ 1 & 6 & 6 & 9 \\ 3 & 6 & 2 & 2 \end{pmatrix}$.

References