Minimum Spanning Trees

Given a weighted undirected graph G = (V,E,w), one often wants to find a minimum spanning tree (MST) of G: a spanning tree T for which the total weight $w(T) = \sum_{(u,v\in)T} w(u,v)$ is minimal.

Input: A connected, undirected weighted graph G = (V,E,w) Output:

A spanning tree T such that the total weight

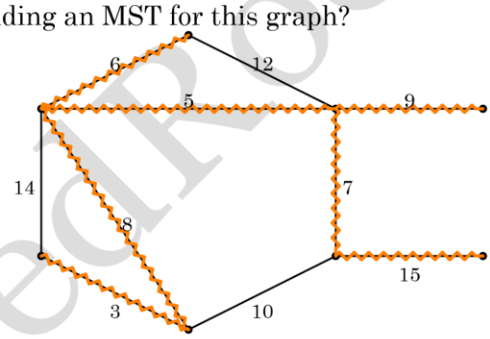
$$ww(u,v)_{T}$$

is minimal.

For example, if V is the set of buildings in a city and E is the complete graph on V, and if $w(u,v)$ is the distance between u and v, then a minimum spanning tree would be very useful in building a minimum-length fiber-optic network, pipeline network, or other infrastructure for the city.

There are many other, less obvious applications of minimum spanning trees. Given a set of data points on which we have defined a metric (i.e., some way of quantifying how close together or similar a pair of vertices are), we can use an MST to cluster the data by starting with an MST for the distance graph and then deleting the heaviest edges. (See Figure 3.3 and Lecture 21.) If V is a set of data fields and the distance is mutual information, then this clustering can be used to find higher-order correlative relationships in a large data set.

How would we go about finding an MST for this graph?



There are several sensible heuristics:

- *Avoid large weights.* We would like to avoid the 14 and 15 edges if at all possible.

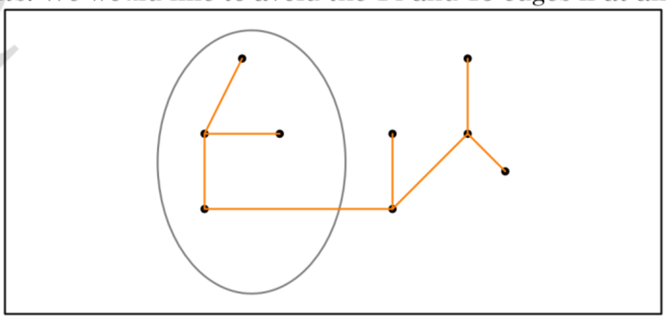


Figure 3.4. Illustration of the MST property.

- Include small weights. If the 3 and 5 edges provide any benefit to us, we will probably want to include them.

Source:Stanford CS 161 Lecture

- Some edges are inevitable. The 9 and 15 edges must be included in order for the graph to be connected.
- Avoid cycles, since an MST is not allowed to have cycles.

Some more thoughts:

- Would a greedy algorithm be likely to work here? Is there something about the structure ofthe problem which allows locally optimal decisions to be informed enough about the global solution?
- Should we start with all the edges present and then remove some of them, or should we startwith no edges present and then add them?
- Is the MST unique? Or are there multiple solutions?

A key observation to make is the following: If we add an edge to an MST, the resulting graph will have a cycle, by Proposition 3.1(iv). If we then remove one of the edges in this cycle, the resulting graph will be connected and therefore (by Proposition 3.1(iii)) will be a spanning tree. If we have some way of knowing that the edge we removed is at least as heavy as the edge we added, then it follows that we will have a minimum spanning tree. This is the idea underlying the socalled "MST property."

Theorem 3.3 (MST Property). Let G = (V,E,w) be a connected, weighted, undirected graph. Let U

be a proper nonempty subset of V.[1] Let S be the set of edges (x, y) with $x \in U$ and $y \in V \setminus U$.

Suppose

(u,v) is the lightest edge (or one of the lightest edges) in S. Then there exists an MST containing

(u,v).

This partition of V into U and $V \setminus U$ is called a "cut." An edge is said to "cross" the cut if one endpoint is in U and the other endpoint is in $V \setminus U$. Otherwise the edge is said to "respect" the cut. So S is the set of edges which cross the cut and $E \setminus S$ is the set of edges that respect the cut. An edge in S of minimal weight is called a "light edge" for the cut.
Proof. Let $T \subseteq E$ be an MST for G, and suppose (u,v) 6∈ T. If we add the edge (u,v) to T, the

resulting graph To has a cycle. This cycle must cross the cut $(U, V \setminus U)$ in an even number of places,

so in addition to (u,v), there must be some other edge (uo,vo) in the cycle, such that (uo,vo) crosses

the cut. Remove (uo,vo) from To and call the resulting graph Too. We showed above that Too is a

spanning

tree. Also, since (u,v) is a light edge, we have w(u,v) ≤ w(uo,vo). Thus

$$w(T_{oo}) = w(T) + w(u,v) - w(u_o,v_o) \leq w(T).$$

Since T is a minimum spanning tree and Too is a spanning tree, we also have w(T) ≤ w(Too).

[1] That is, ;áU áV.

Therefore $w(T) = w(T'')$ and T'' is a minimum spanning tree which contains (u,v). □

Corollary 3.4. Preserve the setup of the MST property. Let T be any MST. Then there exists an edge $(u',v') \in T$ such that $u' \in U$ and $v' \in V \setminus U$ and $\left(T \setminus \{(u',v')\}\right) \cup \{(u,v)\}$ is an MST.

Source: CS 161 Lecture 1

Proof. In the proof of the MST property, $T'' = \left(T \setminus \{(u',v')\}\right) \cup \{(u,v)\}$. □

Corollary 3.5. Let $G = (V,E,w)$ be a connected, weighted, undirected graph. Let T be any MST and let $(U,V \setminus U)$ be any cut. Then T contains a light edge for the cut. If the edge weights of G are distinct, then G has a unique MST.

Proof. If T does not contain a light edge for the cut, then the graph T'' constructed in the proof of the MST property weighs strictly less than T, which is impossible.

Suppose the edge weights in G are distinct. Let M be an MST. For each edge $(u,v) \in M$, consider the graph $(V, M \setminus \{(u,v)\})$. It has two connected components[2]; let U be one of them. The only edge in M that crosses the cut $(U,V \setminus U)$ is (u,v). Since M must contain a light edge, it follows that (u,v) is a light edge for this cut. Since G has distinct edge weights, (u,v) is the unique light edge for the cut, and every MST must contain (u,v). Letting (u,v) vary over all edges of M, we find that every MST must contain M. Thus, every MST must equal M. □

[2] Given a graph G, we say a vertex v is reachable from u if there exists a path from u to v. For an undirected graph, reachability is an equivalence relation on the vertices. The restrictions of G to each equivalence class are called the connected components of G. (The restriction of a graph $G = (V,E)$ to a subset $V^0 \subseteq V$ is the subgraph (V_0,E_0), where E_0 is the set of edges whose endpoints are both in V_0.) Thus each connected component is a connected subgraph, and a connected undirected graph has only one connected component.