An Algorithmist's Toolkit	Brown CSCI 2952T – Spring 2026

Lecture 8: Metrics, LDDs, Multicut

March 18, 2026 Lecturer: D Ellis Hershkowitz Scribe: Sriniketh Vangaru, Zihan Jack Zhang

1 Metrics

Metrics, or metric spaces, are a mathematical formulation and formalization of the intuitive concept we know as "distance".

Definition 1. *A **metric space**, or **metric** for short, consists of a set of "points" V and a function $f : V \times V \to \mathbb{R}_{\geq 0}$ such that:*

1. *$f(x,x) = 0,\ \forall x \in V$. (In reality, we typically place the stricter condition that $f(x,y) = 0 \iff x = y$; if that doesn't hold, then the metric is considered a **pseudo-metric**, which we will ignore here.)*
2. *$f(x,y) = f(y,x),\ \forall x, y, \in V$. (The "symmetry" condition.)*
3. *$f(x,y) \leq f(x,z) + f(z,y),\ \forall x, y, z \in V$. (The triangle inequality.)*

Examples of **Definition 1**:

- $(V := \mathbb{R}^n, f := d)$, where $d(x,y) := \|y - x\| = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$.
  - We get (1) and (2) trivially.
  - (3) follows from the inequalities lecture.
- $(V := V, f := d)$, where $V \subseteq \mathbb{R}^n$ and $d(x,y) := \|y - x\|$.
  - For the same reasons as above.
  - **Note:** We do not require closure under any combinations of elements; the metric function d just needs to operate on the existing elements of V.
- $(V := V, f := d_G)$, where $G = (V, E, w)$ with $w : E \to \mathbb{R}$ is an edge-weighted, undirected graph and $d_G(u,v)$ gives the length of the shortest $u \to v$ path.
  - Again, (1) and (2) are trivial.
  - For (3):
    - ∗ Let $x, y, z \in V$ be 3 arbitrary nodes, let $P_{i,j}$ be the shortest path $(i, v_1, v_2, \ldots, v_k, j)$ from node i to j, and let $w(P) = \sum_{e \in P} w(e)$ be the length of path P.
    - ∗ Then, since $d_G(x,y) = w(P_{x,y})$ is the shortest path from x to y, $d_G(x,y) \leq w(Q_{x,y})$ for all paths $Q_{x,y}$ from x to y.

* Observe that the concatenated path $P_{x,z} \oplus P_{z,y}$ is a path from x to y, as shown in **Figure 1**.
* Therefore, $d_G(x, y) \leq w(P_{x,z} \oplus P_{z,y}) = w(P_{x,z}) + w(P_{z,y}) = d_G(x, z) + d_G(z, y)$, as required.

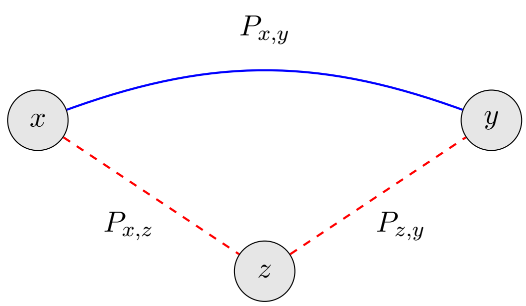

Figure 1: Two paths between x and y: the direct path $P_{x,y}$ (blue) and the path through z via $P_{x,z}$ and $P_{z,y}$ (red).

Claim 1. *If (V, f) is a metric space, then $f(x, y) \geq 0$, $\forall x, y \in V$.*

Proof. We have that

$$\begin{aligned} 0 &= f(x, x) && \text{by (1)} \\ &\leq f(x, y) + f(y, x) && \text{by (3)} \\ &= f(x, y) + f(x, y) && \text{by (2)} \\ &= 2 \cdot f(x, y) \end{aligned}$$

so then $0 \leq f(x, y)$. □

Definition 2. *The **metric framework** provides a way to solve problems using metric spaces. It involves 3 steps:*

1. *Find a metric space in your problem*
2. *Find some structure in your metric space*
3. *Use the structure of your metric to solve your problem*

2 The Multicut Problem

Definition 3. *The **multicut** problem, a generalization of the min-cut problem, is defined as follows:*

* ***Given:*** *A graph $G = (V, E)$ and some vertex pairs (s_i, t_i)*
* ***Find:*** *An edge subset $F \subseteq E$ of minimum size $|F|$ such that $\forall i$, s_i is not connected to t_i in the graph $(V, E \setminus F)$*

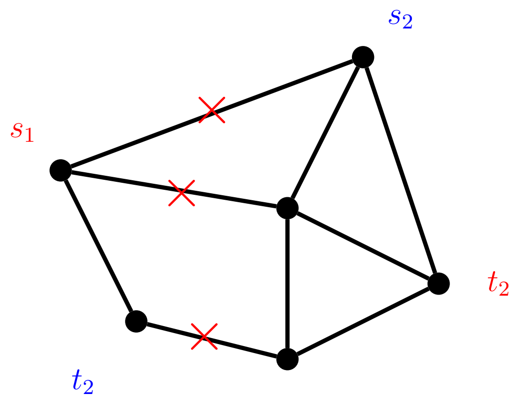

Figure 2: An example of the multicut problem. The red crosses denote the edges in the cut $F \subseteq E$.

An example is shown in **Figure 2**.

Fact 1. *The multicut problem is NP-hard. However, if we let* $OPT := \min_{feasible\ F} |F|$*, then it is still possible to find* $F \subseteq E$ *such that* $|F| \leq \alpha \cdot OPT$ *for small* $\alpha > 1$ *in P.*

More generally, if we are given a family of minimization problems π (such as all multicut instances) with objective function $w : W \to \mathbb{C}$ where W is the solution space, then a randomized algorithm $\mathcal{A} : \pi \to W$ is an α-approximation if $\mathbb{E}\left[w(\mathcal{A}(P))\right] \leq \alpha \cdot \mathrm{OPT}(P),\ \forall P \in \pi$.

Theorem 1. *Let* $n := |V|$ *and* $m := |E|$*. Then there exists a* $\mathrm{poly}(n, m)$*-time randomized* $O(\log(n))$*-approximation for multicut.*

The obvious metric to use here is shortest path distances, but we'll be a little bit more clever soon.

3 Low-Diameter Decompositions (LDDs)

The core idea is that an LDD is a random clustering of metric points such that nearby points are probably in the same cluster.

Definition 4. *A* ***clustering*** *of* V *is a partition of* V *into* $C = \{V_1, V_2, \ldots\}$*.*

Definition 5. *The* ***diameter*** *of a set of points* V_i *is* $\Delta(V_i) = \max_{u,v \in V} d(u, v)$*. The diameter of a clustering* C *is* $\Delta(C) = \max_i \Delta(V_i)$*.*

An example is shown in **Figure 3**.

Definition 6. *Two nodes in a graph* $u, v \in V$ *are* ***separated*** *by a clustering* C *if* $u \in V_i$ *and* $v \in V_j$ *for* $i \neq j$*.*

Definition 7. *A* ***low-diameter decomposition (LDD)*** *with diameter* Δ *and quality factor* q *is a distribution over the set of possible* Δ*-diameter clusterings* $\mathcal{C}$ *such that:*

$$P_{C \sim \mathcal{C}}(u \text{ and } v \text{ are separated by } C) \leq q \cdot d(u, v),\ \forall u, v \in V$$

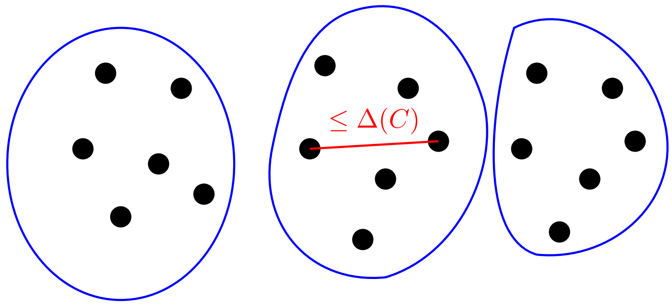

Figure 3: A clustering $C = \{V_1, V_2, V_3, \ldots\}$ of V, where the diameter of each cluster is at most $\Delta(C)$.

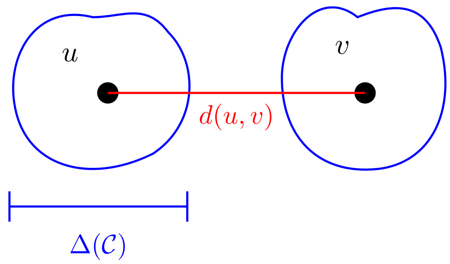

Figure 4: Nodes with a distance larger than the diameter of the cluster.

Some intuitions about **Definition 7**:

- Why does the separation probability for u and v depend on the distance $d(u, v)$? The farther u and v are, the higher the chance that they're not connected in C. For instance, as we see in **Figure 4**, if $d(u, v) > \Delta$, then $P(u \text{ and } v \text{ are separated}) = 1$.
- Why is there a tradeoff between Δ and q? We can look at the extreme cases, as shown in **Figure 5**: if the clustering approaches a case where all nodes are in one cluster, then Δ gets high while q gets low, while if the clustering approaches a case where all nodes are in their own clusters, then Δ gets low while q gets high.
- Ideally, we would like to keep Δ low while also keeping q low.

Claim 2. *If $G = (V, E, w)$ and $\mathcal{C}$ is an LDD with quality q, then $\forall (u, v) = e \in E$:*

$$P(u \text{ and } v \text{ are separated}) \leq q \cdot w(e)$$

Proof. *The edge e is a path from u to v, so $d_G(u, v) \leq w(e)$. Then $P(u \text{ and } v \text{ are separated}) \leq q \cdot d_G(u, v) \leq q \cdot w(e)$.* □

Claim 3. *If $G = (V, E)$ is a unit-length path, then $\forall \Delta > 0$, the metric space (V, d_G) has a poly-time computable LDD with $q \leq \frac{1}{\Delta}$.*

Proof. *Let $r \sim U[0, \Delta - 1] \cap \mathbb{Z}$, $V_0 = \{1, 2, \ldots, r\}$, and $V_i = (r + \Delta \cdot (i - 1), r + \Delta \cdot i] \cap \mathbb{Z}\ \ \forall i \geq 1$. Then Δ is the size of each cluster V_i for $i \geq 1$, and the clustering $C = \{V_0, V_1, V_2, \ldots\}$ has diameter $\Delta - 1 \leq \Delta$ by construction.*

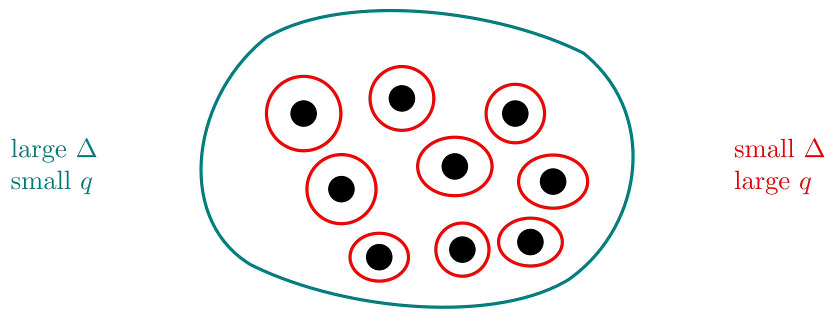

Figure 5: Example of high-diameter and low-diameter clusterings.

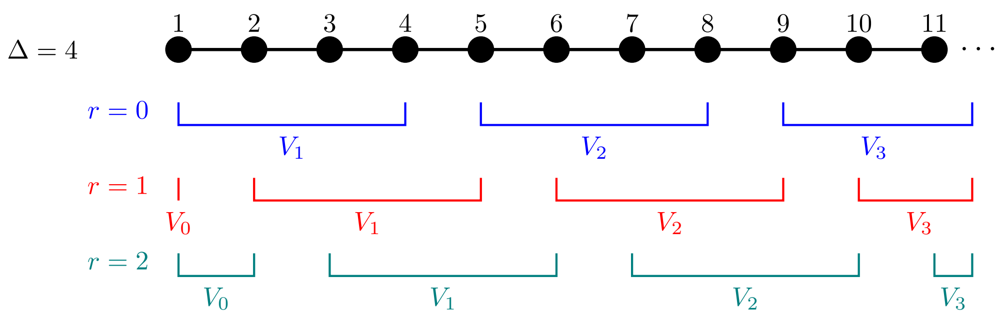

Figure 6: Example of a path consisting of unit-length edges. r controls the shift of the intervals, where each interval has Δ nodes in it.

Now, consider two nodes $x, y \in V$ such that $x \leq y$. There are 2 cases to consider:

- *If $d(x, y) \geq \Delta$, then regardless of the value of r, there will be at least one split between V_i and V_{i+1} that falls on the path between x and y, where $i \geq 0$; therefore, $P(x \text{ and } y \text{ are separated}) = 1$.*
- *If $d(x, y) < \Delta$, then $P(x \text{ and } y \text{ are separated}) = \frac{d(x,y)}{\Delta}$, since r has Δ possible choices and only $d(x, y)$ of them will place a split between x and y. This is shown in* ***Figure 6***.

As a result, we have that $P(x, y \text{ are separated}) \leq \frac{d(x,y)}{\Delta} = q \cdot d(x, y)$, where $q \leq \frac{1}{\Delta}$. $\square$

4 Existence of LDDs and Algorithm for Multicut

We would like to prove the following:

Claim Given any n-point metric space (V, d), for any $\Delta > 0$, there exists a poly-time computable LDD with $q \leq \frac{4 \ln n}{\Delta}$.

As usual, we first describe the algorithm:

Note that $\text{diam}(C) \leq \Delta$ (triangle inequality).

Algorithm 1 LDD

1: $r \sim \mathcal{U}(0, \Delta/2]$
2: $\pi : [n] \to V$ a permutation chosen uniformly randomly
3: **for** each $i = 1, \ldots, n$ **do**
4: $\quad C_i \leftarrow B(\pi(i), r) \setminus \bigcup_{j<i} C_j$ ▷ Remove previously selected points
5: **end for**
6: **return** $C \coloneqq \{C_i\}$

Proof. For a pair $t = \{u, v\}$, we want to show that

$$\mathbb{P}(\text{u, v separated}) \leq \frac{4 \ln n}{\Delta} d(u, v).$$

Let $d(x, t) \coloneqq \min(d(x, u), d(x, v))$, and let all the vertices $u_1, \ldots, u_n$ be sorted by $d(\cdot, t)$. We say that a vertex u_i **cuts** t if and only if $|t \cap B(u_i, r)| = 1$, i.e. the r-ball around u_i captures exactly one element in the pair t. In particular, if u_i cuts t, then $r \in [d(u_i, t), d(u_i, t) + d(u, v)]$, so

$$\mathbb{P}(u_i \text{ cuts } t) \leq 2\frac{d(u, v)}{\Delta}.$$

We say u_i **settles** t if and only if i is the minimum i such that $t \cap B(u_i, r) \neq \varnothing$. Thus u_i settles t only if u_i precedes $u_1, \ldots, u_{i-1}$ under π. We have that

$$\mathbb{P}(u_i \text{ settles } t \mid u_i \text{ cuts } t) \leq \frac{1}{i}.$$

The pair t is separated if and only if there exists i such that u_i settles and cuts t. We have

$$\begin{aligned}
\mathbb{P}(t \text{ separated}) &= \sum_i \mathbb{P}(u_i \text{ settles and cuts } t) \\
&= \sum_i \mathbb{P}(u_i \text{ cuts } t)\mathbb{P}(u_i \text{ settles } t \mid u_i \text{ cuts } t) \\
&= \sum_i \frac{2d(u, v)}{\Delta} \frac{1}{i} \\
&= \frac{2d(u, v)}{\Delta} \sum_i \frac{1}{i} \\
&\leq 4\frac{d(u, v)}{\Delta} \ln n.
\end{aligned}$$

□

From now on we use the special case of the claim where $\Delta = 8 \ln n$ which gives $q = \frac{1}{2}$ and the separation probability is bounded by $\frac{1}{2} d(u, v)$.

We now formulate the LP relaxation of the problem. Let x_e be variables for $e \in E$, so $x \in \mathbb{R}^m$, and we have

$$\begin{aligned}
\min \quad & \sum_e x_e \\
\text{s.t.} \quad & \sum_{e \in P} x_e \geq 1 \quad P \text{ (paths from } s_i \text{ to } t_i\ \forall i) \\
& x_e \in [0, 1] \quad \forall e \in E
\end{aligned}$$

We have the following observations about this relaxation. Firstly $\mathrm{OPT}_{\mathrm{LP}} \leq \mathrm{OPT}$. Secondly, we can solve the LP in poly(n) time with ellipsoid. Let x be an optimal LP solution we propose the following algorithm:

Algorithm 2 Separation

1: Let d_C be the graph distance metric where the weights $w \coloneqq 9 \ln n x$
2: Let C be a $\Delta = 8 \ln n$ LDD ▷ By claim above
3: **return** $F \coloneqq \{e \in E : e \text{ separated by } C\}$

Is the set F produced by 2 feasible? Yes, consider the pair (s_i, t_i), we have $d_C(s_i, t_i) \geq 9 \ln n$, but $\mathrm{diam}(C) = 8 \ln n$, so this pair is separated.

How good is this algorithm? We have

$$\begin{aligned}
\mathbb{E}|F| &= \sum_e \mathbb{P}(e \text{ separated by } C) \\
&\leq \sum_{e=\{u,v\}} \frac{d_C(u,v)}{2} \\
&\leq \sum_{e=\{u,v\}} \frac{w(u,v)}{2} \\
&= \sum_e \frac{9 \ln n x_e}{2} \\
&= \frac{9 \ln n}{2} \sum_e x_e \\
&= \frac{9 \ln n}{2} \mathrm{OPT}_{\mathrm{LP}} \\
&\leq O(\log n) \mathrm{OPT}.
\end{aligned}$$