1 Distances and Metric Spaces

Given a set X of *points*, a *distance function* on X is a map $d : X \times X \to \mathbb{R}_+$ that is symmetric, and satisfies $d(i,i) = 0$ for all $i \in X$. The distance is said to be a *metric* if the *triangle inequality* holds, i.e.,

$$d(i,j) \leq d(i,k) + d(k,j) \qquad \forall i,j,k \in X.$$

It is often required that metrics should also satisfy the additional constraint that $d(i,j) = 0 \iff i = j$, and the distance is called a *pseudo-metric* or a *semi-metric* if this additional condition does not hold. We will be very lax with this distinction; however, you now know what these terms mean, just in case.

1.1 Finite Metrics and Graphs

In this class, we shall usually study *finite metric spaces*, i.e., spaces (X,d) where X is a finite set of points; we will use n to denote the size of X. As described above, a metric can be specified by $\binom{n}{2}$ non-negative numbers, which give the distance between unordered pairs $\{i,j\} \in \binom{X}{2}$. E.g., the following matrix completely specifies a metric:

	a	b	c	d	e
a	0	3	8	6	1
b		0	9	7	2
c			0	2	7
d				0	5
e					0

Figure 1: A metric specified by a distance matrix.

It is often difficult to visualize metrics when specified thus, and hence we will use a natural correspondence between graphs and metrics. Given a graph G on n vertices endowed with lengths on the edges, one can get a natural metric d_G by setting, for every $i,j \in V(G)$, the distance $d_G(i,j)$ to be the length of the *shortest-path* between vertices i and j in G.

Conversely, given a metric (X,d), we can obtain a weighted graph $G(d)$ *representing* or *generating* the metric thus: we set X to be the vertices of the graph, add edges between all pairs of vertices, and set the length of the edge $\{i,j\}$ to be $d(i,j)$. It is trivial to see that the shortest path metric $d_{G(d)}$ is identical to the original metric d.

In fact, we may be able to get a "simpler" graph representing d by dropping any edge e such that the resulting metric $d_{G(d)-e}$ still equals d. We can continue removing edges until no more edges can be removed without changing the distances; the resulting minimal graph will be called the *critical graph* for the metric d. It can be checked that the critical graph for the metric above is the tree given in Figure 2.

We will usually blur the distinction between finite metrics and graphs, unless there are good reasons not to do so.

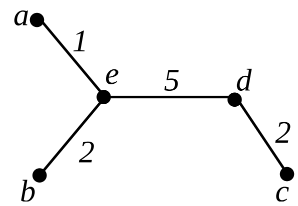

Figure 2: The graph corresonding to the metric of Figure 1

1.2 Infinite Metric spaces

The most commonly used infinite metric spaces are obtained by equipping real space $\mathbb{R}^k$ with one of the so-called Minkowski norms ℓ_p for any $1 \leq p \leq \infty$. Given a point $x \in \mathbb{R}^k$, its ℓ_p length is given by

$$\|x\|_p = (\sum_{i=1}^{k} |x_i|^p)^{1/p} \qquad \text{for } 1 \leq p < \infty;$$

the special case of the ℓ_∞-norm specified by $\|x\|_\infty = \max_{i=1}^{k}\{|x_i|\}$. (All this can be extended to non-finite dimensions, but we will not encounter such situations in this class.) Given two points $x, y \in \mathbb{R}^k$, the ℓ_p-distance between them is naturally given by $\|x - y\|_p$.

Some of these spaces should be familiar to us: ℓ_2 is just Euclidean space, while ℓ_1 is real space endowed with the so-called Manhattan metric. It is often instructive to view the *unit balls* in these ℓ_p metrics; here are the balls for $p = 1, 2, \infty$:

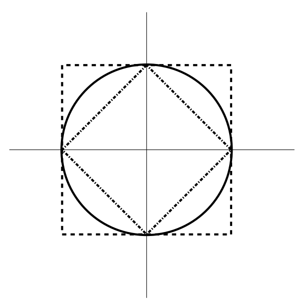

Figure 3: Unit balls in ℓ_1, ℓ_2 and ℓ_∞.

1.3 Embeddings

This class deals with the properties of metric spaces and for this we will have to analyze basic metric spaces and the similarities and differences between them. Formally, we compare metric spaces by using an embedding.

Definition 1.1 *Given metric spaces* (X, d) *and* (X, d') *a map* $f : X \to X'$ *is called an* embedding*. An embedding is called* distance-preserving *or* isometric *if for all* $x, y \in X$, $d(x, y) = d'(f(x), f(y))$.

Note that embeddings are a generic term for any map from a metric into another; transforming the metric (X, d) into its graphical representation $G(d)$ also gave us an isometric embedding.

We call a finite metric (X, d) an ℓ_p-metric if there exists an embedding from X into $\mathbb{R}^k$ (for some k) such that $\|f(i) - f(j)\|_p = d(x, y)$; by a cavalier abuse of notation, we will often denote this by $d \in \ell_p$. To denote the fact that the ℓ_p space had k dimensions, we will call the space ℓ_p^k as well.

1.4 Distortion

It is very rare to find cases where isometric embeddings can be found between two spaces of interest, and hence we often have to allow the mappings to alter distances in some (hopefully restricted) fashion.

There are many notions of "close"; most of the course will focus on the following notions. Given two metrics (X, d) and (X', d') and a map $f : X \to X'$, the *contraction* of f is the maximum factor by which distances are shrunk, i.e.,

$$\max_{x,y \in X} \frac{d(x, y)}{d'(f(x), f(y))},$$

the *expansion* or *stretch* of f is the maximum factor by which distances are stretched:

$$\max_{x,y \in X} \frac{d'(f(x), f(y))}{d(x, y)},$$

and the *distortion* of f, denoted by $\|f\|_{\text{dist}}$, is the product of the distortion and the expansion. An isometric embedding has distortion 1, and this is the lowest it can get. (Right?)

Another equivalent definition is the following: the distortion of f, denoted by $\|f\|_{\text{dist}}$, is the smallest value $\alpha \geq 1$ for which there exists an $r > 0$ such that for all $x, y \in X$,

$$r\, d(x, y) \leq d'(f(x), f(y)) \leq \alpha\, r\, d(x, y).$$

(Check.) A very useful property of distortion is that it is invariant under scaling; i.e., replacing d and d' by, say $10\, d$ and $\frac{1}{3}\, d'$ does not change $\|f\|_{\text{dist}}$. Hence, in many arguments, we will feel free to rescale embeddings to make arguments easier.

2 Why embeddings?

While we may want to investigate the properties of metric spaces for their rich mathematical content, the study is rewarding from an algorithmic viewpoint as well. The algorithms based on metric methods can be roughly classified as follows.

1. **Metric Data**
   The input data for the problem at hand is a metric or can be pre-processed to form a metric. The idea is to embed the metric into (a combination of) simpler metrics, on which the problem can be solved more easily. If the embedding has a distortion larger than 1 this technique results in approximaton algorithms for the given problem. Otherwise it gives an exact solution.

2. **Metric Relaxations**
   The given problem is formulated as a mathematical program which can be relaxed such that an optimal solution can be viewed as a metric. Rounding techniques based on embeddings can give rise to approximate solutions.

3. **Problems on Metrics**
   Those are the problems in which metrics are the objects of study. For example, given an arbitrary metric, the goal is to find a tree metric that is closest (in some sense) to it. This has applications in building evolutionary trees in computational molecular biology.

In the following we give examples for the first two themes and show how solutions can be obtained via metric methods.

2.1 Metric Data

Suppose we wanted to solve the *traveling salesman problem (TSP)* for the 5-point metric given by the matrix in Figure 1. (This is clearly a toy example!) We can run over all the possible tours, and output the cheapest; or we can use the (isometric) embedding of the metric into the tree (Figure 2), and the fact that TSP can be solved optimally on trees to get a solution.

In a similar manner non-isometric embeddings give rise to approximation algorithms. To continue the toy TSP example, if we could embed the given metric into a tree with distortion D, then the optimal solution to the TSP on this tree would be within D of the optimal TSP tour on the input metric; hence if $D < 1.5$, we would get an improved approximation algorithm. We will see more compelling algorithmic applications of low distortion embeddings later in the course.

2.1.1 Furthest points in ℓ_1

A more substantial example is the following: suppose we are given a set X of n points in ℓ_1^k (with $n \gg 2^k$), and we want to figure out the *furthest pair* of points in this set. Clearly, a brute-force algorithm can do this in $O(kn^2)$ time; we now show how to do this in much less time using an isometric embedding from ℓ_1^k to $\ell_\infty^{2^k}$, where it is much easier to solve the furthest pair problem.

The Embedding: Consider $x \in \ell_1^k$. Then $\|x\|_1 = \sum_i |x_i| = \sum_i \operatorname{sgn}(x_i) \cdot x_i \geq \sum_i y_i \cdot x_i$ for all vectors $y \in \{-1, 1\}^k$. This means we can write $\|x\|_1$ as

$$\|x\|_1 = \max\{ \langle x, y \rangle \mid y \in \{-1, +1\}^k \}. \tag{1}$$

This immediately suggests the following mapping: we create a coordinate $f_y : X \to \mathbb{R}$ for each vector $y \in \{-1,1\}^k$, with $f_y(z) = \langle z, y\rangle$. The final map f is simply obtained by concatenating these coordinates together, i.e., $f = \oplus f_y$. That this is an isometric embedding follows from $\|f(u) - f(v)\|_\infty = \|f(u-v)\|_\infty = \max_y\{\langle u-v, y\rangle\} = \|u-v\|_1$, where we used the linearity of the map and Equation 1.

Solving the transformed problem: We claim that the furthest pair of points in $\ell_\infty^{k'}$ must be the furthest pair of points when projected down onto one of the k' dimensions. Indeed, the distance between the furthest pair in the set S is

$$\begin{aligned}\max_{u,v\in S}\|u-v\|_\infty &= \max_{u,v\in S}\max_{i=1}^{k'}|u_i - v_i| \\ &= \max_{i=1}^{k'}\max_{x,y\in S}|u_i - v_i| \\ &= \max_{i=1}^{k'} \text{ furthest pair along the } i\text{-th coordinate} .\end{aligned}$$

However, the problem of finding the furthest pair along any coordinate can be solved by finding the largest and the smallest value in that coordinate, and hence takes $2n$ time. Doing this for all the k' coordinates takes $O(k'n) = O(n\,2^k)$ time.

2.1.2 The Universal Space: ℓ_∞

We just saw that the metric space ℓ_1^k isometrically embeds into $\ell_\infty^{2^k}$. In fact, a stronger result can be shown: *any* metric space can be embedded isometrically into ℓ_∞. We will show the finite version of this:

Theorem 2.1 (Fréchet) *Any n point metric space (X, d) can be embedded into ℓ_∞.*

Proof. For each point $x \in X$, let us define a coordinate $f_x : X \to \mathbb{R}_+$ thus: $f_x(u) = d(x, u)$. We claim this is an isometric embedding of d into ℓ_∞. Indeed,

$$\begin{aligned}\|f(u) - f(v)\|_\infty &= \max_{x\in X}|f_x(u) - f_x(v)| \\ &= \max_{x\in X}|d(x,u) - d(x,v)| \qquad (2) \\ &\le d(u,v) \ .\end{aligned}$$

However, the value of Expression 2 is also *at least* $d(u, v)$ (obtained by setting $x = u$), and hence the embedding is isometric. ■

Exercise 2.2 *Show that the number of coordinates can be reduced to $n-1$. Show that, for any tree metric, the number can be reduced to $O(\log n)$.*

In contrast, other metric spaces, say ℓ_2, are not universal in this sense. E.g., consider the metric generated by the 3-star $K_{1,3}$. A simple argument shows that this cannot be embedded into ℓ_2 isometrically, regardless of the number of dimensions that are used.

2.2 Metric Relaxations

We illustrate how the method of metric relaxation can be used to solve problems involving cuts.

2.2.1 Metrics and Cuts

Suppose V is some finite set. For $S \subseteq V$, let $\partial(S) := \{\{i,j\} : i \neq j \in V, |S \cap \{i,j\}| = 1\}$.

Definition 2.3 *A metric d on V is an* elementary cut metric*, if there exists a subset $S \subseteq V$ such that for all $i, j \in V$,*

$$d_{ij} = \begin{cases} 1, & \{i,j\} \in \partial S; \\ 0, & \textit{otherwise.} \end{cases}$$

Such a metric will often be denoted by δ_S.

Definition 2.4 *A metric d on V is a* cut metric *if there exists $y : 2^V \to \mathbb{R}_+$ such that for all $i, j \in V$,*

$$d_{ij} = \sum_{S \subseteq V : \{i,j\} \in \partial S} y(S).$$

I.e., the metric d is a cut metric exactly when it is a non-negative linear combination $\sum y(S)\delta_S$ of elementary cut metrics.

2.2.2 Solving Min-Cut via Metric Relaxation

Given a connected undirected graph $G = (V, E)$, a capacity function $c : E \to \mathbb{R}_+$ on edges, and two distinct vertices s and t, the s-t Min-Cut Problem asks for a minimum capacity cut that separates s and t. In the language of cut metrics, the capacity of an s-t minimum cut is:

$$\min\{\sum_{e \in E} c_e d_e : d \text{ is an elementary cut metric such that } d_{st} = 1\}$$

Since it is unclear how to write the above expression as a polynomial sized linear program, we relax the condition and only require d to be a metric such that $d_{st} \geq 1$. I.e.,

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e d_e & & (3) \\ \text{subject to} \quad & d_{ij} \leq d_{ik} + d_{kj}, & & \forall i, j, k \\ & d_{st} \geq 1 & & \\ & d_{ij} \geq 0, & & \forall i, j. \end{aligned}$$

Suppose the metric d^* is an optimal solution to (3). Since d^* may not be a cut metric in general, we need to round the solution in order to obtain a cut. To this end, let us order the vertices in V into *level sets* based on their distances from s in d^* (Figure 4). Let $x_0 < x_1 < \ldots < x_l$ be the distinct distances in the set $\{d^*(s, v) \mid v \in V\}$ in increasing order. Clearly, the smallest distance $x_0 = 0$ (corresponding to $d^*(s, s)$), and $x_l \geq d^*(s, t)$.

For each $1 \leq j \leq l$, define $V_j = \{v \in V \mid d^*(s, v) \leq x_j\}$ be the vertices in the first j levels, and $E_j = \partial(V_j)$ be the edges that leave V_j. Also, define $y_j = x_j - x_{j-1}$, and $C_j = \sum_{e \in E_j} c_e$.

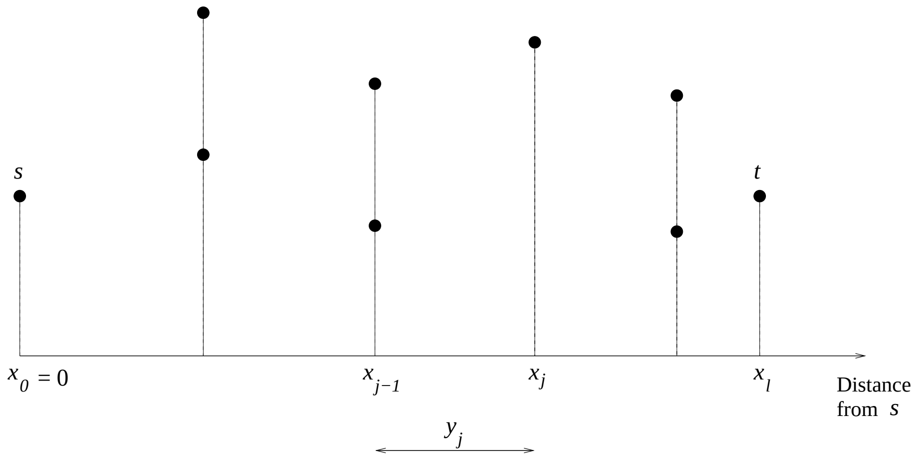

Figure 4: Ordering vertices into level sets

Lemma 2.5 *For each edge $e \in E$, the length given to e by (3) is*

$$d_e^* \geq \sum_{j:e \in E_j} y_j.$$

Proof. For the edge $e = (u, v) \in E$, the length $d_e^* \geq |d^*(s,u) - d^*(s,v)|$ by the triangle inequality. Furthermore, for each vertex $u \in V$, it holds that $d^*(s,u) = \sum_{j:\{s,u\} \in E_j} y_j$. Combining the two, we get that $d_e^* \geq \sum_{j:e \in E_j} y_j$. ■

Theorem 2.6 *Let $C^* = \min_{j=1}^{l} C_j$ be attained at $j = j_0$. Then the cut E_{j_0} is an minimum s-t cut.*

Proof. Let Z^* be the capacity of a minimum cut. Since Z^* is the value of an optimal integral solution to (3), it can be no smaller than the value of the optimal fractional solution, and hence $Z^* \geq \sum_e c_e d_e^*$. Now using Lemma 2.5, we get

$$\sum_{e \in E} c_e d_e^* \geq \sum_{e \in E} c_e \sum_{j:e \in E_j} y_j = \sum_{j=1}^{l} \sum_{e \in E_j} c_e y_j = \sum_{j=1}^{l} C_j y_j,$$

simply by changing the order of summation, and usign the definition of C_j. However, since $C^* = \min_j C_j$, we get that

$$\sum_{j=1}^{l} C_j y_j \geq \sum_{j=1}^{l} C^* y_j = C^*(x_l - x_0) \geq C^* d^*(s,t) \geq C^*,$$

the final inequality using that $d^*(s,t) \geq 1$. This shows that $Z^* \geq C^*$, which proves the result. ■

2.2.3 An ℓ_1-embeddings view

As an aside, the proof can also be looked at as embedding the distance d^* into the *cut* metric $d' = \sum_j y_j \delta_{V_j}$, which is the same as $d'(x,y) = d^*(s,x) - d^*(s,y)$. (Check that these are two equivalent definitions!)

Since $d'(x,y) \leq d^*(x,y)$ for all x, y, we get that $\sum_e c_e d'_e \leq \sum_e c_e d_e$. Now picking a random value $j \in \{1, \ldots, l\}$ with probability $y_j / (\sum y_j)$ and taking the cut $E_j = \partial(V_j)$ ensures that the expected value of the cut is:

$$\mathbf{E}\left[\sum_e c_e \delta_{V_j}(e)\right] = \frac{1}{\sum_j y_j} \sum_e c_e d'_e \leq \sum_e c_e d_e \leq Z^*,$$

using the fact that $\sum y_j \geq d^*(s,t) \geq 1$. Since the expected value of the random is cut is at most Z^*, there must be some cut among the E_j with value at most Z^*. Since we return the smallest such cut E_{j_0}, we get that $C^* = Z^*$, proving the result.

3 The main results

The main results in low-distortion embeddings can be classified into three main groups based on the basic techniques involved in obtaining them.

3.1 Fréchet-type embeddings

These results embed a metric (X, d) into ℓ_p spaces, and fall into the following general framework. Let us define a mapping $f_S : X \to \mathbb{R}$ for each subset of the vertices $S \subseteq X$ thus:

$$f_S(x) = d(x, S) = \min_{s \in S} d(x, s).$$

A *Fréchet-type embedding* is a map $f : X \to \mathbb{R}^k$, each of whose coordinates is a scaled one of these mappings above. Formally, we will denote this as:

$$f(x) = \bigoplus_{S \subseteq X} \beta_S \, f_S,$$

for $\beta_S \in \mathbb{R}$. Indeed, to get Fréchet's mapping of metrics into ℓ_∞ from Section 3.1, we can set $\beta_S = 1 \iff |S| = 1$. The main result in this area was proved by Bourgain (1985), who showed the following theorem:

Theorem 3.1 (Bourgain (1985)) *Given a metric (X, d), the map f obtained by setting*

$$\beta_S = \frac{1}{|S| \binom{n}{|S|}}$$

is an $O(\log n)$-distortion embedding of d into ℓ_1.

This result is tight for ℓ_1, as was shown by Linial, London and Rabinovich (1995). They extended a theorem of Leighton and Rao (1989) to show that constant-degree expanders require a distortion of $\Omega(\log n)$ to embed into ℓ_1.

In 1995, Matoušek sharpened Theorem 3.1 to show that a suitable choice of β_S allows us to embed any n-point metric into ℓ_p with distortion $O((\log n)/p)$. Furthermore, he showed a matching lower bound of $\Omega((\log n)/p)$ on the distortion into ℓ_p, using the same constant-degree expanders.

Theorem 3.2 *Given a metric and $p \in [1, \infty)$, one can embed d into ℓ_p^k with $O((\log n)/p)$ distortion.*

The embeddings can be modified by random sampling ideas to use a small number of dimensions:

Theorem 3.3 *Given a metric and $p \in [1, \infty)$, one can embed d into ℓ_p^k with $k = O(\log^2 n)$ dimensions and $O(\log n)$ distortion.*

3.2 Flattening Lemmas

Perhaps the most surprising result is the last one, which reduces the dimensions required for representing n-point Euclidean point sets, if inter-point distances are allowed to be slightly distorted. The theorem is exceedingly simple to state:

Theorem 3.4 (Johnson and Lindenstrauss, 1984) *Given $X \subseteq \mathbb{R}^k$, there is an embedding $f : X \to \mathbb{R}^{O(\log n/\epsilon^2)}$ with distortion $(1 + \epsilon)$.*

In fact, the proof of this result also shows that such a map can be obtained (with high probability) by just projecting the point set on a *random* $k' = O(\log n/\epsilon^2)$-dimensional subspace, and then scaling up the resulting distances by $\sqrt{k/k'}$. (This should be contrasted with the observation that we require $n - 1$ dimensions to embed the n-point uniform metric into ℓ_2.)

We will see some (fairly simple) proofs of this theorem later in the course, as well as many applications of this powerful theorem; let us allude to two of the applications here:

3.3 Random Tree Embeddings

Since many algorithmic problems are simple on trees, one would want to embed a given metric into a tree with low distortion. However, this is not possible, and one can show that the n-cycle requires distortion $\Omega(n)$ to embed into any tree. However, much better results can be obtained if one wants to embed the given metric (X, d) into a *distribution of trees*, instead of one tree, so that the *expected distances* are preserved.

Formally, we say that the metric (X, d) α-probabilistically embeds into a distribution $\mathcal{D}$ of trees if

- Each tree $T = (V_T, E_T)$ in the support of the distribution $\mathcal{D}$ contains the points of the metric; i.e., $X \subseteq V_T$. Furthermore, the distances in T *dominate* those in d; i.e., $d_T(x, y) \geq d(x, y)$ for all $x, y \in X$.

- Given a pair of vertices $x, y \in X$, the expected distance is not too much larger than $d(x, y)$, i.e.,
$$\mathbf{E}[\, d_T(x, y)\,] \le \alpha\, d(x, y).$$

The added power of randomization can be seen by taking the n-cycle with unit length edges: the distribution we want is obtained by picking one of the edges uniformly at random and deleting it. It can be verified that this gives us a $2(1 - \frac{1}{n})$-probabilistic embedding of the cycle into its subtrees.

This concept has been widely studied in many papers, and culminated in the following result due to Fakcharoenphol, Rao and Talwar (2003):

Theorem 3.5 *Any n-point metric $O(\log n)$-probabilistically embeds into a distribution of trees; furthermore, samples from this distribution can be generated in polynomial time.*

Embeddings into random trees (as we shall refer to them) have enjoyed very wide applicability, and we will see examples later in the course. Essentially, any intractable problem on a metric space (that has a cost function linear in the distances) can be now solved on a tree instead, and Theorem 3.5 guarantees that we lose only $O(\log n)$ in the cost (in expectation).

References

[Bou85] Jean Bourgain. On Lipschitz embeddings of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985.

[Das99] Sanjoy Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 634–644, 1999.

[FRT03] Jittat Fakcharoenphol, Satish B. Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*, pages 448–455, 2003.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. Also in *Proc. 35th FOCS*, 1994, pp. 577–591.

[LR99] Frank Thomson Leighton and Satish B. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[Mat97] Jiří Matoušek. On embedding expander graphs into ℓ_p spaces. *Israel Journal of Mathematics*, 102:189–197, 1997.

[Mat02] Jiří Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer, New York, 2002.