14 Optimal embedding into ℓ_2 via SDP

In the previous section, we saw that any metric space embeds into the Euclidean space with $O(\log n)$ distortion. One can ask the following questions.

Question 1: Given a metric (X, d), can we check in polynomial time if $(X, d) \overset{1}{\hookrightarrow} \ell_2$?

Furthermore, Theorem **??** implies that $D \le O(\log n)$, but it could be much smaller for a specific metric space. This motivates the question:

Question 2: If a metric space (X, d) embeds into ℓ_2 with distortion D, i.e. $(X, d) \overset{D}{\hookrightarrow} \ell_2$, can we compute D in polynomial time? Failing that, can we approximate D?

Answer 1: Let $X = \{x_1, x_2, \ldots, x_n\}$. Let f denote the mapping from X to ℓ_2. Let $f(x_i) = v_i$. By suitable change of coordinates, we can assume that $f(x_1) = v_1 = \vec{0}$. Now answering Question 1 in the affirmative is equivalent to checking if

$$
\begin{aligned}
& & \|v_i - v_j\|^2 &= d_{ij}^2 & \forall i, j \\
&\Leftrightarrow & \|v_i\|^2 + \|v_j\|^2 - 2\langle v_i, v_j\rangle &= d_{ij}^2 \\
&\Leftrightarrow & \langle v_i, v_j\rangle &= \tfrac{1}{2}(d_{1i}^2 + d_{1j}^2 - d_{ij}^2),
\end{aligned}
$$

where the last line follows from the fact that, for all i, $\|v_i\|^2 = \|v_i - v_1\|^2 = d_{1i}^2$.

Imagine a matrix A, with $A_{ij} = \frac{1}{2}(d_{1i}^2 + d_{1j}^2 - d_{ij}^2)$; the above argument implies that we want $A_{ij} = \langle v_i, v_j\rangle$. However, this is equivalent to asking whether A can be written as $B^T B$ for matrix B defined as

$$
B = \begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & & \vdots \end{pmatrix} .
$$

It is known that such a decomposition exists if and only if A is a *positive semi-definite matrix*, and that it can be obtained in polynomial time; see, e.g., page 295 of Grötschel, Lovász, and Schrijver (1988).

Answer 2: The basic ideas in the previous answer can be used to answer Question 2 as well. Formally, we will write a *Semi-Definite Program* (SDP) to obtain the value of the best distortion. One way of writing an SDP is as a linear program, where the "variables" are inner products of vectors.

Example: Here is an example of a semi-definite program for the MAX-CUT problem:

$$
\begin{aligned}
\min \quad & \sum_{i,j} w_{ij}\left(\frac{1 - \langle v_i, v_j\rangle}{2}\right) \\
\text{s.t.} \quad & \langle v_i, v_i\rangle = 1 & \forall\, i \\
& \langle v_i, v_j\rangle + \langle v_j, v_k\rangle + \langle v_k, v_i\rangle \ge -1 & \forall\, i, j, k
\end{aligned}
$$

$\blacksquare$

In our problem, we want to minimize D, subject to the constraints that all the pairwise distances are distorted by atmost D. To formulate it as an SDP, we change D to a vector whose norm will denote the distortion.

$$\begin{aligned}
\min \quad & \langle \vec{D}, \vec{D} \rangle \\
\text{s.t.} \quad & d_{ij}^2 \leq \langle v_i, v_i \rangle + \langle v_j, v_j \rangle - 2\langle v_i, v_j \rangle \leq \langle \vec{D}, \vec{D} \rangle d_{ij}^2 && \forall\, i, j \\
& d_{1i}^2 \leq \langle v_i, v_i \rangle \leq \langle \vec{D}, \vec{D} \rangle d_{1i}^2 && \forall\, i
\end{aligned}$$

It is a well known fact that, there is a polynomial time algorithm to solve an SDP to any desired precision. Thus we have an algorithm which finds the minimum possible distortion for embedding an arbitrary metric space (X, d) into the Euclidean space.

14.1 Problems in ℓ_1-embeddability

The analogous questions can also be asked about embeddability into ℓ_1:

Question 1: Given a metric (X, d), can we check in polynomial time if $(X, d) \overset{1}{\hookrightarrow} \ell_1$, i.e., if it embeds isometrically into ℓ_1?

However, it is known that this problem is NP-hard, as was shown by Karzanov (1985) via a reduction from Max-Cut. On the other hand, little is known about the following problem:

Question 2: Given a metric (X, d), approximate the value of the best distortion D such that $(X, d) \overset{D}{\hookrightarrow} \ell_1$.

Since $D \geq 1$ for all metrics, Bourgain's theorem implies a $O(\log n)$ approximation algorithm, and this is the best known. In fact, even if we are given a metric that is *isometrically* ℓ_1 embeddable, we have no algorithms that can guarantee a distortion of $o(\log n)$.

14.2 SDP-relaxation for sparsest cut

The above techniques that we used to find ℓ_2-embeddings can be also used for the sparsest cut problem. In previous lectures we used an LP-relaxation for sparsest cut:

$$\begin{aligned}
\text{minimize} \quad & \sum_{x,y} \operatorname{cap}(x, y) \cdot d(x, y) \\
\text{subject to} \quad & \sum_{x,y} \operatorname{dem}(x, y) \cdot d(x, y) = 1 \\
& d(x, y) \leq \sum_{(u,v) \in P} d(u, v) \qquad \forall \text{ path } P \text{ between } x \text{ and } y \\
& d(x, y) \geq 0
\end{aligned} \tag{LP 1}$$

The LP asks for a metric that creates a large total distance (at least 1) between demand-pairs (weighted with the respective demand), while keeping the weight of the metric (sum of all edge lengths) small. The integrality gap of this LP is $\Omega(\log n)$ for expander graphs and therefore one cannot hope to obtain a better than $\Theta(\log n)$-approximation by rounding this LP.

However there is a different relaxation of sparsest cut. For the LP, we viewed the integral solution as a metric that creates a distance of one between points that are on opposite sides of the cut. Instead, we can view a cut in a graph $G = (V, E)$ as a function that assigns a value of of either 1 or -1 to the nodes in V with the understanding that nodes that are assigned the same value are on the same side of the cut. The value of the cut is then $\frac{1}{2}\sum_{(x,y)\in E} \text{cap}(x,y) \cdot |f(x) - f(y)|$, or $\frac{1}{4}\sum_{(x,y)\in E} \text{cap}(x,y) \cdot (f(x) - f(y))^2$. With this in mind we get the following program for sparsest cut (where we assume unit edge-capacities and uniform demands).

$$\begin{aligned} \text{minimize} \quad & \frac{1}{4}\sum_{x,y} \|v_x - v_y\|^2 \\ \text{subject to} \quad & \sum_{x,y} \|v_x - v_y\|^2 = 1 \\ & \|v_x - v_y\|^2 \leq \|v_x - v_z\|^2 + \|v_z - v_y\|^2 \\ & v_x \in \{-1, 1\} \end{aligned} \tag{SDP 1}$$

By relaxing the v_x to be vectors in n-dimensions and omitting the integrality constraint one gets an SDP-relaxation to sparsest cut. In the following we deal with a problem that is closely related to the uniform sparset cut problem. Find a cut in the graph that minimizes the number of edges in the cut subject to the constraint that both sides contain at least $c \cdot n$ vertices. This problem is known as the c-balanced cut problem. The following is an SDP-relaxation to the c-balanced cut problem.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{4}\sum_{x,y} \|v_x - v_y\|^2 \\ \text{subject to} \quad & \|v_x\|_2 = 1 \\ & \|v_x - v_y\|^2 \leq \|v_x - v_z\|^2 + \|v_z - v_y\|^2 \\ & \sum_{x<y} \|v_x - v_y\|^2 \geq c(1-c)n^2 \end{aligned} \tag{SDP 2}$$

There is an interesting connection of these relaxations to so-called spectral graph partitioning methods. Let L denote the Laplacian of the graph G this means the $|V| \times |V|$-matrix where the entry $L_{x,y} = -1$ if there is an edge between x and y, and a diagonal element $L_{x,x}$ is equal to the degree of the node x in G. Let $\vec{v}$ denote a vector from $\{-1, 1\}^n$ such that v_x is the value assigned to node x by the cut-function. Then the capacity of the cut can be expressed as $\vec{v}^t \cdot L \cdot \vec{v}$ (up to the scaling factor 1/4). Simply searching for $\vec{v}$ that minimizes this objective causes a problem as the number of solutions is not bounded since the minimum is invariant under translations. Therefore one adds the constraint that the vector $\vec{v}$ is balanced this means the average over all entries is 0 (in the integral setting that says that both sides of the cut contain half of the vertices). $\vec{v} \cdot \vec{e} = 0$, where $\vec{e} = (1, \ldots, 1)^t$. However, the problem now still has the trivial solution $\vec{v} = 0$. Adding the constraint $\vec{v}^t\vec{v} = 1$ gets rid of this solution.[1]

Note that the solution that minimizes $\vec{v}^t \cdot L \cdot \vec{v}$ has to be an eigenvector of L. Further observe that the vector e is an eigenvector of L with eigenvalue 0. Therefore the solution of the system that we just defined is the second-largest eigenvalue of L.

The SDP-approach that we use is an extension of the eigenvalue approach where now the elements of the vectors $\vec{v}$ are now n-dimensional vectors.

14.3 Approximation to c-balanced separators

In the following lecture we prove a pseudo approximation to c-balanced separators. The relaxation SDP 2 outputs points in Euclidean space for which the squared distances obey the triangle inequality. This gives a squared ℓ^2-metric on this point set. The following theorem will be proved in the next lecture.

Theorem 14.1 (ARV04) *Let (X, ρ) be an n-point squared ℓ_2-metric over vectors from the unit sphere and let $\sum_{x,y \in X} \rho(x, y) \geq c \cdot n^2 > 0$. There exist subsets $L, R \subseteq X$ with $|L|, |R| = \Omega(c \cdot n)$ and $\rho(L, R) \geq \Omega(\frac{1}{\sqrt{\log n}})$.*

We can use this theorem to get a pseudo-approximation to c-balanced cut. A pseudo-approximation is a solution that cuts a small number of edges (comparable to the optimum solution) but slightly violates the balance constraint (by a constant factor), i.e., it only guarantees that $c' \cdot n$ vertices are in the smaller side of the cut, where in general c' is smaller than c but usually larger than say $c/100$. We get this approximation guarantee by considering level cuts around the set L from the above partitioning (a level cut at distance r contains all edges that are at distance r from L). We output the minimum level cut for $r \in \{0, 1/\sqrt{\log n}\}$. The capacity of this cut is at most $\sqrt{\log n}$ times the total weight of all edges. Furthermore, each of these cuts separates the sets L and R and hence both sides contain a linear number of vertices.

References

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings, and graph partitionings. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, pages 222–231, 2004.

[1] Note that any normalization would do here. Perhaps the normalization $\vec{v}^t\vec{v} = n$ would be more appropriate when considering the integral solution where $\vec{v}$ is a vector from $\{-1, 1\}^n$.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization.* Springer, New York, 1988.

[HIL98] Johan Håstad, Lars Ivansson, and Jens Lagergren. Fitting points on the real line and its application to RH mapping. In *Proceedings of the 6th European Symposium on Algorithms (ESA)*, pages 465–476, 1998.

[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.