Chapter 1

Introduction: Why are representations interesting?

To represent a graph geometrically is a natural goal in itself, but in addition it is an important tool in the study of various graph properties, including their algorithmic aspects. We describe three examples of increasing complexity, and then discuss some goals of this book.

1. Disjoint paths. Let us start with a very simplified example. Suppose that we have a 2-connected graph G and two specified nodes s and t. Two "requests" come in for two nodes x and y, and we have to find two disjoint paths connecting s and t to x and y (it does not matter which of x and y will be connected to s). This can be computed by one of zillions of flow or connectivity algorithms in reasonable time.

Now suppose that we have to compute such paths for many requests $\{x, y\}$. Do we have to repeat the computation each time? We can do much better if we use the following theorem: *Given a* 2*-connected graph and two specified nodes s and t, we can order all nodes so that s is first, t is last, and every other node v has a neighbor that comes earlier as well as a neighbor that comes later.* Such an ordering is called an *s-t numbering.*

Once we know an s-t numbering, and a request $\{x, y\}$ comes in, it is trivial to find two disjoint paths: let (say) x precede y in the ordering, then we can move from x to an earlier neighbor x', then to an even earlier neighbor x'' of x' etc. until we reach s. Similarly, we can move from y to a later neighbor y', then to an even later neighbor y'' of y' etc. until we reach t. This way we trace out two paths as requested.

The ordering can be thought of as representing the nodes of G by points on the line, and the easy procedure to find the two paths uses this geometric representation. Of course, one-dimensional geometry is not "really" geometry, and we better give a higher-dimensional example.

2. Monotone paths. The following was proved in [**?**]: Let $V = \{1, \dots, n\}$, and let

$f:\binom{V}{2}\to\mathbb{R}$. Let us call a sequence $1\le k_0<k_1<\dots<k_r\le n$ such that

$$f(k_0,k_1)\le f(k_1,k_2)\le\dots\le f(k_{r-1},k_r),$$

a *monotone increasing path of length* r. We define monotone decreasing paths analogously. The result of Chvátal and Komlós asserts that it if the maximum length of a monotone increasing path is p, and the maximum length of a monotone decreasing path is q, then $n\le\binom{p+q}{p}$. (The result's background is in combinatorial geometry, for which we refer to the original paper.)

The proof uses a geometric representation of the data. For $1\le u<v\le n$, let $g(u,v)$ be the maximum length of a monotone increasing path starting with the pair (u,v), and let $h(u,v)$ be the maximum length of a monotone decreasing such path. Thus each pair $u<v$ is represented by a point $(g(u,v),h(u,v))$ in the plane. By hypothesis, these points all belong to the rectangle $[1,p]\times[1,q]$.

For $1\le v<n$, let $P_v=\{(g(v,w),h(v,w)):\ v<w\le n\}$, and let $\overline{P}_v$ consist of all maximal elements of P_v (with respect to the partial ordering $(x,y)\le(x',y')$ iff $x\le x'$ and $y\le y'$). We claim that the sets $\overline{P}_v$ are different. For suppose that $\overline{P}_v=\overline{P}_w$, where $v<w$. Then $(g(v,w),h(v,w))\in P_v$, so by the definition of $\overline{P}_v$, there is a point $(x,y)\in\overline{P}_v$ such that $g(v,w)\le x$ and $h(v,w)\le y$. Since $\overline{P}_v=\overline{P}_w$, there is an integer s, $w<s\le n$, such that $g(w,s)=x$ and $h(w,s)=y$. Without loss of generality, we may assume that $f(v,w)<f(w,s)$. But then we get a monotone increasing path if we start with v, followed by a monotone increasing path of length x starting with (w,s). This shows that $g(v,w)\ge 1+x$, contradicting the definition of x.

All that remains is to count how many sets come into consideration as $\overline{P}_v$. Clearly every such set is of the form $\{(x_1,y_1),\dots,(x_k,y_k)\}$, where $1\le x_1<\dots<x_k\le p$ and $q\ge y_1>\dots>y_k\ge 1$. The number of ways to choose these values is

$$\sum_{k\ge0}\binom{p}{k}\binom{q}{k}=\sum_{k\ge0}\binom{p}{k}\binom{q}{q-k}=\binom{p+q}{p},$$

which completes the proof.

Perhaps this example is still not completely convincing: we use two dimensions, but only ordering of the coordinates, no “true” geometry. So we continue with an application of a 3-dimensional geometric representation with more geometric content. (Relax, there will be no fourth example using 4-dimensional geometry.)

3. Shannon capacity. The following problem in information theory was raised by Claude Shannon, and it motivated the introduction of orthogonal representations [**?**] and several of the results to be discussed in this book.

Consider a noisy channel through which we are sending messages composed of a finite alphabet V. There is an output alphabet U, and each $v\in V$, when transmitted through the channel, can come out as any element in a set $U_v \subseteq U$. Usually there is a probability distribution specified on each set U_v, telling us the probability with which v produces a given $u \in U_v$, but for the problem we want to discuss, these probabilities don't matter. As a matter of fact, the output alphabet will play no role either, except to tell us which pairs of input characters can be confused: those pairs (v, v') for which $U_v \cap U_{v'} \neq \emptyset$.

We want to select as many words of length k as possible so that no two can possibly be confused. As we shall see, the number of words we can select grows as Θ^k for some $\Theta \geq 1$, which is called the *Shannon zero-error capacity* of the channel. A simple and natural way to create such a set of words is to pick a non-confusable subset of the alphabet, and use only those words composed from this set. So if we have α non-confusable characters in our alphabet, then we can create α^k non-confusable messages of length k. But, as we shall see, making use of other characters in the alphabet we can create more! How much more, is the issue in this discussion.

One way to model the problem is as follows: We consider V as the set of nodes of a graph, and connect two of them by an edge if they can be confused. This way we obtain a graph G, which we call the *confusion graph* of the alphabet. The maximum number of non-confusable messages of length 1 is the maximum number of nonadjacent nodes (the maximum size of a stable set) in the graph G, which we denote by $\alpha(G)$.

Let us look at two simple examples (Figure 1.1.

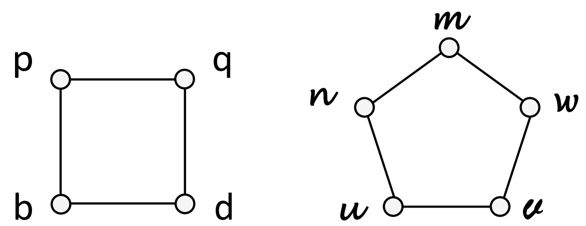

Figure 1.1: Two confusion graphs. In the alphabet $\{p, q, b, d\}$ two letters that are related by a reflection in a horizontal or vertical line are confusable, but not if they are related by two such reflection. The confusability graph of the alphabet $\{m, n, u, v, w\}$ is only convincing a little in handwriting, but this graph plays an important role in this book.

Example 1.0.1 Let us consider the simple alphabet (p, q, d, b), where the pairs $\{p, q\}$, $\{q, d\}$, $\{d, b\}$ and $\{b, p\}$ are confusable (Figure 1.1, left). We can just keep p and d (which are not confusable), which allows us 2^k non-confusable messages of length k. On the other hand, if we use a word, then all the 2^k words obtained from it by replacing some occurrences of p and q by the other, as well as some occurrences of b and d by the other, are excluded. Hence the number of messages we can use is at most $4^k/2^k = 2^k$. ♦

Example 1.0.2 (5-cycle) If we switch to alphabets with 5 characters, then we get a much more difficult problem. Let $V = \{m, n, u, v, w\}$ be our alphabet, with confusable pairs

$\{m,n\}$, $\{n,u\}$, $\{u,v\}$, $\{v,w\}$ and $\{w,m\}$ (Figure 1.1, right; we refer to this example as the "pentagon"). Among any three characters there are two that can be confused, so we have only two non-confusable characters, and restricting the alphabet to two such characters (say, m and v), we can for 2^k non-confusable messages of length k.

But we can do better: the following 5 messages of length two are non-confusable: mm, nu, uw, vn and wv. This takes some checking: for example, mm and nu cannot be confused, because their second characters, m and u, cannot be confused. If k is even, then we can construct $5^{k/2}$ non-confusable messages, by concatenating any $k/2$ of the above 5. This number grows like $(\sqrt{5})^k \approx 2.236^k$ instead of 2^k, a substantial gain! ♦

Can we do better by looking at longer messages (say, messages of length 10), and by some *ad hoc* method finding among them more that 5^5 non-confusable messages? We are going to show that we cannot, the use of 5 messages of length 2 is optimal.

The trick is to represent the alphabet in a different way. Let us assign to each character $i \in V$ a vector $\mathbf{u}_i$ in some euclidean space $\mathbb{R}^d$. If two characters are non-confusable, then we represent them by orthogonal vectors. Figure 1.2 shows such an assignment of vectors to the 5-element alphabet in Example 1.0.2.

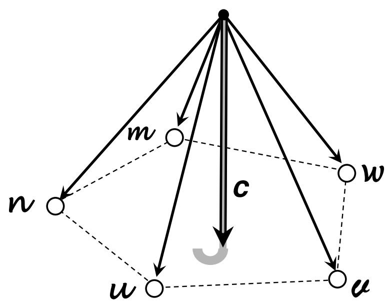

Figure 1.2: An umbrella representing the pentagon.

If a subset of characters S is non-confusable, then the vectors $\mathbf{u}_i$ $(i \in S)$ are mutually orthogonal unit vectors, and hence for every unit vector $\mathbf{c}$,

$$\sum_{i\in S} (\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 \le 1.$$

Hence $|S| \min_{i\in S}(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 \le 1$, or

$$|S| \le \max_{i\in S} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2} \le \max_{i\in V} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2}.$$

So if we find a representation $\mathbf{u}$ and a unit vector $\mathbf{c}$ for which the squared products $(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2$ are all large (which means that the angels $\measuredangle(\mathbf{c},\mathbf{u}_i)$ are all small), then we get a good upper bound on $|S|$. Let us denote the best bound we can get this way by ϑ. We call the vector $\mathbf{c}$ the *handle*.

For the alphabet in Example 1.0.2 (the pentagon), we use the vectors in Figure 1.2. To describe these, consider an umbrella in $\mathbb{R}^3$ with 5 ribs of unit length. Open it up to the point when non-consecutive ribs are orthogonal. This way we get 5 unit vectors $\mathbf{u}_m, \mathbf{u}_n, \mathbf{u}_u, \mathbf{u}_v, \mathbf{u}_w$, assigned to the nodes of the pentagon so that each $\mathbf{u}_i$ forms the same angle with the "handle" $\mathbf{c}$ and any two non-adjacent nodes are labeled with orthogonal vectors. With some effort, one can compute that $(\mathbf{c}^\mathsf{T}\mathbf{u}_i)^2 = 1/\sqrt{5}$ for every i, and so $\vartheta = \sqrt{5}$, and we get that $|S| \le \sqrt{5}$ for every non-confusable set S. Since $|S|$ is an integer, this implies that $|S| \le 2$.

This is ridiculously much work to conclude that the 5-cycle does not contain 3 non-adjacent nodes! But the vector representation is very useful for the handling of longer messages. We define the *tensor product* of two vectors $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$ and $\mathbf{v} = (v_1, \dots, v_m) \in \mathbb{R}^m$ as the vector

$$\mathbf{u} \circ \mathbf{v} = (u_1v_1, \dots, u_1v_m, u_2v_1, \dots, u_2v_m, \dots, u_nv_1, \dots, u_nv_m)^\mathsf{T} \in \mathbb{R}^{nm}.$$

It is easy to see that $|\mathbf{u} \circ \mathbf{v}| = |\mathbf{u}|\,|\mathbf{v}|$, and (more generally) if $\mathbf{u}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v}, \mathbf{y} \in \mathbb{R}^m$, then $(\mathbf{u} \circ \mathbf{v})^\mathsf{T}(\mathbf{x} \circ \mathbf{y}) = (\mathbf{u}^\mathsf{T}\mathbf{x})(\mathbf{v}^\mathsf{T}\mathbf{y})$. For a $k \ge 1$, if we represent a message $i_1 \dots i_k$ by the vector $\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k}$, then non-confusable messages will be represented by orthogonal vectors. Indeed, if $i_1 \dots i_k$ and $j_1 \dots j_k$ are not confusable, then there is at least one subscript r for which i_r and j_r are not confusable, hence $\mathbf{u}_{i_r}^\mathsf{T}\mathbf{u}_{j_r} = 0$, which implies that

$$(\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k})^\mathsf{T}(\mathbf{u}_{j_1} \circ \dots \circ \mathbf{u}_{j_k}) = (\mathbf{u}_{i_1}^\mathsf{T}\mathbf{u}_{j_1}) \dots (\mathbf{u}_{i_k}^\mathsf{T}\mathbf{u}_{j_k}) = 0.$$

As for handle, we use $\mathbf{c} \circ \dots \circ \mathbf{c}$ (k factors), where $\mathbf{c}$ is the optimal handle for single characters. We get that for any set S of non-confusable messages of length k,

$$|S| \le \max_{i_1,\dots,i_k} \frac{1}{((\mathbf{c} \circ \dots \circ \mathbf{c})^\mathsf{T}(\mathbf{u}_{i_1} \circ \dots \circ \mathbf{u}_{i_k}))^2} = \max_{i_1,\dots,i_k} \frac{1}{(\mathbf{c}^\mathsf{T}\mathbf{u}_{i_1})^2 \dots (\mathbf{c}^\mathsf{T}\mathbf{u}_{i_k})^2} = \vartheta^k.$$

In particular, for the pentagon, every set of non-confusable messages of length k has at most $(\sqrt{5})^k$ elements. We have seen that this bound can be attained, at least for even k. Thus we have established that *the Shannon zero-error capacity of the pentagon is* $\sqrt{5}$.

We will return to this topic in Sections **??** and **??**, where the zero-error capacity problem will be discussed for general confusability graphs, both in classical and quantum information theory.

4. Proofs, algorithms and geometry. There are several levels of this interplay between graph problems and geometry.

— Often the aim is to find a way to represent a graph in a "good" way. We refer to Kuratowski's characterization of planar graphs, to its more recent extensions (most notably the work of Robertson, Seymour and Thomas), and to Steinitz's theorem representing 3-connected planar graphs by 3-dimensional polyhedra. Many difficult algorithmic problems in connection with finding these representations have been studied.

— In other cases, graphs come together with a geometric representation, and the issue is to test certain properties, or compute some parameters, that connect the combinatorial and geometric structure. A typical question in this class is rigidity of bar-and-joint frameworks, an area whose study goes back to the work of Cauchy and Maxwell.

— Most interesting are the cases when a good geometric representation of a graph not only gives a useful way of understanding its structure, but it leads to algorithmic solutions of purely graph-theoretic questions that, at least on the surface, do not seem to have anything to do with geometry. The first simple example above illustrates this point: a geometric structure (ordering) provides a good data structure for computing the required paths.

This book will contain many more applications of geometric representation in proofs and algorithms (the list is far from complete): rubber band representations can be used in planarity testing and graph drawing; repulsive springs lead to approximating the maximum cut; coin representations can be used in approximating optimal bisection; nullspace representations provide 3-polytopes with specified skeleton graphs; orthogonal representations play a role in algorithms for graph connectivity, graph coloring, finding maximum cliques in perfect graphs, and estimating capacities of channels in information theory; volume-respecting embeddings are used in approximation algorithms for bandwidth.