1 Metric spaces

For completeness, we recall the definition of metric spaces and the notions relating to measures on metric spaces. A metric space is a pair (M, d) where M is a set and d is a function from the Cartesian product $M \times M$ to the non-negative real numbers, such that

- $d(x, x) = 0$ for all $x \in M$,
- $d(x, y) = d(y, x)$ for all $x, y \in M$,
- $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in M$ (the triangle inequality),

Example: edit distance

1.1 Edit distance review

The concept of string-edit distance d_e is based on balancing two different types of edits. The simplest is replacement of a letter. That is, if two strings x and y differ only in the k-th position, then $d_e(x, y) = D_A(x_k, y_k)$ for some metric D_A on the alphabet A.

In general, when there are multiple replacements, string edit distance is based on just summing the effects.

However, string-edit distance also allows a different kind of change as well: insertion and deletion.

For example, we can define $x_{\hat{k}}$ to mean the string x with the k-th entry removed. It might be that $x_{\hat{k}}$ agrees perfectly with the string y, and so we assign $d(x, y) = \delta$ where δ is the deletion penalty.

Similarly, insertions of characters are allowed to determine edit distance. Clearly, if $y = x_{\hat{k}}$, then adding x_k to y at the k-th position yields x.

Again, the effect of multiple insertions/deletions is additive, and this allows strings of different lengths to be compared.

1.2 Edit distance review, continued

The use of both replacements and instertion/deletions to determine edit distance complicates the picture substantially.

The representation of an edit path from x to y using both replacements and insertions/deletions is not unique.

Thus edit distance is defined by taking the minimum over all possible representations.

In general, this will not yield a metric unless constraints on δ and D_A are imposed.

This can be done in a very simple and elegant way by extending the alphabet A and metric D_A to include a "gap" as a character, say "_" (let $\widetilde{A}$ denote the extended alphabet), and by assigning a distance $D_{\widetilde{A}}(x, \_)$ for each character x in the original alphabet.

Theorem 9.4 of [1] tells us that d_e is a metric on strings of letters in A whenever $D_{\widetilde{A}}$ is a metric on the extended alphabet.

1.3 Example: two-letter alphabet

The simplest non-trivial example is an alphabet with two letters, say x and y, when there is only one distance $D_A(x, y)$ that is non-zero.

The requirement that the triangle inequality hold for $D_{\widetilde{A}}$ reduces to three inequalities that can be expressed as

$$|D_{\widetilde{A}}(x, \_) - D_{\widetilde{A}}(y, \_)| \leq D_A(x, y) \leq D_{\widetilde{A}}(x, \_) + D_{\widetilde{A}}(y, \_). \tag{1.1}$$

The left hand inequality derives from the two inequalities

$$\begin{aligned} D_{\widetilde{A}}(x, \_) &\leq D_A(x, y) + D_{\widetilde{A}}(y, \_) \\ D_{\widetilde{A}}(y, \_) &\leq D_A(y, x) + D_{\widetilde{A}}(x, \_) = D_A(x, y) + D_{\widetilde{A}}(x, \_) \end{aligned} \tag{1.2}$$

Together with the condition that all distances be non-negative, we see that (1.1) characterizes completely the requirement for $D_{\widetilde{A}}$ to be a metric in the case of a two-letter alphabet A.

1.4 Two-letter alphabet, continued

For a general alphabet A, if

$$\alpha \leq D_A(x, y) \leq 2\alpha \tag{1.3}$$

for all $x \neq y$ (including _) for some $\alpha > 0$, then D_A is a metric (that is, the triangle inequality holds).

This is because

$$D_A(x, y) \leq 2\alpha \leq D_A(x, z) + D_A(z, y) \tag{1.4}$$

for any $z \in A$.

One simple choice for a metric on letters is to choose $D_A(x, y) = 1$ for all $x \neq y$, and then to take $D_{\widetilde{A}}(x, \_) = 2$; the resulting $D_{\widetilde{A}}$ satisfies (1.3) for $\widetilde{A}$.

However, condition (1.3) is far from optimal as the example (1.1) shows.

1.5 Edit distance definition

Edit distance d_e is derived from the extended alphabet distance $D_{\widetilde{A}}$ as follows.

We introduce the notion of *alignment* $\mathcal{A}$ of sequences $(x^*, y^*) = \mathcal{A}(x, y)$ where $x*$ has the letters of x in the same order but possibly with gaps _ inserted, and similarly for y^*.

We suppose that $x*$ and y^* have the same length even if x and y did not, which can always be achieved by adding gaps at one end or the other. Then

$$d_e(x, y) = \min_{\mathcal{A}} \sum_i D_{\widetilde{A}}(x_i^*, y_i^*). \tag{1.5}$$

The minimum is over all alignments $\mathcal{A}$ and the sum extends over the length of the sequences.

Fortunately, string-edit distance d_e, and even more complex metrics involving more complex gap penalties, can be computed efficiently by the dynamic programming algorithm [1].

References

[1] Michael Waterman. *Introduction to Computational Biology*. Chapman & Hall/CRC Press, 1995.