i. *Mutation*, where one symbol is replaced by another symbol. Note that a mutation can always be performed by an insertion followed by a deletion, but if we allow mutations, then this change counts for only 1, not 2, when computing the edit distance.

ii. *Transposition*, where two adjacent symbols have their positions swapped. Like a mutation, we can simulate a transposition by one insertion followed by one deletion, but here we count only 1 for these two steps.

Repeat Exercise 3.5.7 if edit distance is defined to be the number of insertions, deletions, mutations, and transpositions needed to transform one string into another.

! **Exercise 3.5.9:** Prove that the edit distance discussed in Exercise 3.5.8 is indeed a distance measure.

Exercise 3.5.10: Find the Hamming distances between each pair of the following vectors: 000000, 110011, 010101, and 011100.

3.6 The Theory of Locality-Sensitive Functions

The LSH technique developed in Section 3.4 is one example of a family of functions (the minhash functions) that can be combined (by the banding technique) to distinguish strongly between pairs at a low distance from pairs at a high distance. The steepness of the S-curve in Fig. 3.8 reflects how effectively we can avoid false positives and false negatives among the candidate pairs.

Now, we shall explore other families of functions, besides the minhash functions, that can serve to produce candidate pairs efficiently. These functions can apply to the space of sets and the Jaccard distance, or to another space and/or another distance measure. There are three conditions that we need for a family of functions:

1. They must be more likely to make close pairs be candidate pairs than distant pairs. We make this notion precise in Section 3.6.1.
2. They must be statistically independent, in the sense that it is possible to estimate the probability that two or more functions will all give a certain response by the product rule for independent events.
3. They must be efficient, in two ways:
   (a) They must be able to identify candidate pairs in time much less than the time it takes to look at all pairs. For example, minhash functions have this capability, since we can hash sets to minhash values in time proportional to the size of the data, rather than the square of the number of sets in the data. Since sets with common values are colocated in a bucket, we have implicitly produced the

candidate pairs for a single minhash function in time much less than the number of pairs of sets.

(b) They must be combinable to build functions that are better at avoiding false positives and negatives, and the combined functions must also take time that is much less than the number of pairs. For example, the banding technique of Section 3.4.1 takes single minhash functions, which satisfy condition 3a but do not, by themselves have the S-curve behavior we want, and produces from a number of minhash functions a combined function that has the S-curve shape.

Our first step is to define "locality-sensitive functions" generally. We then see how the idea can be applied in several applications. Finally, we discuss how to apply the theory to arbitrary data with either a cosine distance or a Euclidean distance measure.

3.6.1 Locality-Sensitive Functions

For the purposes of this section, we shall consider functions that take two items and render a decision about whether these items should be a candidate pair. In many cases, the function f will "hash" items, and the decision will be based on whether or not the result is equal. Because it is convenient to use the notation $f(x) = f(y)$ to mean that $f(x, y)$ is "yes; make x and y a candidate pair," we shall use $f(x) = f(y)$ as a shorthand with this meaning. We also use $f(x) \neq f(y)$ to mean "do not make x and y a candidate pair unless some other function concludes we should do so."

A collection of functions of this form will be called a *family* of functions. For example, the family of minhash functions, each based on one of the possible permutations of rows of a characteristic matrix, form a family.

Let $d_1 < d_2$ be two distances according to some distance measure d. A family $\mathbf{F}$ of functions is said to be (d_1, d_2, p_1, p_2)-*sensitive* if for every f in $\mathbf{F}$:

1. If $d(x, y) \leq d_1$, then the probability that $f(x) = f(y)$ is at least p_1.
2. If $d(x, y) \geq d_2$, then the probability that $f(x) = f(y)$ is at most p_2.

Figure 3.10 illustrates what we expect about the probability that a given function in a (d_1, d_2, p_1, p_2)-sensitive family will declare two items to be a candidate pair. Notice that we say nothing about what happens when the distance between the items is strictly between d_1 and d_2, but we can make d_1 and d_2 as close as we wish. The penalty is that typically p_1 and p_2 are then close as well. As we shall see, it is possible to drive p_1 and p_2 apart while keeping d_1 and d_2 fixed.

3.6.2 Locality-Sensitive Families for Jaccard Distance

For the moment, we have only one way to find a family of locality-sensitive functions: use the family of minhash functions, and assume that the distance

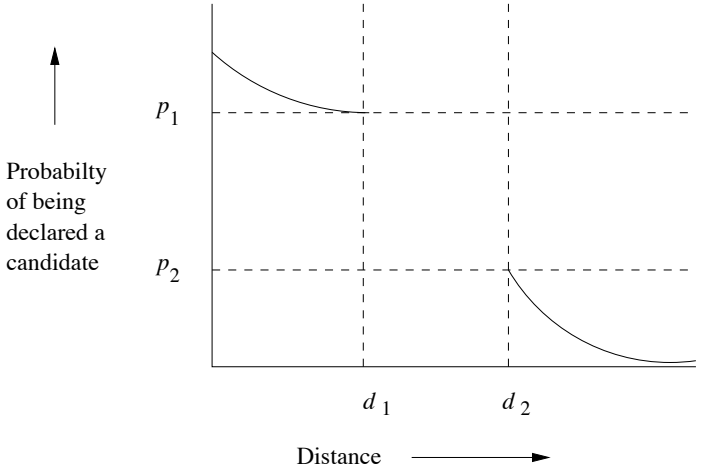

Figure 3.10: Behavior of a (d_1, d_2, p_1, p_2)-sensitive function

measure is the Jaccard distance. As before, we interpret a minhash function h to make x and y a candidate pair if and only if $h(x) = h(y)$.

- The family of minhash functions is a $(d_1, d_2, 1-d_1, 1-d_2)$-sensitive family for any d_1 and d_2, where $0 \leq d_1 < d_2 \leq 1$.

The reason is that if $d(x, y) \leq d_1$, where d is the Jaccard distance, then $\text{SIM}(x, y) = 1 - d(x, y) \geq 1 - d_1$. But we know that the Jaccard similarity of x and y is equal to the probability that a minhash function will hash x and y to the same value. A similar argument applies to d_2 or any distance.

Example 3.18: We could let $d_1 = 0.3$ and $d_2 = 0.6$. Then we can assert that the family of minhash functions is a $(0.3, 0.6, 0.7, 0.4)$-sensitive family. That is, if the Jaccard distance between x and y is at most 0.3 (i.e., $\text{SIM}(x, y) \geq 0.7$) then there is at least a 0.7 chance that a minhash function will send x and y to the same value, and if the Jaccard distance between x and y is at least 0.6 (i.e., $\text{SIM}(x, y) \leq 0.4$), then there is at most a 0.4 chance that x and y will be sent to the same value. Note that we could make the same assertion with another choice of d_1 and d_2; only $d_1 < d_2$ is required. □

3.6.3 Amplifying a Locality-Sensitive Family

Suppose we are given a (d_1, d_2, p_1, p_2)-sensitive family $\mathbf{F}$. We can construct a new family $\mathbf{F}'$ by the *AND-construction* on $\mathbf{F}$, which is defined as follows. Each member of $\mathbf{F}'$ consists of r members of $\mathbf{F}$ for some fixed r. If f is in $\mathbf{F}'$, and f is constructed from the set $\{f_1, f_2, \ldots, f_r\}$ of members of $\mathbf{F}$, we say $f(x) = f(y)$ if and only if $f_i(x) = f_i(y)$ for all $i = 1, 2, \ldots, r$. Notice that this construction mirrors the effect of the r rows in a single band: the band makes x and y a

candidate pair if every one of the r rows in the band say that x and y are equal (and therefore a candidate pair according to that row).

Since the members of $\mathbf{F}$ are independently chosen to make a member of $\mathbf{F}'$, we can assert that $\mathbf{F}'$ is a $\big(d_1, d_2, (p_1)^r, (p_2)^r\big)$-sensitive family. That is, for any p, if p is the probability that a member of $\mathbf{F}$ will declare (x, y) to be a candidate pair, then the probability that a member of $\mathbf{F}'$ will so declare is p^r.

There is another construction, which we call the *OR-construction*, that turns a (d_1, d_2, p_1, p_2)-sensitive family $\mathbf{F}$ into a $\big(d_1, d_2, 1-(1-p_1)^b, 1-(1-p_2)^b\big)$-sensitive family $\mathbf{F}'$. Each member f of $\mathbf{F}'$ is constructed from b members of $\mathbf{F}$, say $f_1, f_2, \ldots, f_b$. We define $f(x) = f(y)$ if and only if $f_i(x) = f_i(y)$ for one or more values of i. The OR-construction mirrors the effect of combining several bands: x and y become a candidate pair if any band makes them a candidate pair.

If p is the probability that a member of $\mathbf{F}$ will declare (x, y) to be a candidate pair, then $1-p$ is the probability it will not so declare. $(1-p)^b$ is the probability that none of $f_1, f_2, \ldots, f_b$ will declare (x, y) a candidate pair, and $1-(1-p)^b$ is the probability that at least one f_i will declare (x, y) a candidate pair, and therefore that f will declare (x, y) to be a candidate pair.

Notice that the AND-construction lowers all probabilities, but if we choose $\mathbf{F}$ and r judiciously, we can make the small probability p_2 get very close to 0, while the higher probability p_1 stays significantly away from 0. Similarly, the OR-construction makes all probabilities rise, but by choosing $\mathbf{F}$ and b judiciously, we can make the larger probability approach 1 while the smaller probability remains bounded away from 1. We can cascade AND- and OR-constructions in any order to make the low probability close to 0 and the high probability close to 1. Of course the more constructions we use, and the higher the values of r and b that we pick, the larger the number of functions from the original family that we are forced to use. Thus, the better the final family of functions is, the longer it takes to apply the functions from this family.

Example 3.19: Suppose we start with a family $\mathbf{F}$. We use the AND-construction with $r = 4$ to produce a family $\mathbf{F}_1$. We then apply the OR-construction to $\mathbf{F}_1$ with $b = 4$ to produce a third family $\mathbf{F}_2$. Note that the members of $\mathbf{F}_2$ each are built from 16 members of $\mathbf{F}$, and the situation is analogous to starting with 16 minhash functions and treating them as four bands of four rows each.

The 4-way AND-function converts any probability p into p^4. When we follow it by the 4-way OR-construction, that probability is further converted into $1-(1-p^4)^4$. Some values of this transformation are indicated in Fig. 3.11. This function is an S-curve, staying low for a while, then rising steeply (although not too steeply; the slope never gets much higher than 2), and then leveling off at high values. Like any S-curve, it has a *fixedpoint*, the value of p that is left unchanged when we apply the function of the S-curve. In this case, the fixedpoint is the value of p for which $p = 1-(1-p^4)^4$. We can see that the fixedpoint is somewhere between 0.7 and 0.8. Below that value, probabilities are decreased, and above it they are increased. Thus, if we pick a high probability

p	$1-(1-p^4)^4$
0.2	0.0064
0.3	0.0320
0.4	0.0985
0.5	0.2275
0.6	0.4260
0.7	0.6666
0.8	0.8785
0.9	0.9860

Figure 3.11: Effect of the 4-way AND-construction followed by the 4-way OR-construction

above the fixedpoint and a low probability below it, we shall have the desired effect that the low probability is decreased and the high probability is increased.

Suppose $\mathbf{F}$ is the minhash functions, regarded as a $(0.2, 0.6, 0.8, 0.4)$-sensitive family. Then $\mathbf{F}_2$, the family constructed by a 4-way AND followed by a 4-way OR, is a $(0.2, 0.6, 0.8785, 0.0985)$-sensitive family, as we can read from the rows for 0.8 and 0.4 in Fig. 3.11. By replacing $\mathbf{F}$ by $\mathbf{F}_2$, we have reduced both the false-negative and false-positive rates, at the cost of making application of the functions take 16 times as long. □

p	$\left(1-(1-p)^4\right)^4$
0.1	0.0140
0.2	0.1215
0.3	0.3334
0.4	0.5740
0.5	0.7725
0.6	0.9015
0.7	0.9680
0.8	0.9936

Figure 3.12: Effect of the 4-way OR-construction followed by the 4-way AND-construction

Example 3.20 : For the same cost, we can apply a 4-way OR-construction followed by a 4-way AND-construction. Figure 3.12 gives the transformation on probabilities implied by this construction. For instance, suppose that $\mathbf{F}$ is a $(0.2, 0.6, 0.8, 0.4)$-sensitive family. Then the constructed family is a

$$(0.2, 0.6, 0.9936, 0.5740)\text{-sensitive}$$