The theory of locality-sensitive function is similar with locality-sensitive hashing, the only different is people use other functions to instead of hashing function to produce candidate pairs and also those functions can apply to the space of sets and the Jaccard distance or to another space and distance measure. For those functions they have multiple conditions, the first condition is they have to pick the closest pairs to be candidate pairs, because when we use a functions to consider two values are candidate pairs or not, we need to hashing those two values first, if two hashing value are equal then those two values are candidate pairs, if the two hashing values are not equal, the functions will determinate those two values are not candidate pairs. And those groups of functions we called family of functions. For example, family of hashing functions. And usually we use form (d1,d2,p1,p2) to expression locality-sensitive function, and d1 smaller than d2. When the distance d(x,y) smaller or equal to d1, then x and y is a candidate pair probability is least p1. When the distance d(x,y) bigger or equal to d2, then x and y is a candidate pair probability will not be bigger than p2. By the word, people do not have any rule to determinate the distance between the d1 and d2, but we should let d1 and 2 as closest as possible, but when the distance between the d1 and d2 smaller, the p1 and p2 will be close too. The second condition is every function we used must be independent of each other. In other words, the probability between multiple functions should be same with every function’s probability multiply together. Third conditions are the functions should have highly efficient on determinate candidate pairs and can reduce the false-positive rate and false-negative rate when the functions combined together. For example, the single hashing function will not fit this condition, the min Hash can find candidate pairs in a short time, but the single hashing function incompatible S-curve format, but if we combine all the min hashing function together, the function will conform S-curve format. If we want to find the family of locality-sensitive function, we only can use family of min hashing functions and assume the measure distance is Jaccard distance, and we use similar way to explain the min hash function number h, and if h(x) = h(y), the x and y will be candidate pairs, and we will get the family of sensitive min hash functions is (d1, d2, 1-d1, 1-d2). This conclusion will be proved when Jaccard distance between x and y smaller than d1, then SIM(x,y) = 1 – d(x,y) >= 1-d1. And the Jaccard between x and y will equal to the probability that a min hash function will hash x and y with the same value. And we can use the same way to find the d2. In the amplifying a locality-sensitive family, we have two algorithms, AND-construction and OR-construction. When we assume family of sensitive function F is (d1, d2, p1, p2), and we use AND-construction to construct the family of sensitive function f we can get a new family of function F’, every members of functions in F’ is produce by r number of members of functions in F. When we want to find f(x)=f(y), we need to make sure every i has fi(x) = fi(y), if in one row every x and y are equal in r row, we can assume x and y is candidate pairs. And we can conclude that the family of sensitive function F’ is (d1, d2, (p1)^r, (p2)^r). Another construction algorithm is OR-construction, we can use OR-construction to construct F to (d1, d2, 1-(1-p1)^b, 1-(1-p2)^b). By the word, when we use AND-construction, we will reduce every probability, and OR-construction will raise every probability, so the number F and r will be very important, because if F and r are suitable, we will make construction more trustable.