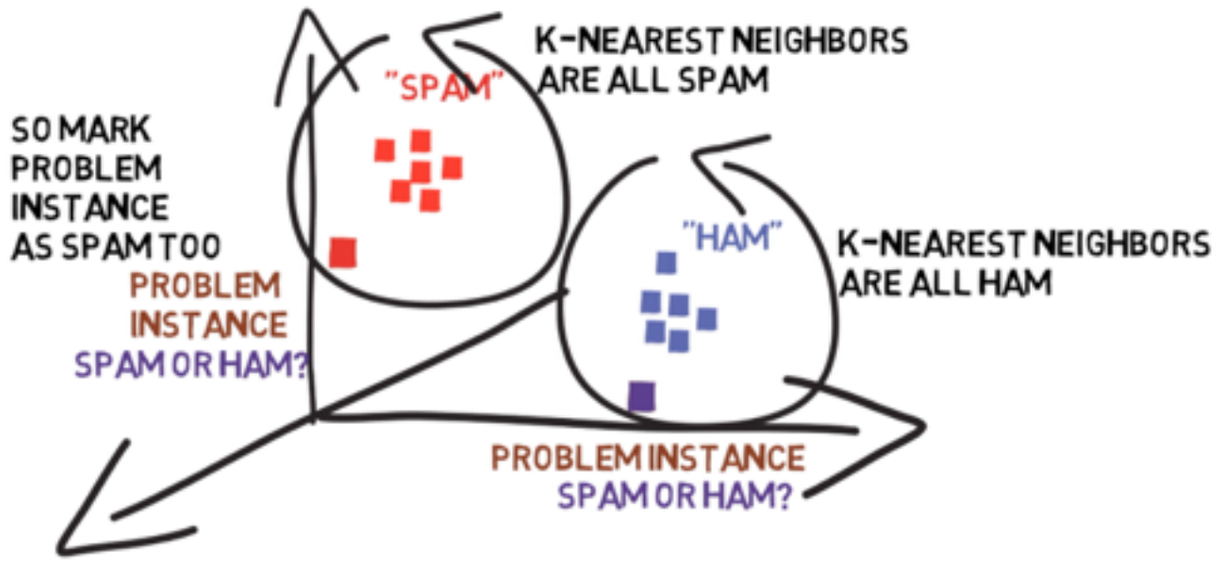
K-NEAREST NEIGHBOURS



K-NEAREST NEIGHBORS IS A
RELATIVELY SIMPLE ALGORITHM
TO VISUALIZE

BUT THERE ARE A FEW DIFFERENT WRINKLES
WE SHOULD UNDERSTAND TOO

DEFINITION OF DISTANCE

DATA REDUCTION

THE CHOICE OF K

DIMENSIONALITY REDUCTION

USE IN PREDICTION

DIMENSIONALITY REDUCTION

DIMENSIONALITY : LENGTH OF THE FEATURE VECTOR : NUMBER OF COORDINATES NEEDED TO EXPRESS EACH POINT

AS THE DIMENSIONALITY INCREASES, BOTH THE EFFICACY AND EFFICIENCY OF K-NEAREST NEIGHBOR SUFFER

EFFICIENCY SUFFERS BECAUSE FINDING THE POINTWISE DISTANCE BECOMES VERY COMPUTATIONALLY EXPENSIVE

EFFICACY SUFFERS BECAUSE IN A HIGH-DIMENSIONAL SPACE, THE DISTANCE FORMULA CHOSEN MIGHT FAIL TO DIFFERENTIATE BETWEEN NEIGHBORS - THEY MIGHT ALL SEEM TO BE APPROXIMATELY EQUIDISTANT

FOR THIS REASON, SOME FORM OF DIMENSIONALITY REDUCTION IS USUALLY APPLIED TO FEATURE VECTORS BEFORE K-NEAREST NEIGHBOR IS USED:

FEATURE EXTRACTION

FOR INSTANCE USING PRINCIPAL COMPONENTS ANALYSIS

ANOTHER SMART DIMENSIONALITY REDUCTION TRICK IS HASHING THE FEATURES USING A HASH FUNCTION THAT MAPS SIMILAR ITEMS TO SIMILAR BUCKETS

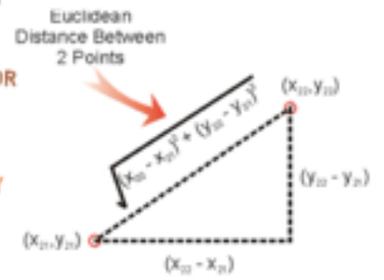
LOCALITY SENSITIVE HASHING

DEFINITION OF DISTANCE

EUCLIDEAN DISTANCE IS THE MOST COMMON DEFINITION OF DISTANCE USED IN GEOMETRY

BUT THIS IS NOT ALWAYS A SUITABLE DEFINITION FOR USE IN K-NEAREST NEIGHBOR

THE ALGORITHM IS OFTEN MODIFIED TO DOWN-WEIGHT THE IMPORTANCE OF FAR POINTS RELATIVE TO THAT OF NEARBY POINTS



ALSO, EUCLIDEAN DISTANCE ONLY WORKS WHEN THE FEATURE IS A CONTINUOUS VARIABLE (RATHER THAN A DISCRETE ONE)

FOR 2 STRINGS, EDIT DISTANCE IS THE MINIMUM NUMBER OF EDITS NEEDED TO GO FROM ONE TO THE OTHER

FOR DISCRETE VARIABLES, SOME OTHER DISTANCE MEASURE, SUCH AS EDIT DISTANCE OR HAMMING DISTANCE WILL NEED TO BE USED

CHOICE OF K

(THIS IS CALLED PARAMETER SELECTION)

AN IMPORTANT SPECIAL CASE IS K = 1, CALLED THE NEAREST NEIGHBOR ALGORITHM

LARGE VALUES OF K HELP REDUCE THE EFFECT OF OUTLIERS IN THE DATA -

BUT ALSO INCREASE THE CHANCES THAT WE WILL PULL IN A LARGE NUMBER OF INSTANCES OF THE WRONG CATEGORY

DATA REDUCTION

WE HAVE SPOKEN BRIEFLY ABOUT DIMENSIONALITY REDUCTION - WHICH INVOLVES REDUCING THE NUMBER OF COORDINATES OF EACH DATA POINT -

BUT ALSO IMPORTANT IN K-NEAREST NEIGHBOR IS DATA REDUCTION, WHICH INVOLVES TRIMMING THE NUMBER OF DATA POINTS WHERE POSSIBLE

GETTING RID OF THE RIGHT SET OF POINTS WILL IMPROVE BOTH THE COMPUTATIONAL EFFICIENCY AND EFFICACY OF THE K-NN

A COMMON DATA REDUCTION TECHNIQUE IS TO DIVIDE THE TRAINING DATA INTO 3 CATEGORIES

PROTOTYPES WHICH REPRESENT THE TRAINING DATA PARTICULARLY WELL (THEY ARE SELECTED BY SOME PROTOTYPE SELECTION ALGORITHM)

CLASS OUTLIERS ARE POINTS IN THE TRAINING DATA THAT ARE NOT CORRECTLY CLASSIFIED USING THE PROTOTYPES

ABSORBED POINTS POINTS IN THE TRAINING DATA THAT ARE INDEED CORRECTLY CLASSIFIED BY THE PROTOTYPES

THE DATA REDUCTION PROCESS THEN RETAINS ALL THE PROTOTYPES, DISCARDS ALL THE ABSORBED POINTS, AND SELECTIVELY KEEPS THE CLASS OUTLIERS

USE IN PREDICTION

OUR DISCUSSION SO FAR HAS FOCUSED ON THE USE OF K-NN FOR **CLASSIFICATION**

GIVEN A PROBLEM INSTANCE (POINT TO CLASSIFY), WE CHECKED WHAT CATEGORY A MAJORITY OF ITS NEIGHBORS BELONG TO, AND ASSIGN THAT CATEGORY TO THE PROBLEM INSTANCE

WE CAN JUST AS EASILY USE THIS METHOD TO "PREDICT" THE VALUE OF ANY FUNCTION FOR A PROBLEM INSTANCE -

SIMPLY CALCULATE THE FUNCTION TO BE PREDICTED FOR EACH OF THE K NEAREST NEIGHBORS, AND USE THE AVERAGE AS OUR PREDICTION