Expert Data Mining: k-value

# Introduction

For this next iteration of the Expert Data Mining software, the program now works with k-value instead of just Boolean. K-value refers to how many values that a particular attribute can have. The below k-values follow a convention that they are always 0 through k – 1.. For example, a k-value of 2 is equivalent to Boolean values 0 and 1, but a k-value of 3 is equivalent to values of 0 through 2. Let’s say that we have a dataset of just 2 dimensions, and the k-value for each attribute is 3. The following datapoints will be in the dataset:

(0, 0)

(1, 0)

(0, 1)

(1, 1)

(2, 0)

(0, 2)

(2, 1)

(1, 2)

(2, 2)

Furthermore, the sections below will describe how k-value will be different for the corresponding options.

# Majority Flag

K-value for majority flag is quite simple: a datapoint is a majority datapoint if the Hamming norm is greater than or equal to half of the maximum Hamming norm of a datapoint. E.g., let’s say that we have a dataset with a dimension of 2 with values of k = {3, 2}. Therefore, the largest datapoint of (2, 1) has a maximum Hamming norm of 3. Therefore, (2, 0) and (1, 1) are both majority datapoints.

If the user decides to use the majority flag, then all majority vectors are asked first, and once those are done, the normal sequence of questions is asked. If the user does not know roughly how many majority vectors are true, then all majority vectors are asked first. If the user specifies that a number of these majority vectors are true, then we ask at random, and if half of those majority vectors are true, the normal sequence of questions is asked. The idea is that it is likely that for a given dimension, it will take half or slightly over half of the attributes to be true for the vector to be true, but that is not guaranteed. The majority flag can be paired with chain jumping. In this case, after the majority questions are asked, the Hansel Chains which have majority vectors that are true are asked before returning to the normal sequence of Hansel Chains.

# True Attributes

Originally, a true attribute was an attribute that needed to be true for the datapoint to have a class of 1. That is still the case with k-value, but now the value of that attribute needs to be specified as well if the k-value is greater than 2. For example, if the k-value of an attribute is 4, then the user could specify that the value of that attribute needs to be 2 for that attribute to be true.

# f-changes

f-changes are mostly the same as they were previously. However, if the f-change causes a violation of monotonicity and the user decides to add an attribute to fix the violation, then we need to ask the k-value for the new attribute. Otherwise, this feature will function the same as before.

# Nested Functions

INSERT OG DESCRIPTIONS HERE