**Analyzing multiplicative vs additive curve approximation**

**Analysis regarding the specific NN algorithm**

The additive NN algorithm stops the pruning process if an upper bound is less than or equal to the additive approximate threshold. The multiplicative (1+) NN algorithm does not have this logic since it doesn’t know what the approximate threshold value is yet at the pruning stage. For certain datasets with specific additive approximate threshold values this results in fewer nodes searched vs the multiplicative method. For example, here are results for the pen-tip dataset with errors that are relatively similar in size:

* Additive NN – 282 average node searches
* Multiplicative NN – 666 average node searches

**General analysis on the multiplicative curve approximation**

The location of the query curve w.r.t. the dataset curves determines the effectiveness of the multiplicative curve approximation. For example, if the query curve happens to be very close to a curve in the dataset then the next closest curve may be 300% away in relative distance. If the multiplicative approximation factor is 100% then it performs an exact search on the pruned candidate curves. However, if the query curve happens to be in the “middle” of many curves where the distances from the query to the curves are fairly close, then the approximation part of the algorithm works better. The problem is, the user has no idea for a given query curve if it will be very close to a single curve in the dataset, or in the “middle” of a set of curves.

**General analysis on the additive curve approximation**

The additive curve approximation overcomes the multiplicative issue above, but has two issues:

1. Users must know the additive distances and UOM they want to measure for the approximation threshold which can be a bit difficult depending on the dataset.
2. Larger or longer curves probably require larger additive distances compared to smaller curves, so users have to constantly adjust the additive approximation distance depending on the query.

The two issues above can be somewhat resolved by using the Reach concept. The user supplies an approximation factor that is a percent of the query curve’s Reach. This is converted into an additive approximation value in the query algorithm. By doing this the user does not have to be concerned with the UOM or additive distances, and it automatically scales to smaller or larger curves.