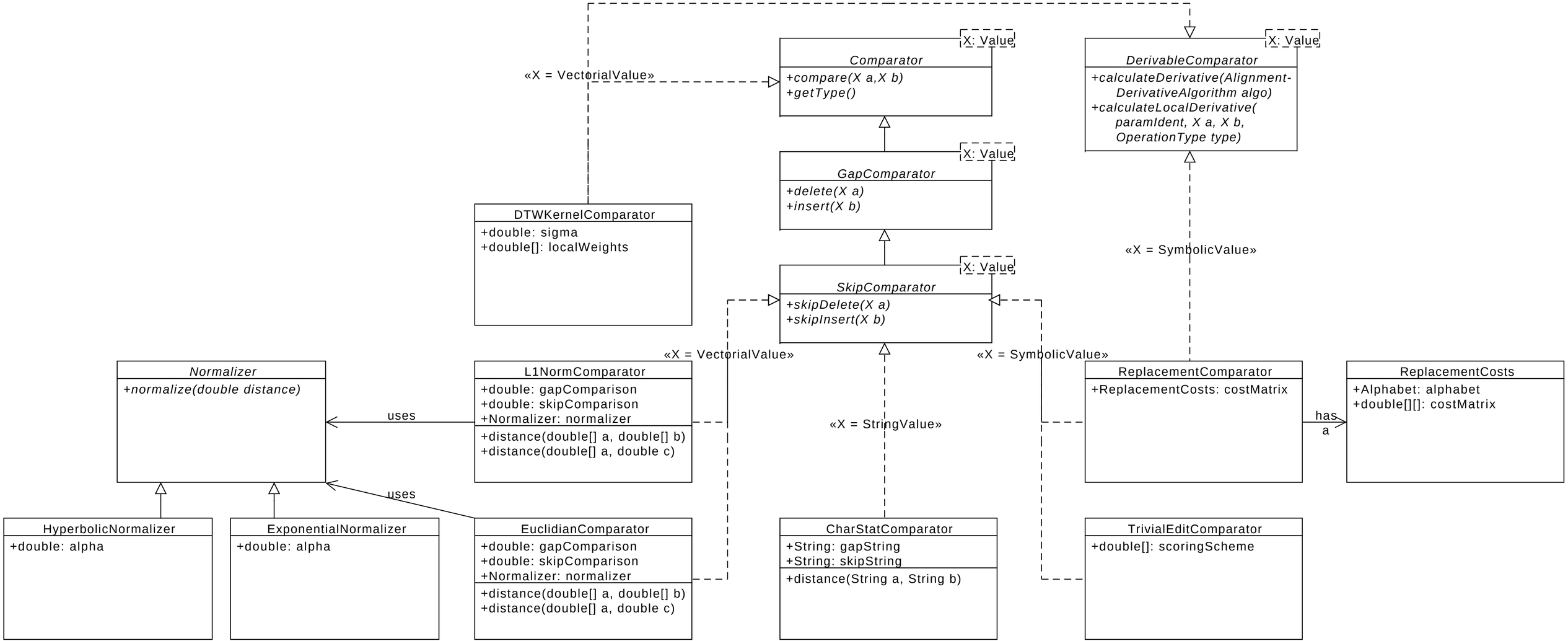
X: Value
Comparator
+compare(X a,X b)
+getType()
X: Value
DerivableComparator
+calculateDerivative(Alignment-DerivativeAlgorithm algo)
+calculateLocalDerivative(paramIdent, X a, X b, OperationType type)
«X = VectorialValue»
X: Value
GapComparator
+delete(X a)
+insert(X b)
DTWKernelComparator
+double: sigma
+double[]: localWeights
«X = SymbolicValue»
X: Value
SkipComparator
+skipDelete(X a)
+skipInsert(X b)
«X = VectorialValue»
«X = SymbolicValue»
Normalizer
+normalize(double distance)
L1NormComparator
+double: gapComparison
+double: skipComparison
+Normalizer: normalizer
+distance(double[] a, double[] b)
+distance(double[] a, double c)
uses
«X = StringValue»
ReplacementComparator
+ReplacementCosts: costMatrix
has a
ReplacementCosts
+Alphabet: alphabet
+double[][]: costMatrix
uses
HyperbolicNormalizer
+double: alpha
ExponentialNormalizer
+double: alpha
EuclidianComparator
+double: gapComparison
+double: skipComparison
+Normalizer: normalizer
+distance(double[] a, double[] b)
+distance(double[] a, double c)
CharStatComparator
+String: gapString
+String: skipString
+distance(String a, String b)
TrivialEditComparator
+double[]: scoringScheme