Probabilistic Feature

The probabilistic feature provides functions and operators to process discrete and continuous probabilistic values in a data stream. Continuous probabilistic values are represented using Gaussian Mixtures.

To enable the probabilistic processing you have to include the probabilistic feature and use the probabilistic metadata *(#METADATA Probabilistic)* in your Odysseus script.

Estimating probabilistic values

ToDo:

Expectation Maximization

The EM operator allows the fit a Gaussian mixture model (GMM) with predefined number of mixtures to the values of a data stream.

Kalman Filter

The Kalman Filter operator can be used if the variance of of the values in the data stream is known from some datasheet.

Filtering probabilistic values

For filtering probabilistic values you can use the same syntax that you already use for deterministic values. However, the result of the operators differ. In case of discrete probabilistic values the Select operator returns a tuple with a lower tuple existence probability.

Lets assume you have an attribute x and that attribute is 1.0 with probability 0.25, 2.0 with probability 0.25, and 3.0 with probability 0.5. The following Select operation will now filter the attribute value such that the resulting attribute value can only be instantiated to 2.0 and the resulting tuple existence is reduced to 0.25.

Probabilistic discrete select

```
output = SELECT({predicate = ProbabilisticRelationalPredicate('x > 1.0 AND x < 3.0')}, input)
```

The filtering of continuous probabilistic distributions is similar to the processing of discrete probabilistic values in the fact that it may reduce the tuple existence probability.

Lets assume you have a random variable x with mean 0.0 and ^{2} 1.0 the following query will set the tuple existence to the cumulative probability that this random variable will take a value between the upper and lower bound that is ~0.1586235826896239.

Probabilistic continuous select

```
output = SELECT({predicate = ProbabilisticRelationalPredicate('x > 1.0 AND x < 4.0')}, input)
```

Joining probabilistic values

The join with a predicate based on probabilistic discrete values uses the same syntax as for deterministic values. Although it looks similar the result is different in the sense that the Join operator performs a join of the input streams in each possible world and as such the operator may produce more tuple.

Probabilistic Join

```
Select * From input1,input2 WHERE input1.x=input2.y;
```

As you can see, the probabilistic processing is not limit to PQL. You can use the same CQL syntax you already used for deterministic values.

Working with probabilistic values

Now that you know how to filter and join probabilistic values you probably want to do something with the values like performing mathematic operations on them. To do so you can use the algebraic operator (+, *, -, /, ^) on probabilistic values in i.e. a Map operator. Attention, when using multiplication or division on continuous probabilistic values, the result is estimated by fitting Gaussian mixture models to resulting distribution.

Mathematical Functions

Int(Distribution, Lower Limit, Upper Limit)

Estimates the multivariate normal distribution probability with lower and upper integration limit.

Statistical Functions

Similarity(Distribution, Distribution)

Calculates the Bhattacharyya distance between two distributions.

Example

```
SELECT similarity(as2DVector(x1,y1), as2DVector(x2,y2)) FROM stream
```

Distance(Distribution, Value)

Calculates the Mahalanobis distance between the distribution and the value. The value can be a scalar value or a vector.

Example

```
SELECT distance(as3DVector(x, y, z), [1.0;2.0;3.0]) FROM stream
```

Datatype Functions

as2DVector(Object, Object)

Converts the two object into a 2D vector.

as3DVector(Object, Object, Object)

Similar to the as2DVector function, this function creates a 3D vector with the given objects.

Access to tuple existence

To access the tuple existence during processing you can use the ExistenceToPayload operator that copies the tuple existence to the payload where you can access them with the attribute name "meta_existence".

Probabilistic continuous select

```
output = ExistenceToPayload(SELECT({predicate = ProbabilisticRelationalPredicate('x > 1.0 AND x < 4.0')},
input))
```

`ProbabilisticRelationalPredicate`