Weighted RNG Sets

By Joseph Juma, July 20th 2023

> This document covers the definition of RNG sets and their associated value types. These are two kinds of sets used to create groups of values which have associated with them a “weight” that dictates how likely they are to be selected from the sample set in a fair randomly chosen sampling (”roll”). The P-Value is for when someone wants to be able to work directly with the probabilities, but should be careful as it does not automatically normalize the probabilities of all elements within it. The V-Value is for when someone wants an arbitrary weight object or number which is then used to compute a probability that’s automatically normalized for all values in the set. Subsequently, P-Sets require more manual effort to use, but store less data and run faster, while V-Sets store more information and run slower but perform the calculations of probability normalization automatically. It’s up to a developer to figure out which suites their use case, and the two can eve be interchanged.

Probability Weighted Value

$$P_v = \{v, p\} | p \in [0, 1]$$

- v is an arbitrary value. It can be any kind of value or object.
- p is a probability within the [0,1] domain.

Value Weighted Value

$$V_v = \{v, w, p\} | p \in [0, 1] \therefore V_v = \{P_v, w\}$$

- v is an arbitrary value. It can be any kind of value or object.
- p is a probability within the [0,1] domain.
- w is a weight value. It's used to calculate p when combined with other weight values.

Technically, a Value Weighted Value cannot calculate a p value without being in a set with other VWVs.

Probability Weighted Set (P-Weighted Set)

$$P_s = \{P_{(v,0)}, P_{(v,1)} ..., P_{(v,n-1)}\} | n = \#P_s$$

An arbitrary length ordered set (n-tuple or n-vector) of probability weighted values.

- The sum of each elements p-value should never go above one.
- Can be "renormalized" if the values go above one by using the normalization formula.

$$x = \sum_{i=0}^{n} p \in P_{(v,i)}$$
$$f : P_s \to P'_s | f(P_{(v,i)}) = (p \in P_{(v,i)} = \frac{p}{x})$$

Value Weighted Set (V-Weighted Set)

$$V_s = \{V_{(v,0)}, V_{(v,1)}, ..., V_{(v,n-1)}\} | n = \#V_s$$

An arbitrary length ordered set (n-tuple or n-vector) of value weighted values.

- The probability of each element is set via the normalization equation whenever a new value is added, or an element's weight is changed. This is the same formula as the "Renormalization" formula used in a P-weighted set.

$$x = \sum_{i=0}^{n} p \in V_{(v,i)}$$
$$f : V_s \to V'_s | f(V_{(v,i)} = (p \in V_{(v,i)} = \frac{w \in V_{(v,i)}}{x})$$

Normalization Formula

The normalization formula is the same one used in any arbitrary vector.

1. You sum together every element in a set into a sum (x)
2. You divide each element in the set by the sum, to get the normalized value.

In P-Sets

In P-Sets the probability should already be normalized (all p-values in the set sum to 1), but in the case it isn't, renormalization works just as the standard normalization formula. In this sense, the p-value of each value in the set is set via the formula.

In V-Sets

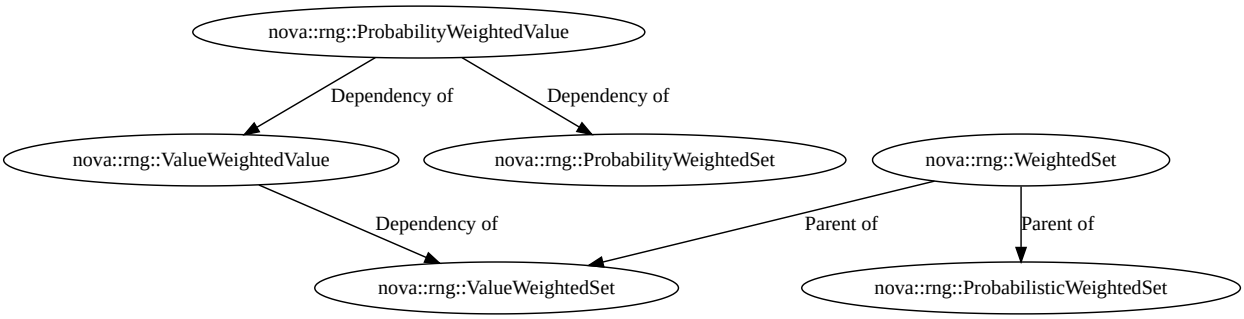
In V-Sets the probability is calculated on each element by normalizing the weights. In this sense, the weights are summed, then each weight divided by that sum to set the p-

value. This is so weights are still kept track of separately, and can be reused in later calculations.

Code Classes

1. `rng::WeightedSet` an interface defining the methods for the types of weighted sets.
2. `rng::ProbabilityWeightedValue` a probability weighted value (P_v).
3. `rng::ValueWeightedValue` A value weighted value (V_v).
4. `rng::ProbabilityWeightedSet` a P-set (P_s).
5. `rng::ValueWeightedSet` a V-set (V_s).

Class Relationships Diagram



Method Notes

- Because `ValueWeightedSet` is a set of `ValueWeightedValue` elements, and a `ProbabilityWeightedSet` is a set of `ProbabilityWeightedValues` and it is possible to convert a `ProbabilityWeightedValue` and `ValueWeightedValue`, conversion between the two is functional. The only missing value between a P_v and V_v is the w value, which has a calculable relationship to the p value of $p * \alpha = w$ where α is an arbitrary value that can be used to generate a weight from a probability. (`ValueWeightedSet::toProbabilityWeightedSet()` and `ProbabilityWeightedSet::toValueWeightedSet( unit )`)
- There should be a method for getting a value from these sets without their waiting effecting them. This is due to them inherently having non-primitive elements, so the

additional logic is for getting the actual value element from the structured value objects. This should be set in the interface as `WeightedSet::rollHomogenously()`. Alternatively, `WeightedSet::sampleHomogenously()`.

- There should be a general method for randomly selecting from a weighted set, `WeightedSet::roll()` or alternatively, `WeightedSet::sample()`.
- There's also the act of "adding" to the set, but this is where the set's specific elements come into play. As such, `WeightedSet` might benefit from being a template interface, and the two types of WeightedSets (P-Sets and V-Sets) as derived classes of specific template uses.

Weighted Set - Methods

```
WeightedSet<T>

// Generic set operations,
push(T& element);
T& peek();
T pop();

remove(i);
remove(T);
insert(T,i);

prepend // append
presect // adsect

// Operations shared by P-Set and V-Set
sample(); // Technically is 'sampleHeterogenous' :P
roll(); // Alias of sample

sampleHomogenous();
rollHomogenous(); // Alias of sampleHomogenous

normalize(); // Normalizes every element.
```

- It can be noted that this is a union of generic set operations, and the shared operations of a P-Set and V-Set. As such this class should be *theoretically* derivable from the union of an Array interface, and the interface generated by the intersection of P-Set and V-Set operations.

P-Set - Methods

```
ProbabilityWeightedSet : WeightedSet<ProbabilityWeightedValue>
toValueWeightedSet();
```

V-Set - Methods

```
ValueWeightedSet : WeightedSet<ValueWeightedValue>
toProbabilityWeightedSet();
```
