Attributes for Graph Objects

Seth Falcon

November 1, 2022

1 Introduction

The *graph* package provides representations of graphs (nodes and edges) as S4 classes. This vignette demonstrates how to add arbitrary node and edge attributes to graph objects.

First, we create a graph to use as an example. We will work with a *graphAM-class* instance, however, any subclass of *graph-class* would work. See Figure 1.

```
> library("graph")
> mat <- matrix(c(0, 0, 1, 1,
+                 0, 0, 1, 1,
+                 1, 1, 0, 1,
+                 1, 1, 1, 0),
+               byrow=TRUE, ncol=4)
> rownames(mat) <- letters[1:4]
> colnames(mat) <- letters[1:4]

> g1 <- graphAM(adjMat=mat)

A graphAM graph with undirected edges
Number of Nodes = 4
Number of Edges = 5
```

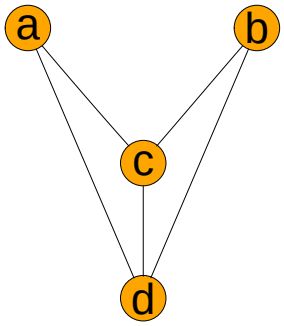

Figure 1: The graph `g1`.

2 Edge Attributes

2.1 Default edge attributes

All edges in a graph support the same set of attributes. The set of supported attributes can be defined and accessed using the `edgeDataDefaults` method. A new graph instance will not have any edge attributes defined.

```
> edgeDataDefaults(g1)

list()
```

When a new edge attribute is defined, a default value must be specified. Here we will define two edge attributes: `weight` and `code` and specify a default value for each one.

```
> edgeDataDefaults(g1, "weight") <- 1
> edgeDataDefaults(g1, "code") <- "plain"
> edgeDataDefaults(g1)

$weight
[1] 1

$code
[1] "plain"
```

The default value for a particular attribute can be obtained by specifying the attribute name in the call to `edgeDataDefaults`.

```
> edgeDataDefaults(g1, "weight")

[1] 1
```

2.2 Getting edge attributes

Edge attributes are set and accessed using the `edgeData` method. Only attributes defined using `edgeDataDefaults` can be accessed using `edgeData`. If an attribute has not be set using `edgeData` for a given edge, then the default value is used.

```
> edgeData(g1, from="a", to="d", attr="weight")

$`a|d`
[1] 1

> edgeData(g1, from="a", attr="weight")

$`a|c`
[1] 1

$`a|d`
[1] 1

> edgeData(g1, to="a", attr="weight")
```

```
$`c|a`
[1] 1

$`d|a`
[1] 1
```

```
> allAttrsAllEdges <- edgeData(g1)
> weightAttrAllEdges <- edgeData(g1, attr="weight")
```

2.3 Setting edge attributes

Attributes are set using the replacement form of `edgeData`. This method allows the user to update the attribute for single edge, set the attributes for a collection of edges to a single value, and to set the attributes for a collection of edges to different values specified by a vector of values.

```
> edgeData(g1, from="a", to="d", attr="weight") <- 2
> edgeData(g1, from="a", attr="code") <- "fancy"
> edgeData(g1, from="a", attr="weight")
```

```
$`a|c`
[1] 1

$`a|d`
[1] 2
```

```
> edgeData(g1, from="a", attr="code")
```

```
$`a|c`
[1] "fancy"

$`a|d`
[1] "fancy"
```

We can set the attributes for multiple edges to a single value.

```
> f <- c("a", "b")
> t <- c("c", "c")
> edgeData(g1, from=f, to=t, attr="weight") <- 10
> edgeData(g1, from=f, to=t, attr="weight")
```

```
$`a|c`
[1] 10

$`b|c`
[1] 10
```

It is also possible to set multiple attributes to different values in a single call to `edgeData`.

```
> edgeData(g1, from=f, to=t, attr="weight") <- c(11, 22)
> edgeData(g1, from=f, to=t, attr="weight")
```

```
$`a|c`
[1] 11

$`b|c`
[1] 22
```

Finally, we can set the an attribute to a vector of values by packing it into a list:

```
> edgeData(g1, from="a", to="d", attr="code") <- list(1:10)
> edgeData(g1, from=f, to=t, attr="weight") <- mapply(c, f, t, "e", SIMPLIFY=FALSE)
> edgeData(g1, from="a", to="d", attr="code")

$`a|d`
 [1]  1  2  3  4  5  6  7  8  9 10

> edgeData(g1, from=f, to=t, attr="weight")

$`a|c`
[1] "a" "c" "e"

$`b|c`
[1] "b" "c" "e"
```

3 Node Attributes

3.1 Default node attributes

Like edge attributes, all nodes in a graph support the same set of attributes. The supported set of attributes and their default values is accessed using the `nodeDataDefaults` method. The interface is similar to `edgeDataDefaults`.

```
> nodeDataDefaults(g1)

list()

> nodeDataDefaults(g1, attr="weight") <- 1
> nodeDataDefaults(g1, attr="type") <- "vital"
> nodeDataDefaults(g1)

$weight
[1] 1

$type
[1] "vital"

> nodeDataDefaults(g1, "weight")

[1] 1
```

As with edge attributes, default values are required for each node attribute. The default value is used as the node attribute for all nodes in the graph that have not had their attribute value explicitly set. Attribute values can be any R object.

3.2 Getting and setting node attributes

Once a node attribute has been defined and given a default value using `nodeDataDefaults`, individual node attributes can be accessed using `nodeData`.

```
> nodeData(g1, n="a")
```

```
$a
$a$weight
[1] 1

$a$type
[1] "vital"
```

```
> nodeData(g1, n="a", attr="weight") <- 100
> nodeData(g1, n=c("a", "b"), attr="weight")
```

```
$a
[1] 100

$b
[1] 1
```

```
> nodeData(g1, n=c("a", "b"), attr="weight") <- 500
> nodeData(g1, n=c("a", "b"), attr="weight")
```

```
$a
[1] 500

$b
[1] 500
```

```
> nodeData(g1, n=c("a", "b"), attr="weight") <- c(11, 22)
> nodeData(g1, n=c("a", "b"), attr="weight")
```

```
$a
[1] 11

$b
[1] 22
```