

Appendix - Useful distributions

Beta

Dirichlet

We will make considerable use of Dirichlet distributions in the chapter on Hierarchical models (hierarchical-models.html). In many models, we want to sample a category from a set of categories (e.g., a word from a list of words). When we use `categorical()`, we need to provide the probability for each category. This is problematic when we don't know the probabilities in question.

As usual when we don't know the value of something, we sample it from a prior. The Dirichlet distribution – which is the higher-dimensional analogue of the Beta distribution (https://en.wikipedia.org/wiki/Beta_distribution) – provides a natural prior for those probabilities. For example:

```
var ps = dirichlet(Vector([1, 1, 1]))

Categorical({
  ps: T.toScalars(ps),
  vs: ['A', 'B', 'C']
})

// Note: Vector() turns the array [1, 1, 1] into the format required by the WebPPL function dirichlet().

// Note also: We return to the function T.toScalars() below.
```

defines a categorical distribution over 'A', 'B', and 'C', where the probabilities for each have been drawn from a Dirichlet with parameter $\alpha = [1, 1, 1]$. To understand α , it's helpful to realize that just like many other distributions we are familiar with (e.g., Gaussian), the Dirichlet distribution is really a family of distributions defined by parameters. For a Gaussian, the parameters are the mean and standard deviation. For Dirichlet, it is a vector $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]$ where n is the number of categories. You can think of α as a kind of prior on the categories:

x	
var dir = function(v) {	
var d = dirichlet(Vector(v))	
print(d)	
}	
print("alpha = [5, 1, 1]")	
repeat(10, function() {dir([5,1,1]}))	
print("alpha = [1, 5, 1]")	
repeat(10, function() {dir([1,5,1]}))	

```
print("alpha = [1, 1, 5]")
```

```
repeat(10, function() {dir([1,1,5])})
```

```
// Note that `dirichlet()` returns a sample from a Dirichlet distribution. If you want the distribution itself, use `Dirichl
```

run ▼

In many cases, we don't have any prior beliefs, so we set $\alpha = [1, 1, 1, \dots]$. WebPPL provides a convenience function `ones(x,y)`, which returns an array of ones of dimensions `x, y`.

```
print(ones([5,1]))
```

```
dirichlet(ones([5,1]))
```

run ▼

Finally, `Categorical` requires a vector of probabilities, whereas `dirichlet()` actually returns an object with several properties, only one of which is the probabilities we want. The WebPPL function `T.toScalars()` extracts the probabilities for us:

```
var dir = function(v) {
```

```
  var d = dirichlet(v)
```

```
  print(T.toScalars(d))
```

```
}
```

```
print("alpha = [1, 1, 1]")
```

```
repeat(10, function() {dir(ones([3,1])})})
```

run ▼

Thus, the following code samples from a categorical distribution over 'A', 'B', and 'C', with probabilities drawn from a Dirichlet with the uninformative prior `[1, 1, 1]`:

```
var ps = dirichlet(ones([3,1]))
```

```
var acat = function () {Categorical({
```

```
  ps: T.toScalars(ps),
```

```
  vs: ['A', 'B', 'C']
```

```
}}}
```

```
sample(acat())
```

