## Tokenizing a univariate time series using 1D convolution kernels

## Model

is the scalar value of the data in a univariate time series at time .

is 2D matrix containing K basis functions (i.e. convolution kernels). These are the tokens.

is a categorical variable with class labels . These are the token labels.

has the prior

Where is the pdf for a categorical distribution.

## Inference

We take point estimates for all parameters, except we will be fully Bayesian on :

Where is looked-up from the data using amortized inference

We can estimate all parameters by minimising the variational free energy

Here we are after a *full* unregularized, representation of the data via a tokenisation. In short, we are aiming for 100% explained variance, and so are happy to overfit and do not want to regularise the parameters at all. As a result, we ignore the terms that control for overfitting, i.e. the KL divergence term in , and use the following cost function

Finally, doing the integral over in the first term via a summation over samples, , from gives

Where we obtain samples, , from using the Gumbel softmax parameterization.

Note that this cost function can also be written as:

Where corresponds to the model

Where is a vector and is a ( vector, where

## Annealing

To help with convergence we do not start by using the categorical model specified above. Instead, we start with a linear mixture of tokens (sometimes called a *partial volume model*). The key difference is that rather than using , which is binary and one-hot, we instead use a new variable which describes linear combinations of the convolution kernels, i.e.

Where is a vector and

This allows for more movement around the parameters space to identify and learn the best candidate convolution kernels, before we enforce selection of one kernel at each timepoint that occurs when we switch to the full model.

To infer on this *partial volume model*, we can use point estimation on all parameters by using a cost function that minimises the negative log likelihood

Where, as in the main model, is looked-up via amortized inference from the data

The annealing is implemented by using the scalar annealing temperature, , to adjust the relative contributions of and

Where is used in the cost function

Where corresponds to the model

Note that if then

Whereas if then

Hence by starting with and moving gradually to , we can move from optimising the *partial volume model* to the full categorical model we actually want to infer on. Crucially, the learnt convolution kernels, , and the RNN weights in are shared between the two models.