Conditional Density Estimation

Overview of Conditional Density Estimation

- The idea behind conditional density estimation is to estimate densities of the form $p(y|x)$, as opposed to marginal densities of the form $p(y)$ – comes up a lot in posterior estimation, Bayesian estimation etc.
- X is the input variable – can be scalar, vector, categorical, continuous, mixture; for any x fixed, we have $\int p(y|x)\,dy = 1$
- Traditional methods for conditional density estimation include:
  - Parametric methods – conditional distribution follows a standard parametric distribution, estimate the parameters by MLE f.ex.
  - Non-parametric methods – these include KDE, nearest neighbours, splines, local polynomial regression
  - ML based methods – use normalizing flows, VAEs, GANs etc to directly or indirectly estimate the conditional density
- Metrics for training/evaluation of the models – maximise log-likelihood or KL divergence are the most common ones

How is the Data Parsed into the Network?

- Network takes in input of vectors of a specified dimension – parsing the data often requires standardisation or normalisation as needed for stability of the training
- Network needs to incorporate X into its structure to model dependency of Y on X: there are various ways of doing this:
  - Direct concatenation of the data – v basic
  - Feature conditioning/conditional embedding – use networks to extract embeddings of X, use this to modulate intermediate layers of the network used to predict Y
  - Conditional batch normalisation – use X to control the parameters of the batch normalisation
  - Attention mechanisms to focus on specific parts of the X
- Based on this we get an output vector Y which is either processed directly, from parameters generated as functions of X, or implicitly – the normalizing flow example of this is to pass through a base density through a series of invertible transformations parameterised by vectors, z_X, which are generated by passing X through an encoder
- Flow of the data through the network: prepare the data by normalising/embedding; condition X somewhere in the network; use this conditioned data to estimate posterior; compute losses and train

Conditional Density Estimation in SBI

- In SBI, for posterior estimation, the quantity we are usually interested in is $p(\theta|x)$ – which is a conditional density
- The neural networks that we use to go from the simulated data to the posterior is the conditional density estimator
- The major SBI methods all estimate conditional densities in a slightly different way:
  - LFI/Posterior matching – there is a density estimator, $q_\phi(\theta|x)$, which is parameterised by a NN and directly conditioned on the data, that approximates the posterior – the loss function is KL divergence or log-likelihood
  - SNRE-A – uses a classifier with data inputs (the conditioning) to estimate the likelihood-to-prior ratio by identifying if data comes from simulator or proposal, which is trained using cross-entropy loss
  - Normalizing flows – as above, parameterise the transformation functions
  - SNPE – the estimated posterior $q_\phi(\theta|x)$ is trained on the conditioned data and is used to sample new proposals for θ
- All the conditioning above is often through embedding networks which produce conditioned vectors that are passed into the conditional density network