Notes on "Optimal Bayesian Design for Model Discrimination via Classification"

- Setup
  - K candidate statistical models, one of them is true
  - Models indexed by M, a random variable in $\{1, \ldots, K\}$
  - For $M = m$, model is $p(y|\theta_m, m, d)$
    - $y \in \mathcal{Y}$ is data vector
    - $\theta_m \in \Theta_m$ is parameter vector
    - $d \in \mathcal{D}$ is design vector, the variables governing the experiment
  - Prior on θ_m is $p(\theta_m|m)$, given for each possible model
  - Prior on m is $p(m)$
- Picking d. Can use loss function that might depend on m, θ_m, and y, write it as

$$l(d) = E_{\theta_m, y, M|d}[l(d, \theta_m, y, M)].$$

  Optimal d is $d^* = \arg\min_{d \in \mathcal{D}} l(d)$. We want to use an l that captures entropy around the distribution of M (which we then minimize). So we want to pick a d, which produces data y, that leads to the most information about the distribution of M. This will help us pick the correct model.

- Choice of loss function.
  - Shannon entropy (MD stands for multinomial deviance).

$$l_{MD} : \begin{cases} \mathcal{D} \times \mathcal{Y} \to [0, \infty) \\ (d, y) \mapsto -\sum_{m=1}^{K} p(m|y, d) \log p(m|y, d) \end{cases}.$$

    y is observed after the experiment, so we don't actually know l_{MD}. So we'll integrate it out: they rewrite l_{MD} as

$$\begin{aligned} l_{MD}(d) &= -\int_y p(y|d) \sum_{m=1}^{K} p(m|y, d) \log p(m|y, d) dy \\ &\overset{\text{Bayes' Rule}}{=} -\sum_{m=1}^{K} p(m) \int_y p(y|m, d) \log p(m|y, d) dy. \end{aligned} \tag{1}$$

  - Misclassification error rate. Use a loss matrix for all combinations of true and selected models:

$$\begin{aligned} l_{01}(d) &= \int_y p(y|d) \sum_{m=1}^{K} p(m|y, d)\{1 - \mathrm{I}[\widehat{m}(y|d) = m]\} dy \\ &= \int_y p(y|d)\{1 - p[\widehat{m}(y|d)|y, d]\} dy, \end{aligned} \tag{2}$$

    where $\widehat{m} : \mathcal{Y} \to \{1, \ldots, K\}$ is a classifier.

The loss functions above can be hard to compute analytically, so we can use Monte Carlo integration, by sampling from $p(m|y,d)$. Some common issues:

- Need to draw a lot of samples from $p(m|y,d)$
- $p(m|y,d)$ is not always available and needs to be estimated
- The likelihood $p(y|\theta_m,m,d)$ is intractable
- Need a lot of data/Monte Carlo samples for reasonable accuracy

Some remedies:

- using conjugate priors so integrals can be computed analytically
- quadrature
- sequential Monte Carlo (only applicable to sequential experimental designs, where design space is small)
- Gaussian-based posterior approximation
- ABC

Some shortcomings of ABC are addressed by classification method.

- Classification approach. Consider the Monte Carlo integral of (2):

$$\widehat{l}_{01}(d) = 1 - \sum_{m=1}^{K} p(m) \frac{1}{J} \sum_{j=1}^{J} \mathrm{I}[\widehat{m}(y^{m,j}|d) = m], \tag{3}$$

where $y^{m,j} \sim p(y|m,d)$ for $j \in \{1,\ldots,J\}$, $m \in \{1,\ldots,K\}$, and $\widehat{m}$ is the Bayes classifier, which depends on $p(m|y,d)$. As before, this has a few issues:

- Optimizing this loss function, we may need a high number of replicates J to estimate l with low variance.
- $p(y|m,d)$ may need to be estimated before sampling $y^{m,j}$.
- $p(m|y,d)$ is hard to estimate when the likelihood $p(y|\theta_m,m,d)$ is difficult to compute.

Consider the following scheme: consider a sample $\mathcal{T} = \{m^j, y^j\}_{n=1}^{J}$ from

$$p(y,m|d) = \int_{\theta_m} p(y|\theta_m,m,d)p(\theta_m|m)p(m)d\theta_m,$$

then using $\mathcal{T}$ to train a classifier $\widehat{m}_C : \mathcal{Y} \to \{1,\ldots,K\}$ using $\mathcal{T}$. Using this classifier, and new data $\mathcal{T}_* = \{m_*^j, y_*^j\}_{j=1}^{J_*}$, the analogue of (3) would be

$$\widehat{l}_{01}(d) = 1 - \frac{1}{J_*} \sum_{j=1}^{J_*} I[\widehat{m}_C(y_*^j|d,\mathcal{T}) = m_*^j].$$

For intractable likelihoods, the number of samples J and J^* to approximate l_{01} are smaller than required by ABC. This scheme just requires being able to efficiently simulate from the K models. The misclassification error rate may not be well estimated, but we just require that the designs are ranked correctly according to $\hat{l}_{01}$. Classification methods can also be adapted to l_{MD} in (1), but estimates of the posterior over models can be poor. But we'll just focus on constructing a classifier that is order-correct, that is it assigns $\arg\max_m p(m|y,d)$ for each $y \in \mathcal{Y}$. That way, the misclassification rate for a classifier that is order correct except on a small subset of $\mathcal{Y}$ will be still be close to the Bayes error rate.

  - We use CART and not random forests since their posterior estimates $\widehat{p}$ may be 0 and thus the multinomial deviance would be ∞.
  - Since the loss function is evaluated many times, may need to trees just trained on training set, which may lead to over-fitting
    * Again we just need the ranking of the designs to be correct

- Examples