Thesis Notes / Drafts

Devin Ti Tze Hong

May 2023

Contents

Chapter 1

Partition Regression Models

Partition Models

This section introduces random partition models. Specifically we are interested in a class of partition models called non-exchangeable random partition models. We focus particularly on the the Ewens Pitman Attraction which is a random partition model indexed by pairwise similarity between data points.

Random partition models are intimately connected with Bayesian non-parametric sampling models. Consider the model for data $y_1, ..., y_n$ given by:

$$y_i|\theta_i \sim p(y_i|\theta_i) \tag{1.1}$$

$$\theta_i|F \sim F \tag{1.2}$$

$$F \sim Q \tag{1.3}$$

For example, the dirichlet process mixture model, consists of a Dirichlet process prior for Q. Each sample F from the dirichlet process prior produces an atomic distribution. Each data point is associated with a sample θ_i from F, with the implication (explain more) that multiple data points will have the same parameter value θ_i (i.e. they are clustered together). Lastly the sampling model is $p(y_i|\theta_i)$ can be chosen as a normal distribution, implying that θ_i consists of a mean vector and covariance matrix.

For a finite n we can reparamatize this model in terms of the induced partition as well as a set of cluster specific parameters associated with each cluster in the partition. That is $(\theta_1, ..., \theta_n)$ are q_n unique values of θ, $\theta = (\theta_1, ..., \theta_{q_n}$.

EPA Distribution

The EPA distribution is a distribution over partitions indexed by a pairwise similarity function, a mass parameter α and a discount parameter δ. That is, suppose we have points $X_1, ..., X_n$ and a similarity function $\lambda(X_i, X_j)$. Denote π_t as partition of the dataset of the first t data points. The distribution is then given by the recursive formula:

$$P(\pi_n|\alpha, \delta, \lambda, \sigma) = \prod_{t=1}^{n} P_t(\alpha, \sigma, \lambda, \pi(\sigma_1, ..., \sigma_{t-1})) \tag{1.4}$$

Where each $P_t(.)$ is given by.

$$\begin{aligned} P_t(\alpha, \sigma, \lambda, \pi(\sigma_1, ..., \sigma_{t-1})) &= P(\sigma_t \in S|\alpha, \delta, \lambda, \pi(\delta_1, ..., \delta_{t-1}) \\ &= \begin{cases} \frac{t-1-\delta q_{t-1}}{\alpha+t-1} \frac{\sum_{\sigma_s \in S} \lambda(\sigma_t, \sigma_s)}{\sum_{s=1}^{t-1} \lambda(\sigma_t, \sigma_s)}, & \text{if } S \in \pi(\sigma_1, ..., \sigma_{t-1}) \\ \frac{\alpha+\delta q_{t-1}}{\alpha+t-1}, & \text{if } S \text{ in a new subset} \end{cases} \end{aligned} \tag{1.5}$$

q_{t-1} is the number of clusters in the $t-1$ partition. σ refers to the ordering of the data, which is not necessarily the same as the order in the dataset. That is the model is not exchangeable, this is an important fact that has to be accounted for when doing gibbs sampling. Looking at the equation, we can see that the EPA distribution allocates items sequentially, with probabilities proportion to the total similarity between an item and items currently allocated in a cluster, at the same time, there is also a chance that the distribution allocates the item to a new cluster, which is more likely if the similarities to existing clusters are small.

The similarity function we use is the exponential similarity function, given by:

$$\lambda(X_{\lambda_s}, X_{\lambda_t}) = \exp\{-\tau \|X_{\lambda_s} - X_{\lambda_t}\|\} \tag{1.6}$$

Which is similar to the popular radial basis function but paramatized by a precision term τ instead of the usual $\frac{1}{2\sigma^2}$. We take τ to be a hyperparameter, but it can generally be taken as a parameter to be inferred as well.

EPA Regression Model

The modelling approach is a non-parametric regression model for the observations Y_i given the treatment assignments D_i and covariates (or possibly confounders) X_i. Each observation Y_i is modelled as a linear function of D_i, X_i and possible interaction effects. That is:

$$Y_i = \phi_{0i} + \phi_{1i} D_i + \phi_{2i} X_i + \epsilon \tag{1.7}$$

Where $\epsilon \sim N(0, \sigma^2)$. Letting ϕ_i be the regression coefficient vector for the i^{th} data point, $\phi_i = (\phi_{0i}, \phi_{1i}, \phi_{2i})$, we have that ϕ_i is taken as the cluster specific parameter vector, that is, where Z_i is the cluster index of the i^{th} data point:

$$\phi_i = \sum_{k=1}^{q_n} \tilde{\phi_k} 1_{\{Z_i = k\}} \tag{1.8}$$

Each $\tilde{\phi_k}$ is drawn independently from a multivariate normal prior, where the prior mean and covariance serve as hyperparameters.

Model Inference Inference proceeds by way of a Gibbs - Sampler over the parameters π, ϕ, α, δ and β. For π the Gibbs sampler iterates over each data point, proposing for each data point a possible move to other existing clusters, or starting a new cluster. This is given by :

$$P(i \in S_j^{-i} | .) \propto P(\pi_n^{i \to j} | \alpha, \delta, \lambda, \sigma) P(Y_i | \tilde{\phi_j}) \quad \text{,for } j = 0, 1, ..., q_n \tag{1.9}$$

Where S_j^{-i} refers to j^{th} subset but without point i and $\pi_n^{i \to j}$ is the partition obtained by moving point i to cluster S_j^{-i}.

Each $\tilde{\phi_k}$ is updated only by points allocated to its cluster k. With our normal likelihood and normal prior over $\tilde{\phi_k}$ in 1.7, the updates are conditionally conjugate given the partition. That is:

$$\tilde{\phi_k} | \pi_n, \alpha, \delta, \beta, \sigma \sim N(\tilde{\mu}_k, \tilde{\Sigma}_k). \tag{1.10}$$

Where $\tilde{\mu}_k = \tilde{\Sigma}_k \left(\frac{K^T y_k}{\sigma^2} + \Sigma_0^{-1} \beta_0 \right)$ and $\tilde{\Sigma}_k = (\frac{K^T K}{\sigma^2} + \Sigma_0^{-1})^{-1}$. Where K is the regression independent variables from cluster k and y_k the outcomes of points in cluster k.

We refer readers to [**?**] for details on sampling α, β and σ, generally these are sampled using a metropolis in gibbs scheme, where we take a metropolis-hasting step to sample these parameters. For proposal distributions, we generally use random walks. Note that for for σ we use a uniform prior and a uniform proposal.

Chapter 2

Quantile Regression

Simultaneous Linear Quantile Regression

Idea: Want to simultaneously define a model which captures all possible quantiles. Hence we need a model with generates a model for each input quantile τ. To do this specify quantile model $Q(\tau|x)$ as:

$$Q_Y(\tau|x) = \beta_0(\tau) + x'\beta(\tau)$$

That is coefficients β_0 and β are functions of the input quantile τ. Key challenge is how to specify $\beta_0(\tau)$ and $\beta(\tau)$ such that the resulting generated quantile functions are always valid.

β specification Firstly the quantile function is re-specified as:

$$Q_Y(\tau|x) = \mu + \gamma x + \frac{1-x}{2}\eta_1(\tau) + \frac{1+x}{2}\eta_2(\tau)$$

Thus we need priors on:

1. μ: Scalar
2. γ: Scalar
3. $\eta_1(\tau)$: Function
4. $\eta_2(\tau)$: Function

μ and γ can be given typical priors (?). For $\eta_1(\tau)$ and $\eta_2(\tau)$ we have to use priors over functions. The prior used is a logistic transformed gaussian process.

Specifying η We specify:

$$\eta_1(\tau) = \sigma_1\tilde{Q}(\xi_1(\tau))$$
$$\eta_2(\tau) = \sigma_2\tilde{Q}(\xi_2(\tau))$$

Where σ_1 and σ_2 are scalars. $\tilde{Q}$ is the quantile function of a chosen distribution, eg normal or t.
For $\xi_i(\tau)$. Let $w(i,\tau)$ be GP defined on $\{1,2\} \times [0,1]$ with:

- Mean function 0
- Covariance : $Cov(w(i,\tau), w(i',\tau')) = \kappa^2 c_{ii'} \exp\left(-\lambda^2(\tau-\tau')^2\right)$.

Then:

$$\xi_i(\tau) = \frac{\int_0^\tau e^{w(i,t)}dt}{\int_0^1 e^{w(i,t)}dt}$$

To operationalise this prior, we evaluate $w(i,t)$ at a fixed grid of values $\{0.1, 0.2, ..., 1.0\}$ giving 100 evaluation of each $w(i,t)$ at various points t. That is we can draw a 200 dimensional sample j:

$$w^{(j)} \sim N(\mathbf{0}, K)$$

Where $\mathbf{0}$ is 200 dimensional 0 vector, and K is the covariance matrix for the set of values $\{(1, 0.1), (1, 0.2), ...$ $(1, 1.0), (2, 0.1), (2, 0.2), ..., (2, 1.0)\}$. Then for example, the first dimension of $w^{(j)}$ represents a sampled value for $w(1, 0.1)$.
With the vector of values representing $w(i,t)$ we can easily calculate $\xi_i(\tau)$ using trapezoidal approximations for both the numerator and the denominator. Letting $\zeta_i(\tau) = e^{w(i,\tau)}$. We have for a given τ:

$$\hat{\xi_i}(\tau) = \frac{(\tau - t_{l-1})\xi_i(t_l) + (t_l - \tau)\xi_i(t_{l-1}) - (\tau - t_{l-1})(t_l - \tau)(\zeta(t_l) - \zeta(t_{l-1}))}{t_l - t_{l-1}}$$

Where t_l and t_{l-1} are grid points such that $\tau \in [t_l, t_{l-1}]$.

For $\xi_i(t_{l-1})$ and $\xi_i(t_l)$, which are evaluated on grid points, we evaluate the values based on trapezoidal approximation given by:

$$\int_a^b f(x)dx = \frac{b-a}{2n}[f(a) + 2f(x_1) + 2f(x_2) + ... + 2f(x_{n-1} + f(b)]$$

Likelihood Evaluation Key to doing MCMC is likelihood evaluation of the model. We have:

$$\sum_i \log f_Y(y_i|x_i) = -\sum_i \log \frac{\partial}{\partial \tau} Q_Y(\tau_{x_i}(y_i)|x_i)$$

Implementation of Algorithm

Discretization of $w^{(j)}$

GP Approximation

It is computationally intensive to sample the full L dimensional w_1 and w_2 instead as in (cite) we consider a knot-based approximation. This requires us to sample the GPs at K predetermined knot points where KL.

MCMC Algorithm Sampling of the model proceeds by way of a block metropolis in gibbs sampler. We consider 4 blocks. Blocks 1 and 2 are K dimensional blocks consisting of samples of knot points for $w^{(1)}$ and $w^{(2)}$ respectively. Block 3 is a 4 dimensional block of other parameters μ γ, σ_1, σ_2. Lastly block 4 is a combined block of blocks 1,2 and 3, i.e. a 22 dimensional block consisting of $w^{(1)}, w^{(2)}, \mu, \gamma, \sigma_1, \sigma_2$. At each iteration of the sampler, we pick one block at random and update parameters listed in the block. These updates are done via an adaptive metropolis sampler.

1. Sample $R_c \sim Cat(9, 9, 4, 22)$. R_c determines for sampling iteration c which of the 4 blocks are updated. R_c is sampled directly proportion to their associated block sizes.

2. Let $P_c^{R_c}$ be the parameters associated parameters from chosen block R_c. Then update $P_c^{R_c}$ as follows.

3. Firstly sample a proposal for the chosen block.

$$P_{proposal}^{(R_c)} \sim N(P_{c-1}^{(R_c)}, \lambda_c^{(R_c)} \Sigma^{(R_c)}) \tag{2.1}$$

4. Generated implied Quantile function Q, with all parameters in block R_c changed to the proposed value, producing $Q_{proposal}$. $Q_{proposal}$ is a N by L matrix, that is it is sized as the number of data points by the size of the τ grid. Each row of $Q_{proposal}$ corresponds to $Q_(x|\tau)$, that is the τ^{th} quantile of the distribution at value x.

5. Compute the likelihood under proposed $Q_{proposal}$. Using formula:

$$\sum_i \log f_Y(y_i|x_i) = -\sum_i \log \frac{\partial}{\partial \tau} Q_Y(\tau_{x_i}(y_i)|x_i)$$

and thus:

$$f_Y(y_i|x_i) = -\frac{\partial}{\partial \tau} Q_Y(\tau_{x_i}(y_i)|x_i)$$

Where: $\tau_{x_i}(y_i)$ is the quantile of of $f_Y(y_i|x_i)$ that produces y_i. This can be approximated as follows.

   (a) Given x_i, extract corresponding row in Q matrix. Let the corresponding L dimensional row be Q_{x_i}

   (b) Find index t_l such that $y_i \leq Q_{x_i}[t_l]$. t_l corresponds to the index of the τ grid closest to the value $\tau_{x_i}(y_i)$.

   (c) Approximate derivative as:
$$\frac{Q_{x_i}[t_l] - Q_{x_i}[t_l - 1]}{\tau[t_l] - \tau[t_l - 1]}$$

   (d) Importantly, if $t_l > L$ or $t_l = 0$. Then y_i exceeds the boundaries of our current estimated distribution. For these points set the unit log-likelihood to be -inf. We want to reject such parameters.

6. Take a metropolis step based on:

$$TransWeight = \sum_i \log f_{\phi_{Prop}}(y_i|x_i) + \log P_\phi(\phi_{Prop}) + \log P(\phi_{Curr}|\phi_{Prop})$$
$$- \left(\sum_i \log f_{\phi_{Curr}}(y_i|x_i) + \log P_\phi(\phi_{Curr}) + \log P(\phi_{Prop}|\phi_{Curr}) \right)$$

$$\alpha(\phi_{Prop}, \phi_{Curr}) = \min(0, TransWeight)$$

And use e^a as the metropolis acceptance probability. In choosing whether to accept proposal ϕ_{Prop}. Set ϕ_{i+1} as either ϕ_{Prop} or ϕ_{Curr} depending on which is chosen.

7. Now update adaptive metropolis sampler parameters based on:

$$\log(\lambda_{i+1}^c) = \log(\lambda_i^c) + \gamma_{i+1}^c[\alpha(\phi_{Prop}, \phi_{Curr}) - \alpha^*]$$

$$\mu_{i+1}^c = \mu_i^c + \gamma_{i+1}^c(\phi_{i+1}^c - \mu_i^c)$$

$$\Sigma_{i+1}^c = \Sigma_i^c + \gamma_{i+1}^c[(\phi_{i+1}^c - \mu_i^c)(\phi_{i+1}^c - \mu_i^c)^T - \Sigma_i]$$

ϕ^c is the subset of elements in overall parameters vector ϕ that are updated in the current block (as chosen by R_c). For each block we keep track of a block specific set of adaptive metropolis sampler parameters $(\mu^c, \lambda^c, \Sigma^c)$.

   - α^* is a target accept rate which we set at 0.228 for each block.
   - γ^c is a pre-initialized vector of step-sizes. Given by:

$$\gamma_i^c = \frac{C}{i^\alpha}, \quad \text{for} \quad \alpha \in \left((1+\lambda)^{-1}, 1\right]$$

Sampling Transformation for non-GP parameters Due to the gamma distribution on σ_1 and σ_2 these parameters are transformed with an exponential transform from their sampled value in the proposal distribution. That is for block 3 and block 4, instead of sampling σ_1 and σ_2 we instead sample $\log(\sigma_1)$ and $\log(\sigma_2)$. The calculation of the acceptance probability is modified to account for the jacobian of these transformations.

Predictive Performance

Chapter 3

Combining Quantile Regression and EPA

In a classical regression mixture model, the conditional distribution of the response Y given covariates X can be written as:

$$f(Y|X,\theta) = \sum_{j=1}^{K} \pi_j N(y; X\beta_j, \sigma_j^2)$$

θ represents the parameter vector, including $(\pi_1, ...\pi_K)$ the mixture weights, $(\beta_1, ..., \beta_J)$ the regression coefficients and $(\sigma_1^2, ..., \sigma_J^2)$. We have discussed this model as well as its infinite mixture extension in the earlier section on EPA regression.
We now consider an infinite mixture version of the quantile regression model using the non-parametric EPA distribution. Due to the nature of the EPA distribution (Explain better) we are not able to integrate out the partition distribution, instead weights are implicit..

Let

$$Q_Y(\tau|x,\phi) = \mu + \gamma x + \frac{1-x}{2}\eta_1(\tau) + \frac{1+x}{2}\eta_2(\tau)$$

be a quantile regression model as described in earlier section, here we have added ϕ to represent the parameters in the model. That is $\phi = (w_K^{(1)}, w_K^{(2)}, \mu, \gamma, \sigma_1, \sigma_2)$. We can then describe the model. Now for a dataset of size n consider the collection of parameters $\phi_1, ..., \phi_n$, the partition $\pi_n = \{S_1, ..., S_{q_n}\}$, and shared component parameters $\tilde{\phi} = \left(\tilde{\phi}_1, ...\tilde{\phi}_{q_n}\right)$. As before, we have:

$$\phi_i = \sum_{k=1}^{q_n} \tilde{\phi}_k 1_{\{z_i=k\}} \tag{3.1}$$

Prior specifications for each $\tilde{\phi}_k$ vector is the same as described in earlier section on QR.

Inference

We have 3 sets of parameters to infer, the partition π, the EPA parameters $(\alpha, \delta, \beta, \sigma)$ and the component wise quantile regression parameters $\tilde{\phi}_j = (w_{1j}, w_{2j}, \mu_j, \gamma_j, \sigma_{1j}, \sigma_{2j})$. We first describe a gibbs sampling algorithm. This is largely similar to the sampler used in earlier, except with modifications to the likelihood.

1. First sample the partition π_k. This is done by iterating through each data point i and sampling a new cluster index Z_i according to the conditional.

$$P(i \in S_j^{-i}|.) \propto P(\pi_n^{i\to j}|\alpha, \delta, \lambda, \sigma)P(Y_i|\tilde{\phi}_j) \quad \text{, for } j = 0, 1, ..., q_n \tag{3.2}$$

As before in the EPA regression sampler. $P(\pi_n^{i \to j}|\alpha, \delta, \lambda, \sigma)$ is the EPA probability distribution evaluated for a partition where the i^{th} data point is moved to the j^{th} cluster.

$P(Y_i|\tilde{\phi_j})$ is the unit-loglikelihood of Y_i given the QR parameters ϕ_j for cluster j. This can be similarly as:

$$f_Y(y_i|x_i) = -\frac{\partial}{\partial \tau} Q_Y(\tau_{x_i}(y_i)|x_i) \tag{3.3}$$

Which can be calculated via the approximate derivative discussed earlier.

2. For each cluster j update $\tilde{\phi_j}$. This done via the block metropolis in gibbs sampler described in earlier section.
   - We take N_k metropolis steps for each $\tilde{\phi_j}$ and take the last sampled $\tilde{\phi_j}$, as the updated value.
   - At each gibbs step, we re-initialize all the adaptive metropolis parameters.
3. Update EPA distribution parameters $(\alpha, \delta, \beta, \sigma)$.

Problems

1. -inf in likelihood → cannot transition. For example if initial value and proposed value both have LL as infinity (so transition probability not well defined) then how to transition
   - Soln 1: Take the proposed value (with some high probability?) for the first few sampling iterations → can explore and get out of the poor region
   - Soln 2: Move based on prior/proposal probabilities only. → Works slow in practice.
2. -inf in likelihood. Leads to formation of new clusters. For example suppose we have two clusters both with QRs that have -inf likelihood. This implies that probability of assignment to those clusters is 0.
   - If new cluster proposes ϕ that does not produce -inf point likelihood → then guaranteed allocation to the new cluster.
   - Problem with singleton cluster → QR does not update well with just 1 data point. This implies unit likelihood for small cluster QR will be -inf for many data points, thus no points will be allocated to them.
3. Split-Merge seems unlikely to solve issue as well.

Possible Solutions

1. Modify likelihood to not include infinity? → need a better signal for out of limit points. Else chain is moving aimlessly or gets stuck. (https://discourse.mc-stan.org/t/implementing-quasi-bayes-with-adaptive-covariance-estimation/25644)

Split-Merge

The gibbs sampler algorithm has 2 issues.

1. The need to cycle through every data point results is computationally expensive
2. The updates can be slow, when gibbs sampler ...

We have seen the use of the non-conjugate split-merge sampler for EPA regression, and saw how that led to better convergence. The vanilla split-merge is:

Restricted Gibbs Split-Merge

1. Randomly and uniformly select a pair of indices i and j from $\{1, \ldots, n\}$.
2. If i and j belong to the same component in π, propose a split move to obtain π^*:
   (a) Let S be the component containing i and j.
   (b) Remove indices i and j from S to form temporary singleton sets $S_i = \{i\}$ and $S_j = \{j\}$, and let $S^* = S \setminus \{i, j\}$.
   (c) Randomly and uniformly permute the indices in S^*, and add successive items k from this permutation to S_i with probability:
   $$\Pr(k \in S_i|S_i, S_j, y) = \frac{(|S_i| - \delta)p(y_k|y_{S_i})}{(|S_i| - \delta)p(y_k|y_{S_i}) + (|S_j| - \delta)p(y_k|y_{S_j})}. \quad (20)$$
   Otherwise, add k to S_j. Note that the cardinality of S_i and S_j will change with each successive allocation, and that H_{S_i} and H_{S_j} evolve accordingly. Remove S^* (which is now empty) to obtain π^*.
   (d) If $t = 0$, set $\pi_{\text{launch}} = \pi$. Skip to Step 2f.
   (e) If $t > 0$, perform $t + 1$ restricted Gibbs scans of the indices among S_i and S_j to obtain π^* and π_{launch}, as described in Steps 2e-f of Section 3.2.
   (f) Calculate $a(\pi^*, \pi_{\text{launch}})$. If $a(\pi^*, \pi_{\text{launch}}) > U$, where $U \sim \text{Uniform}(0, 1)$, then accept π^*; otherwise, reject the proposal. Since only the proposal scheme has changed, most of the calculations for this ratio are equivalent to those described above in Section 3.2. However, if $t = 0$, $q(\pi^*|\pi_{\text{launch}})$ is the product over the probabilities computed from (20). If $t > 0$, then $q(\pi^*|\pi_{\text{launch}})$ is comprised of the product of transition probabilities in the final restricted Gibbs update. In both cases, $q(\pi_{\text{launch}}|\pi^*) = 1$.
3. If indices i and j belong to different components in π, propose a merging of the components to obtain π^*:
   (a) Let S_i and S_j denote the components of π that contain the indices i and j, respectively.
   (b) Form the merged component $S = S_i \cup S_j$.
   (c) Remove S_i and S_j from π and add the newly merged S to it to obtain the proposal state π^*.
   (d) If $t = 0$, obtain a hypothetical split state by generating a random permutation of $S \setminus \{i, j\}$, and sequentially allocating each item to be with either i or j using (20). Let π_{launch} be the set partition that results from this allocation, and skip to Step 3f.
   (e) If $t > 0$, obtain an initial random split state by generating a random permutation of $S \setminus \{i, j\}$, and sequentially allocating each item to be with either i or j using (20). Perform a further t restricted Gibbs scans of the indices among S_i and S_j to obtain a hypothetical π_{launch} as described in Section 3.2 Step 3e.
   (f) Calculate $a(\pi^*, \pi)$. If $a(\pi^*, \pi) > U$, where $U \sim \text{Uniform}(0, 1)$, then accept π^*; otherwise, reject the proposal. If $t = 0$, then $q(\pi_{\text{launch}}|\pi^*)$ is the product over the probabilities computed using (20) to arrive at the hypothetical launch state. If $t > 0$, $q(\pi_{\text{launch}}|\pi^*)$ is comprised of the product of transition probabilities in the final restricted Gibbs update to the hypothetical launch state, as in the RGMS sampler. In both cases, $q(\pi^*|\pi_{\text{launch}}) = 1$.

Notes Key to this algorithm is that the model is conjugate meaning that we can integrate out any parameters. The key thing this allows us to do is to **not** have to sample a new parameter vector every time we propose a split or merge. There is a variant of this algorithm NUSAMs that uses approximately closeness between data points to do gibbs updating which could work.

Non-Conjugate Split-Merge

We describe Non-Conjugate Split-Merge for the Normal - Inverse Gamma EPA - Regression model. The normal - inverse gamma regression model falls into a class of 'conditionally' conjugate models. There are two parameters μ_j the regression mean of the j^{th} cluster and σ^2 the error variance. Let $\phi = (\mu, \sigma^2)$, represent these parameters. Then the state of the Markov chain consist of $\gamma = (c, \phi)$ where $c = (c_1, \ldots, c_n)$ and $\phi = (\phi_c : c \in \{c_1, \ldots, c_n\})$.

1. **Random Selection** Select two distinct observations, i and j, at random uniformly.
   - Let S denote the set of observations, $k \in \{1, \ldots, n\}$, for which $k \neq i$ and $k \neq j$, and $c_k = c_i$ or $c_k = c_j$. That is S is the set of points originally clustered with i and j.
2. **Define launch states**. We define two launch states, $\gamma_{L\text{split}}$ and $\gamma_{L\text{merge}}$, that will be used to define Gibbs sampling distributions required for the split and merge proposals.
   - Obtain launch state $\gamma_{L\text{split}} = (c^{L\text{split}}, \phi^{L\text{split}})$ as follows:
     - If $c_i = c_j$, then let $c^{L\text{split}_i}$ be set to a new component such that $c^{L\text{split}_i} \notin \{c_1, \ldots, c_n\}$ and let $c_j^{L\text{split}} = c_j$. Otherwise, when $c_i \neq c_j$, let $c_i^{L\text{split}} = c_i$ and $c_j^{L\text{split}} = c_j$. For every $k \in S$, randomly set $c_k^{L\text{split}}$, independently with equal probability, to either of the distinct components, $c_i^{L\text{split}}$ or $c_j^{L\text{split}}$. Initialize model parameters, $\phi^{L\text{split}}_{c_i^{L\text{split}}}$ and $\phi^{L^{\text{split}}}_{c_j^{L\text{split}}}$, associated with the two distinct components by drawing new values from their prior distribution.
     - Modify $\gamma^{L\text{split}}$ by performing t intermediate restricted Gibbs sampling scans to update $c^{L\text{split}}$, $\phi^{L\text{split}}_{c_i^{L\text{split}}}$, and $\phi^{L^{\text{split}}}_{c_j^{L\text{split}}}$.
   - Obtain launch state $\gamma^{L\text{merge}} = (c^{L\text{merge}}, \phi^{L\text{merge}})$ as follows:
     - If $c_i = c_j$, then let $c_i^{L\text{merge}} = c_j^{L\text{merge}} = c_j$ (which is the same as c_i). Similarly, if $c_i \neq c_j$, then set $c_i^{L\text{merge}} = c_j^{L\text{merge}} = c_j$. For every $k \in S$, set $c_k^{L\text{merge}} = c_j$. Initialize model parameter, $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$, associated with the merged component by drawing a new value from its prior distribution.
     - Modify $\gamma^{L\text{merge}}$ by performing r intermediate restricted Gibbs sampling scans to update $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$.
3. **Generate Proposal and Metropolis Step**. In this step we generate proposals γ^{prop} based on whether or not points i, j were in the same cluster. We then calculate the metropolis hastings transition probability and conduct a sampling step. These transition probabilities are calculated based on transitioning from the launch states calculated earlier to the proposed state.

   (a) If items i and j are in the **same mixture component**, i.e., $c_i = c_j$, then:

   i. Propose a new assignment of data items to mixture components, denoted as c^{split}, in which component $c_i = c_j$ is split into two separate components, c_i^{split} and c_j^{split}, and propose new values for the corresponding components' parameters, $\phi^{\text{split}}_{c_i^{\text{split}}}$ and $\phi^{\text{split}}_{c_j^{\text{split}}}$. Define each element of the candidate state, $\gamma^{\text{split}} = (c^{\text{split}}, \phi^{\text{split}})$, as follows:
      - Let $c_i^{\text{split}} = c_i^{L\text{split}}$ (note that $c_i^{L\text{split}} \notin \{c_1, \ldots, c_n\}$)
      - Let $c_j^{\text{split}} = c_j^{L\text{split}}$ (which is the same as c_j)
      - By conducting **one** final Gibbs sampling scan from the launch state, $\gamma^{L\text{split}}$, for every observation $k \in S$, let c_k^{split} be set to either component c_i^{split} or c_j^{split} and draw values for the model parameters, $\phi^{\text{split}}_{c_i^{\text{split}}}$ and $\phi^{\text{split}}_{c_j^{\text{split}}}$.
      - For observations $k \notin S \cup \{i, j\}$, let $c_k^{\text{split}} = c_k$, and for $c \notin \{c_i^{\text{split}}, c_j^{\text{split}}\}$, let $\phi^{\text{split}}_{c_{\text{split}}} = \phi_c$.

   ii. Compute the proposal densities, $q(\gamma_{\text{split}}|\gamma)$ and $q(\gamma|\gamma_{\text{split}})$, that will be used to calculate the Metropolis-Hastings acceptance probability.

- Calculate the split proposal density, $q(\gamma^{\text{split}}|\gamma)$, by computing the Gibbs sampling transition kernel from the split launch state, $\gamma^{L\text{split}}$, to the final proposed state, γ_{split}. The Gibbs sampling transition kernel is the product of the individual probabilities of setting each element in the launch state to its final proposed value during the final Gibbs sampling scan.
- Calculate the corresponding proposal density, $q(\gamma|\gamma^{\text{split}})$
- Calculate the corresponding proposal density, $q(\gamma|\gamma^{\text{split}})$, by computing the Gibbs sampling transition kernel from the merge launch state, $\gamma^{L\text{merge}}$, to the original merged configuration, γ. The Gibbs sampling transition kernel is the product of the probability of setting each element in the original merge state (in this case, elements of ϕ_{c_j}) to its original value in a (hypothetical) Gibbs sampling scan from the merge launch state.

iii. Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\gamma^{\text{split}}, \gamma)$. If the proposal is accepted, γ^{split} becomes the next state in the Markov chain. If the proposal is rejected, the original configuration and model parameter, γ, remain as the next state.

(b) Otherwise, if i and j are in **different mixture components**, i.e., $c_i \neq c_j$, then:

i. Propose a new assignment of data items to mixture components, denoted as c^{merge}, in which distinct components, c_i and c_j, are combined into a single component, and propose a new value for the corresponding merged component's model parameter, $\phi^{\text{merge}}_{c_j^{\text{merge}}}$. Define each element of the candidate state, $\gamma^{\text{merge}} = (c^{\text{merge}}, \phi^{\text{merge}})$, as follows:

- Let $c_i^{\text{merge}} = c_i^{L\text{merge}}$ (which is the same as c_j)
- Let $c_j^{\text{merge}} = c_j^{L\text{merge}}$ (which is the same as c_j)
- For every observation $k \in S$, let $c_k^{\text{merge}} = c_j^{L\text{merge}}$ (which is the same as c_j)
- For observations $k \notin S \cup \{i, j\}$, let $c_k^{\text{merge}} = c_k$, and for $c \neq c^{\text{merge}}$, let $\phi^{\text{merge}}_{c_{\text{merge}}} = \phi_c$.
- Conduct one final restricted Gibbs sampling scan from the launch state, $\gamma^{L\text{merge}}$, in order to draw a new value for the model parameter, $\phi c_j^{\text{merge}\text{merge}}$.

ii. Compute the proposal densities, $q(\gamma^{\text{merge}}|\gamma)$ and $q(\gamma|\gamma^{\text{merge}})$, that will be used to calculate the Metropolis-Hastings acceptance probability.

- Calculate the merge proposal density, $q(\gamma^{\text{merge}}|\gamma)$, by computing the Gibbs sampling transition kernel from the merge launch state, $\gamma^{L\text{merge}}$, to the final proposed state, γ^{merge}. The Gibbs sampling transition kernel is the probability of setting $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$ to its final proposed value, $\phi^{\text{merge}}_{c_j^{\text{merge}}}$, via one Gibbs sampling scan.
- Calculate the corresponding proposal density, $q(\gamma|\gamma^{\text{merge}})$, by computing the Gibbs sampling transition kernel from the split launch state, $\gamma^{L\text{split}}$, to the original split configuration, γ. The Gibbs sampling transition kernel is the product of the probabilities of setting each element in the original split state to its original value in a (hypothetical) Gibbs sampling scan from the split launch state.

iii. Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\gamma^{\text{merge}}, \gamma)$. If the proposal is accepted, γ^{merge} becomes the next state. If the merge proposal is rejected, the original configuration and model parameters, γ, remain as the next state.

Non-Conjugate Split-Merge For EPA - QR

The main difference is that we no longer have conditional - conjugacy for the model. That is when $f_Y(y_i|x_i)$ is the quantile regression model. We now have $\phi = (w_1, w_2, \mu, \gamma, \sigma_1, \sigma_2)$ These parameters have to be sampled using metropolis hastings and cannot be sampled (with 100% acceptance rate) from full conditionals.

1. **Random Selection** Select two distinct observations, i and j, at random uniformly.

    - Let S denote the set of observations, $k \in \{1, \ldots, n\}$, for which $k \neq i$ and $k \neq j$, and $c_k = c_i$ or $c_k = c_j$. That is S is the set of points originally clustered with i and j.

2. **Define launch states**. We define two launch states, $\gamma_{L\text{split}}$ and $\gamma_{L\text{merge}}$, that will be used to define Gibbs sampling distributions required for the split and merge proposals.

   - Obtain launch state $\gamma_{L\text{split}} = (c^{L\text{split}}, \phi^{L\text{split}})$ as follows:
     - If $c_i = c_j$, then let $c^{L\text{split}_i}$ be set to a new component such that $c^{L\text{split}_i} \notin \{c_1, \dots, c_n\}$ and let $c_j^{L\text{split}} = c_j$. Otherwise, when $c_i \neq c_j$, let $c_i^{L\text{split}} = c_i$ and $c_j^{L\text{split}} = c_j$. For every $k \in S$, randomly set $c_k^{L\text{split}}$, independently with equal probability, to either of the distinct components, $c_i^{L\text{split}}$ or $c_j^{L\text{split}}$. Initialize model parameters, $\phi^{L\text{split}}_{c_i^{L\text{split}}}$ and $\phi^{L^{\text{split}}}_{c_j^{L\text{split}}}$, associated with the two distinct components by drawing new values from their prior distribution.
     - Modify $\gamma^{L\text{split}}$ by performing t intermediate restricted metropolis in gibbs sampling scans to update $c^{L\text{split}}$, $\phi^{L\text{split}}_{c_i^{L\text{split}}}$, and $\phi^{L^{\text{split}}}_{c_j^{L\text{split}}}$. With $\phi^{L\text{split}}_{c_i^{L\text{split}}}$, and $\phi^{L^{\text{split}}}_{c_j^{L\text{split}}}$ updated with the QR block metropolis sampler
   - Obtain launch state $\gamma^{L\text{merge}} = (c^{L\text{merge}}, \phi^{L\text{merge}})$ as follows:
     - If $c_i = c_j$, then let $c_i^{L\text{merge}} = c_j^{L\text{merge}} = c_j$ (which is the same as c_i). Similarly, if $c_i \neq c_j$, then set $c_i^{L\text{merge}} = c_j^{L\text{merge}} = c_j$. For every $k \in S$, set $c_k^{L\text{merge}} = c_j$. Initialize model parameter, $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$, associated with the merged component by drawing a new value from its prior distribution.
     - Modify $\gamma^{L\text{merge}}$ by performing t block metropolis steps to update $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$.

3. **Generate Proposal and Metropolis Step**. In this step we generate proposals γ^{prop} based on whether or not points i, j were in the same cluster. We then calculate the metropolis hastings transition probability and conduct a sampling step. These transition probabilities are calculated based on transitioning from the launch states calculated earlier to the proposed state.

   (a) If items i and j are in the **same mixture component**, i.e., $c_i = c_j$, then we propose a **split**:

      i. Propose a new assignment of data items to mixture components, denoted as c^{split}, in which component $c_i = c_j$ is split into two separate components, c_i^{split} and c_j^{split}, and propose new values for the corresponding components' parameters, $\phi^{\text{split}}_{c_i^{\text{split}}}$ and $\phi^{\text{split}}_{c_j^{\text{split}}}$. Define each element of the candidate state, $\gamma^{\text{split}} = (c^{\text{split}}, \phi^{\text{split}})$, as follows:
      - Let $c_i^{\text{split}} = c_i^{L\text{split}}$ (note that $c_i^{L\text{split}} \notin \{c_1, \dots, c_n\}$)
      - Let $c_j^{\text{split}} = c_j^{L\text{split}}$ (which is the same as c_j)
      - By conducting **one** final Gibbs sampling scan from the launch state, $\gamma^{L\text{split}}$, for every observation $k \in S$, let c_k^{split} be set to either component c_i^{split} or c_j^{split} and draw values for the model parameters, $\phi^{\text{split}}_{c_i^{\text{split}}}$ and $\phi^{\text{split}}_{c_j^{\text{split}}}$ from the prior.
      - For observations $k \notin S \cup \{i, j\}$, let $c_k^{\text{split}} = c_k$, and for $c \notin \{c_i^{\text{split}}, c_j^{\text{split}}\}$, let $\phi^{\text{split}}_{c_{\text{split}}} = \phi_c$. I.e. leave them as they were.

      ii. Compute the proposal densities, $q(\gamma^{\text{split}}|\gamma)$ and $q(\gamma|\gamma^{\text{split}})$, that will be used to calculate the Metropolis-Hastings acceptance probability.
      - Calculate the split proposal density, $q(\gamma^{\text{split}}|\gamma)$, by computing the Gibbs sampling transition kernel from the split launch state, $\gamma^{L\text{split}}$, to the final proposed state, γ^{split}. The Gibbs sampling transition kernel is the product of the individual probabilities of setting each element in the launch state to its final proposed value during the final Gibbs sampling scan. Problem is that the gibbs transition from $\phi^{L\text{split}}$ to ϕ^{split} cannot be computed since this was made by metropolis hastings.
      - Calculate the corresponding proposal density, $q(\gamma|\gamma^{\text{split}})$, by computing the Gibbs sampling transition kernel from the merge launch state, $\gamma^{L\text{merge}}$, to the original merged configuration, γ. The Gibbs sampling transition kernel is the product of the probability of setting each element in the original merge state (in this case, elements of ϕ_{c_j}) to its original value in a (hypothetical) Gibbs sampling scan from the merge launch state.

iii. Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\gamma^{\text{split}}, \gamma)$. If the proposal is accepted, γ^{split} becomes the next state in the Markov chain. If the proposal is rejected, the original configuration and model parameter, γ, remain as the next state.

(b) Otherwise, if i and j are in **different mixture components**, i.e., $c_i \neq c_j$, then:

i. Propose a new assignment of data items to mixture components, denoted as c^{merge}, in which distinct components, c_i and c_j, are combined into a single component, and propose a new value for the corresponding merged component's model parameter, $\phi^{\text{merge}}_{c_j^{\text{merge}}}$. Define each element of the candidate state, $\gamma^{\text{merge}} = (c^{\text{merge}}, \phi^{\text{merge}})$, as follows:

- Let $c_i^{\text{merge}} = c_i^{L\text{merge}}$ (which is the same as c_j)
- Let $c_j^{\text{merge}} = c_j^{L\text{merge}}$ (which is the same as c_j)
- For every observation $k \in S$, let $c_k^{\text{merge}} = c_j^{L\text{merge}}$ (which is the same as c_j)
- For observations $k \notin S \cup \{i, j\}$, let $c_k^{\text{merge}} = c_k$, and for $c \neq c^{\text{merge}}$, let $\phi^{\text{merge}}_{c_{\text{merge}}} = \phi_c$.
- Conduct one final restricted Gibbs sampling scan And one block metropolis step from the launch state, $\gamma^{L\text{merge}}$, in order to draw a new value for the model parameter, $\phi^{\text{merge}}_{c_j^{\text{merge}}}$.

ii. Compute the proposal densities, $q(\gamma^{\text{merge}}|\gamma)$ and $q(\gamma|\gamma^{\text{merge}})$, that will be used to calculate the Metropolis-Hastings acceptance probability.

- Calculate the merge proposal density, $q(\gamma^{\text{merge}}|\gamma)$, by computing the Gibbs sampling transition kernel from the merge launch state, $\gamma^{L\text{merge}}$, to the final proposed state, γ^{merge}. The Gibbs sampling transition kernel is the probability of setting $\phi^{L\text{merge}}_{c_j^{L\text{merge}}}$ to its final proposed value, $\phi^{\text{merge}}_{c_j^{\text{merge}}}$, via one Gibbs sampling scan. Again we will have problems calculating this when using the MCMC sampler for ϕ
- Calculate the corresponding proposal density, $q(\gamma|\gamma^{\text{merge}})$, by computing the Gibbs sampling transition kernel from the split launch state, $\gamma^{L\text{split}}$, to the original split configuration, γ. The Gibbs sampling transition kernel is the product of the probabilities of setting each element in the original split state to its original value in a (hypothetical) Gibbs sampling scan from the split launch state.

iii. Evaluate the proposal by the Metropolis-Hastings acceptance probability $a(\gamma^{\text{merge}}, \gamma)$. If the proposal is accepted, γ^{merge} becomes the next state. If the merge proposal is rejected, the original configuration and model parameters, γ, remain as the next state.