Leveraging Conjugacy in Dirichlet Process Poisson Mixture Models

STAT 447 Final Project

Caden Hewlett

2024-04-14

Introduction

```
# TODO
```

We will begin with a brief literature review, discussing probability measures and Dirichlet Processes from a theoretical standpoint. In the Data Analysis section, we briefly recap the source data aggregation and imputation, inspect the plot, and provide some summary statistics. We will proceed to detail the stick-breaking process used by the algorithm and formally define both the Bayesian model and hyper-parameter choices for the DPMM. In the Results section, we produce the *a posteriori* results, including the table of weighted posterior rates and the posterior distribution plot. We then briefly interpret these results in the context of the problem domain and discuss shortcomings and future work.

Literature Review

Before we discuss Dirichlet Processes, it is crucial to establish a groundwork in probability measure theory. We will briefly revisit the concepts of σ-algebra and probability measures. In the following, we present generalized definitions which are discussed rigorously in works such as (Billingsley 2012) and (Rudin 1986).

Measure Theory

Let $\mathbb{X}$ be a well-defined sample space. A σ-algebra $\mathcal{F} \subseteq P(\mathbb{X})$ is a set satisfying the following:

1. The entire sample space $\mathbb{X}$ is in $\mathcal{F}$ and $\emptyset$ is in $\mathcal{F}$. This is referred to in the literature as "non-emptiness and universality."
2. For all sets $A \in \mathcal{F}$, the complement $A^c \in \mathcal{F}$. This property is referred to as "closure under complementation."
3. For any countable collection of sets $\{A_i\}_{i \in I}$, where I is a countable index set, if $\forall i \in I, A_i \in \mathcal{F}$ then $\bigcup_{i \in I} A_i \in \mathcal{F}$. This is referred to as "closure under countable unions."

For the sake of this work, we are more interested in *probability measures*, which are built on σ-algebra. A probability measure $\mu : \mathcal{F} \mapsto [0, 1]$ satisfies the following familiar axioms of probability:

1. $\forall A \in \mathcal{F},\ \mu(A) \geq 0$. This is referred to as "non-negativity."
2. $\mu(\mathbb{X}) = 1$, and $\mu(\emptyset) = 0$.
3. For any countable set $\{A_i\} \subseteq \mathcal{F}$ where $\forall i \neq j, A_i \cap A_j = \emptyset$, we have that $\mu\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mu(A_i)$, this is referred to as "countable additivity."

Before we move on to Dirichlet Processes, we acknowledge that the above definitions might be abstract for those new to measure theory. For clarity, the Appendix includes examples to illustrate σ-algebra properties and verify a probability measure μ on a simple finite set.

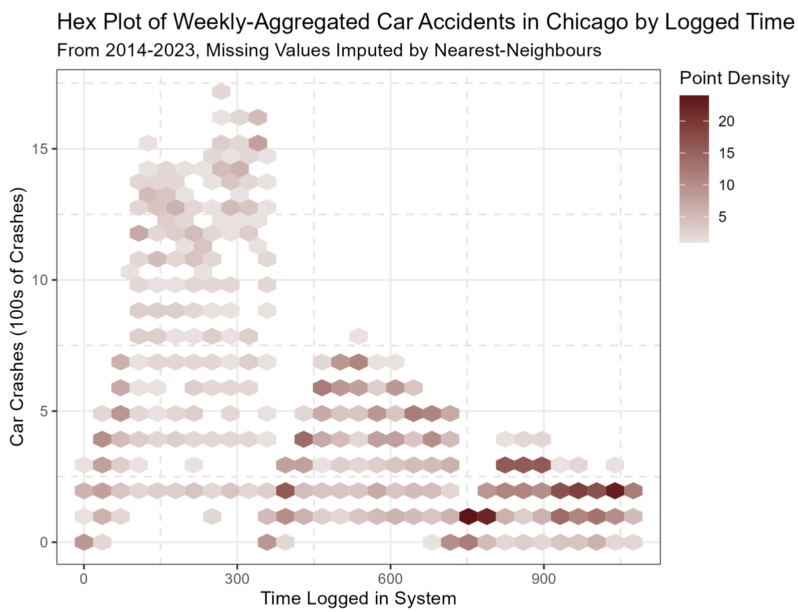
Dirichlett Process: A Distribution Over Measures

Now, we introduce the concept of the Dirichlet Process (DP), a distribution across probability measures. We adopt the description used in (Hannah 2011), where the DP is specified by its base measure $\mathbb{G}_0$ (commonly a distribution) and the concentration parameter, α. A random sample from a Dirichlet Process is a probability measure over a σ-algebra from sample space $\mathbb{X}$, which is often assumed to be countable. The resulting probability measure follows the structure of the base measure $\mathbb{G}_0$ with a degree of deviation controlled by α. A Dirichlet Process can also be thought of conceptually of as a Dirichlet Distribution whose support is an infinite-simplex, rather than a discretely-defined k-simplex (Ferguson 1973).

The α parameter can be thought of as the confidence in the base measure, where a higher α yields greater dispersion and a higher number of clusters of measures about $\mathbb{G}_0$. Conversely, a lower α indicates more confidence in $\mathbb{G}_0$ resulting in fewer, larger clusters (Sethuraman 1994). This property is explored in more detail in the discussion of finite approximation in the Methods section.

Data Analysis and Processing

Now, we will perform some exploratory data analysis and explain the data set used. The data is sourced from the Chicago Police Department (CPD 2024) and contains 794,956 vehicle accident reports from 2014 to 2023. A hexplot of the aggregated results is below.



We applied an aggregation code framework modified from (StackOverflow 2023) in order to coerce the data into weekly crash counts over the time frame. Certain weeks in both 2014 and 2015 had missing counts. In these cases, we performed nearest-neighbours imputation for the missing values. In addition, we grouped the counts by severity to craft an environment in which reports were logged by monetary crash damage, with low-severity accidents ordered first. Furthermore, in an attempt to avoid overflow in training loops we measured crashes in hundreds. As the hexplot shows, there are three distinct categorizations of crash counts, with the count dispersion decreasing across the groups. In addition, as the plot shows, the point density varies across these clusters. These characteristics of the aggregated data create a distinct multimodal plot, which makes these data a good candidate for Bayesian non-parametrics. The code for data processing and plotting are included in the appendix.

Methods

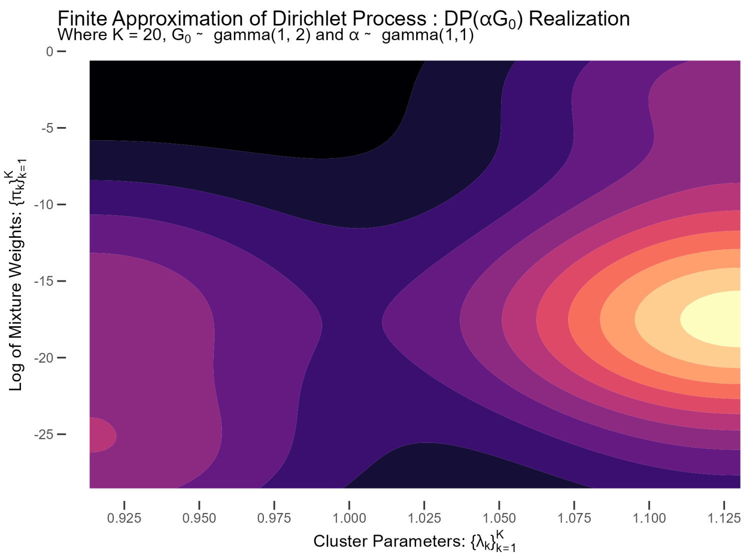
In this section, we define the stick-breaking discrete approximation of the Dirichlet Process implemented in this work and in the Gibbs sampler of the `dirichletprocess` package (Markwick 2023). This library implements the algorithm discussed in-depth in (Neal 2000). As Markwick notes in his work, using a conjugate base prior allows the algorithm to use an optimized method presented by Neal, and hence is generally preferable if fast mixing is desired. However, as of writing, the Gamma-Poisson conjugate pair is not implemented as a default model in the package. Hence, in addition to the prior base measure finite approximation we explicitly define the likelihood, conjugate posterior and posterior predictive for the Gamma-Poisson to be implemented as a mixing distribution object in the DPMM Gibbs sampler. Then, we explicitly state the Bayesian model applied to the data from the previous section.

Finite Approximation

The finite approximation used in this work is known as a "stick-breaking" or Griffiths, Engen, and McCloskey (GEM) process. The purpose of the GEM process in terms of a Dirichlet Process is to generate weights $\{\pi_k\}$, which will be assigned to pulls from $\mathbb{G}_0$ to approximate a sampled measure from $\mathrm{DP}(\alpha\mathbb{G}_0)$.

The idea behind GEM weighing is to take a "stick" with unit length and break it at a location decided by a $\beta_1 \sim \mathrm{beta}(1, \alpha)$ random pull, which we denote π_1. Then, we break the remaining stick length in two by a second $\beta_2 \sim \mathrm{beta}(1, \alpha)$ sample. Hence, $\pi_2 = (1 - \beta_1)\beta_2$, which can be understood as "the remaining stick length after the first break, broken at the second random break location." Then, we generalize this concept for discrete $k = \{1, 2, \ldots, K\} \subseteq \mathbb{Z}$ as follows:

$$\pi \sim \mathrm{GEM}(\alpha) : \text{Let } \beta_k \overset{\text{iid}}{\sim} \mathrm{beta}(1, \alpha), \text{ then } \pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_i) \tag{1}$$



Where π_k can be considered the k-th value returned from the GEM process. The realization is then an estimation of a K-dimensional probability measure. For computational purposes, we treat $\mathrm{GEM}(\alpha)$ as a discrete probability distribution for a reasonably large choice of K, since $\lim_{K\to\infty} \left(\sum_{k=1}^{K} \pi_k\right) = 1$ as noted in (Xing 2014). As mentioned in the literature review, α is the concentration of the finite-approximated Dirichlet Process. With the additional context of the GEM distribution this property becomes more evident. For an arbitrary $\beta_k \sim \mathrm{beta}(1, \alpha)$, a larger α assigns more probability density to lower values in $\mathrm{supp}(\beta_k) = (0, 1)$. This means that the breaks are more likely to happen "earlier on" along each stick, yielding smaller initial clusters and hence more dispersion in the density of pulls from $\mathbb{G}_0$. To contrast, a very low α may result in a β_1 break near 1, implying very low weights for the remaining $\{\beta_k\}_{k=1}^{K-1}$. For the model we apply in this work, we let $\alpha \sim \mathrm{gamma}(1, 1)$ and $\mathbb{G}_0 \sim \mathrm{gamma}(1, 2)$ define $\mathrm{DP}(\alpha\mathbb{G}_0)$. In the figure above, we demonstrate the finite approximation at $K = 20$, where the horizontal axis corresponds to the logarithm of stick-breaking weights $\{\pi_k\}_{k=1}^{N}$ and the vertical axis are the sampled values $\{\lambda_k\}_{k=1}^{N}$ from $\mathbb{G}_0$. The kernel density bin width for contour separation is 0.02, starting from $(0, 0.02]$ and ending at $(0.22, 0.24]$. The full code for the approximation and density plot is included in the appendix. For the full implementation of the DPMM we selected $K = 150$, a choice justified in detail in (Ishwaran and James 2001).

Model Implementation

....

Results

The Gibbs sampling procedure on $10,000$ iterations with a burn-in of $B = 100$ had a total run time of 703.87 seconds, and converged to the posterior rates and distribution consistently across multiple runs and randomization seeds. In the table below, we report the posterior rate estimates and the associated mixing weights.

Table 1: DPMM Posterior Parameters and Weights

Rate λ_k	11.964	3.880	2.034	0.718
Weight π_k	0.198	0.329	0.466	0.007

From the results above, we see that high posterior weight was associate with three distinct rate parameters, indicating that the DPMM correctly identified the different clusters of data, with the majority of the weight mass being placed on the first three rate parameters. The fourth rate parameter with $w_4 \approx 0.718$ had a cluster size of only 8 out of $1,076$ data points indicating that it is likely not significant to the overall mixture model.

Below, we plot the estimated posterior distribution on a sample frame of size $10,000$ alongside the 99% credible interval. We compare these results with the frequentist maximum likelihood estimate of the rate parameter found via Poisson regression, which includes the corresponding 99% confidence interval.

Conclusion and Further Work

```
# TODO: Discuss other unsupported conjugates (such as categorical) as well as the potential to explore
#       non-conjugate mixtures.
```

Appendix

Acknowledgements: Special thanks to Prof. Lasantha Premarathna for sparking my interest in non-parametric statistics!

Section 1: Proofs and Examples

Miscellaneous information such as knowledge on sets, power sets, subsets and countability from (Demirbas and Rechnitzer 2023).

Example of a σ-Algebra

For additional clarity, we provide an example of a σ-algebra $\mathcal{F}$ to demonstrate the properties mentioned in the literature review.

Let's consider the finitely countable and simple set $\mathbb{X} = \{a, b, c\}$.

Directly, $P(\mathbb{X}) = \big\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\big\}$ is the power set of $\mathbb{X}$, where we note $\{a, b, c\} = \mathbb{X}$. Consider $\mathcal{F} = \big\{\emptyset, \{a\}, \{b, c\}, \{a, b, c\}\big\} \subseteq P(\mathbb{X})$.

We will apply the properties discussed in the literature review section to prove that $\mathcal{F}$ is a σ-algebra.

To verify Property 1 (universality and non-emptiness), we note that we can also write $\mathcal{F}$ as $\big\{\emptyset, \{a\}, \{b, c\}, \mathbb{X}\big\}$. From this definition, it is direct to see that $\mathbb{X} \in \mathcal{F}$ and $\emptyset \in \mathcal{F}$, verifying universality and non-emptiness.

To verify Property 2 (closure under complementation), we note that $A^{c^c} = A$. Hence, if we show $A \in \mathcal{F} \implies A^c \in \mathcal{F}$ this is equivalent to showing that $A \in \mathcal{F} \iff A^c \in \mathcal{F}$. Since $\mathcal{F}$ is finitely countable, we can consider a case-wise basis for verification. Firstly, we have $A = \emptyset$. By definition, $A^c = \mathbb{X}$. We see that $\mathbb{X} \in \mathcal{F}$. As mentioned before, this implies that the case where $A = \mathbb{X}$ also holds. Now, we can proceed to verify the case where $A = \{a\}$. We note that $\{a\}^c = \{b, c\}$, and that $\{b, c\} \in \mathcal{F}$. Therefore, this case holds and thus so does the case where $A = \{b, c\}$ by biconditionality. Hence, we can conclude that $\mathcal{F}$ is closed under complementation.

To verify Property 3 (closure under countable unions), we first consider the concrete case where $\{A_i\}_{i=1}^2 = \big\{\{a\}, \{b, c\}\big\}$ where $A_1 = \{a\}$ and $A_2 = \{b, c\}$. We note that $A_1, A_2 \in \mathcal{F}$, so we would expect that $A_1 \cup A_2 \in \mathcal{F}$. Directly, $A_1 \cup A_2 = \{a\} \cup \{b, c\} = \{a, b, c\} = \mathbb{X}$, and we see that $\mathbb{X} \in \mathcal{F}$. For the remaining cases, $\forall A \in \mathcal{F}, A \cup \emptyset = A$ by fundamental set properties, and directly $A \in \mathcal{F}$ by construction. Similarly, $\forall A \in \mathcal{F}, A \cup \mathbb{X} = \mathbb{X}$ and we know that $\mathbb{X} \in \mathcal{F}$. Hence, for all cases, $\mathcal{F}$ is closed under countable unions.

From all of these properties, we can conclude that $\mathcal{F}$ is a σ-algebra, as required $\square$.

Example of a Probability Measure

Using the σ-algebra $\mathcal{F} = \big\{\emptyset, \{a\}, \{b, c\}, \{a, b, c\}\big\}$ we will define $\mu : \mathcal{F} \mapsto [0, 1]$ and prove that μ is a probability measure using the properties discussed in the literature review.

We will define a concrete example as follows, and show it is a probability measure on $\mathcal{F}$.

$$\mu(A) = \begin{cases} 2/3, & |A| = 1 \\ 1/3, & |A| = 2 \\ 1, & |A| = 3 \\ 0, & \text{otherwise} \end{cases}$$

Let's evaluate the properties discussed in the literature review to verify that μ is in fact a probability measure.

First, we evaluate Property 1 (non-negativity.) In effect, we wish to verify that $\forall A \in \mathcal{F}, \mu(A) \geq 0$. We can easily evaluate the universal in a case-wise basis.

a. $\emptyset \in \mathcal{F}$ and $\mu(\emptyset) = 0 \geq 0$.
b. $\{a\} \in \mathcal{F}$ and $\mu(\{a\}) = 2/3 \geq 0$.
c. $\{b, c\} \in \mathcal{F}$ and $\mu(\{b, c\}) = 1/3 \geq 0$.
d. $\mathbb{X} \in \mathcal{F}$ and $\mu(\mathbb{X}) = 1 \geq 0$.

Hence, $\forall A \in \mathcal{F}, \mu(A) \geq 0$, verifying the non-negativity clause. Further, we see that $\forall A \in \mathcal{F}, 0 \leq \mu(A) \leq 1$.

In addition, we can utilize the evaluations above to verify Property 2. Directly, we see that $\mu(\emptyset) = 0$ and $\mu(\mathbb{X}) = 1$, verifying Property 2 that a value of 1 is assigned to the sample space $\mathbb{X}$.

Finally, we verify Property 3 (countable additivity.) Again, because the σ-algebra is finitely countable, we verify all pairwise disjoint intersections on a case-wise basis.

a. First, we consider the set of pairwise disjoint sets $\{\emptyset, A\}$ for $A \in \mathcal{F}$. We see that $\forall A \in \mathcal{F}, \emptyset \cap A = \emptyset$ and $\emptyset \cup A = A$. Using, $\mu(\emptyset) = 0$, we observe that $\mu\Big(\bigcup_i A_i\Big) = \mu(A) = \mu(A) + \mu(\emptyset) = \sum_{i=1} \mu(A_i)$, as required.
b. Similarly, we consider $\mathbb{X} \in \mathcal{F}$, noting that $\forall A \in \mathcal{F}, \mathbb{X} \cap A = A$ is non-disjoint, so the case holds vacuously by falsity of the antecedent. A similar argument can be applied for $\{A, A\}, A \in \mathcal{F}$ which is evidently a non-disjoint pair.
c. The other pairwise disjoint set in $\mathcal{F}$ is $\{A_i\} = \big\{\{a\}, \{b, c\}\big\}$, since $\{a\} \cap \{b, c\} = \emptyset$. Hence, this pair should be countably additive. We can see directly that $\bigcup_i A_i = \{a\} \cup \{b, c\} = \mathbb{X}$, Hence, $\mu\Big(\bigcup_i A_i\Big) = \mu(\mathbb{X}) = 1$, so we anticipate $\sum_i \mu(A_i) = 1$. To verify, we see that $\sum_i \mu(A_i) = \mu(\{a\}) + \mu(\{b, c\}) = 2/3 + 1/3 = 1$, so countable additivity holds.

From all of these properties, we can conclude that μ is a probability measure on σ-algebra $\mathcal{F}$, as required $\square$.

Proof of Gamma-Poisson Conjugacy : Posterior

We noted the posterior conjugate in the methods section, and left the full derivation of the conjugate pair for the appendix here. As mentioned in the methods section, the conjugate pair is helpful for Gibbs samplers where the nonparametric Dirichlet Process is centered about a gamma base measure.

Let $\lambda \sim \text{gamma}(\alpha, \beta)$ be the prior on the Poisson rate parameter λ. Let $x_i \mid \lambda \sim \text{pois}(\lambda)$ for $i = 1, 2, \ldots, n$. We derive the expression for the conjugate prior, beginning from the well-known expression for the posterior.

$$\text{Posterior} \propto \text{Prior} \;\times\; \text{Likelihood}$$
$$\text{Posterior} \propto p_{\text{gam}}(\lambda; \alpha, \beta) \times \prod_{i=1}^{n} p_{\text{pois}}(x_i\,; \lambda)$$
$$\text{Posterior} \propto p_{\text{gam}}(\lambda; \alpha, \beta) \times \prod_{i=1}^{n} \Big(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\Big)$$
$$\text{Posterior} \propto p_{\text{gam}}(\lambda; \alpha, \beta) \times \prod_{i=1}^{n}(e^{-\lambda}) \cdot \prod_{i=1}^{n}(\lambda^{x_i}) \cdot \prod (x_i!)^{-1}$$
$$\text{Posterior} \propto \Big(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}\Big) \times (e^{-n\lambda}) \cdot (\lambda^{\sum x_i}) \cdot \prod_{i=1}^{n}(x_i!)^{-1}$$
$$\text{Posterior} \propto \underbrace{\Big(\frac{\beta^\alpha}{\Gamma(\alpha)\prod_{i=1}^{n} x_i!}\Big)}_{\text{constant wrt } \lambda} \cdot (e^{-n\lambda}e^{-\beta\lambda}) \cdot (\lambda^{\sum x_i}\lambda^{\alpha-1})$$
$$\text{Posterior} \propto (e^{-n\lambda-\beta\lambda}) \cdot (\lambda^{\sum x_i+\alpha-1})$$
$$\text{Posterior} \propto (e^{-\lambda(n+\beta)}) \cdot (\lambda^{\sum x_i+\alpha-1})$$
$$\text{Posterior} \propto \text{gam}(\alpha + \sum_{i=1}^{n} x_i, \beta + n)$$

The above gives the Posterior Distribution used in Part 3 of the DPMM Mixing Distribution definition, as required.

Proof of Gamma-Poisson Conjugacy : Posterior Predictive

As was discussed in the main work, the posterior predictive is also needed to use the sampler. Since we need the full expression (not a proportionality), we utilize line 5 from the previous proof for a single observation.

For a single observation, we have the following from the gamma distribution using x_n and 1 rather than n and the observation sum.

$$p(\lambda \mid x_n) = \Big(\frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)}\lambda^{\alpha+x_n-1}e^{-(\beta+1)\lambda}\Big)$$

Then, we compute $P(x_{n+1})$ by marginalization to arrive at the predictive distribution.

$$p(x_{n+1} \mid x_n) = \int_0^\infty p(x_{n+1} \mid \lambda)p(\lambda \mid x_n)\mathrm{d}\lambda$$
$$p(x_{n+1} \mid x_n) = \int_0^\infty p(x_{n+1} \mid \lambda)\Big(\frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)}\lambda^{\alpha+x_n-1}e^{-(\beta+1)\lambda}\Big)\mathrm{d}\lambda$$
$$p(x_{n+1} \mid x_n) = \int_0^\infty \Big(\frac{e^{-\lambda}\lambda^{x_{n+1}}}{x_{n+1}!}\Big)\Big(\frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)}\lambda^{\alpha+x_n-1}e^{-(\beta+1)\lambda}\Big)\mathrm{d}\lambda$$
$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)x_{n+1}!}\int_0^\infty e^{-\lambda}\lambda^{x_{n+1}}\Big(\lambda^{\alpha+x_n-1}e^{-(\beta+1)\lambda}\Big)\mathrm{d}\lambda$$
$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)x_{n+1}!}\int_0^\infty \Big(\lambda^{\alpha+x_n+x_{n+1}-1}e^{-(\beta+2)\lambda}\Big)\mathrm{d}\lambda$$

At this point, we recall the definition of the complete gamma function.

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\mathrm{d}t, \text{ where } \Re(z) > 0$$

Note by the strictly real-valued supports and parameters of both the Gamma and Poisson distributions that the $\Re(z) > 0$ clause holds trivially in this case.

For our expression, we will proceed with substitution to get it into this form.

First, we let $t = (\beta + 2)\lambda$. To solve via substitution, we note that:

$$\mathrm{d}t = (\beta + 2)\mathrm{d}\lambda, \text{ hence } \mathrm{d}\lambda = \frac{\mathrm{d}t}{(\beta + 2)}$$

Let us substitute this value into our expression and simplify.

$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)x_{n+1}!}\int_0^\infty \left(\frac{t}{(\beta+2)}\right)^{\alpha+x_n+x_{n+1}-1}(e^{-t})\frac{\mathrm{d}t}{(\beta+2)}$$

$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)x_{n+1}!(\beta+2)^{x_n+x_{n+1}-1}(\beta+2)}\int_0^\infty t^{\alpha+x_n+x_{n+1}-1}(e^{-t})\mathrm{d}t$$

$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}}{\Gamma(\alpha+x_n)x_{n+1}!(\beta+2)^{\alpha+x_n+x_{n+1}-1}(\beta+2)^1}\Gamma(\alpha+x_n+x_{n+1})$$

$$p(x_{n+1} \mid x_n) = \frac{(\beta+1)^{\alpha+x_n}\Gamma(\alpha+x_n+x_{n+1})}{\Gamma(\alpha+x_n)x_{n+1}!(\beta+2)^{\alpha+x_n+x_{n+1}}} \quad \square$$

The above gives the Posterior Predictive used in Part 4 of the DPMM Mixing Distribution definition, as required.

Section 2: R Code

Results Code

```
library(dirichletprocess)
# set seed for reproducibility
set.seed(447)
# start the clock
start_time = proc.time()
M = 10000
RUN = FALSE # TRUE if running sampler

###########################
### Mixing Distribution ###
###########################

# define the framework conjugate mixture model
poisMd = MixingDistribution(
  distribution = "poisson",
  priorParameters = c(1, 2),
  conjugate = "conjugate"
)
# Part 1: Poisson Likelihood
Likelihood.poisson = function(mdobj, x, theta){
```

```
  return(as.numeric(dpois(x, theta[[1]])))
}
# Part 2: Gamma Prior : Base Measure
PriorDraw.poisson = function(mdobj, n){
  draws = rgamma(n, mdobj$priorParameters[1], mdobj$priorParameters[2])
  theta = list(array(draws, dim=c(1,1,n)))
  return(theta)
}
# Part 3: Posterior Draw (defined by conjugacy)
PosteriorDraw.poisson = function(mdobj, x, n=1){
  priorParameters = mdobj$priorParameters
  theta = rgamma(n, priorParameters[1] + sum(x),
                    priorParameters[2] + nrow(x))
  return(list(array(theta, dim=c(1,1,n))))
}

# Part 4: Predictive Distribution by Marginalization
Predictive.poisson = function(mdobj, x){
  priorParameters = mdobj$priorParameters
  pred = numeric(length(x))
  for(i in seq_along(x)){
    alphaPost = priorParameters[1] + x[i]
    betaPost =  priorParameters[2] + 1
    pred[i] = (priorParameters[2] ^ priorParameters[1]) / gamma(priorParameters[1])
    pred[i] = pred[i] * gamma(alphaPost) / (betaPost^alphaPost)
    pred[i] = pred[i] * (1 / prod(factorial(x[i])))
  }
  return(pred)
}

###########################
### D.P. Gibbs Sampling ###
###########################

# read in cleaned data frame
df = read.csv("final_project/cleaned_crash_data.csv")
# monthly crash count, in 100s of crashes
y = ( round((df$crash_count)/100) )

# create DP Poisson Mixture Model from mix dist. defined earlier
dirp = DirichletProcessCreate(y, poisMd)

if(RUN){
  # initialize and fit DPMM via Gibbs
  dirp = Initialise(dirp)
  dirp = Fit(dirp, M)
  dirp = Burn(dirp, 100)
  # compute, posterior frame: sampling from the posterior
  cat("Generating Posterior Frame...")
  # include 95% and 99% Credible Intervals
  postf = PosteriorFrame(dirp, 0:22, 10000, ci_size = c(0.1, 0.01))
  # save to avoid repeat simulation
  saveRDS(postf, file = "final_project/posterior_sampleframe.RDS")
```

```r
  saveRDS(dirp, file =  "final_project/posterior_results.RDS")
}
# report runtime
total_time = proc.time() - start_time
cat("Total Runtime of Script: ", total_time['elapsed'], "seconds\n")
```

Dirichlet Process Finite Approximation

```r
library(ggplot2)
library(latex2exp)
library(RColorBrewer)
library(scales)
# seed for reproducibility
set.seed(1924)
# number of clusters
K = 20
# base measure/distribution
G_0 = function(n) {
  rgamma(n, 1, 2)
}
alpha = rgamma(1, 1, 1)

##############################
##### Finite D.P. Appx. #####
##############################


# generate stick breaking finite approximation
b <- rbeta(K, 1, alpha)
# empty vector for pulls
p <- numeric(length = K)
# initial stick break
p[1] <- b[1]
# further breaks following GEM(a) definition from methods
p[2:K] <- sapply(2:K, function(i)
  b[i] * prod(1 - b[1:(i - 1)]))
# then, sample from base distribution by weight probabilities
# this creates the finite approximation as discussed in the methods
theta <- sample(G_0(K), prob = p, replace = TRUE)

##############################
##### FINITE D.P. PLOT ######
##############################


plotDF = data.frame(DirB = theta,  DirP = log(p))
# plot heatmap of results
p1 = ggplot(plotDF, aes(x = DirB, y = DirP)) +
  geom_density_2d_filled() +
  labs(
    title =
```

```
      TeX(
        "Finite Approximation of Dirichlet Process : DP($\\alpha G_0$) Realization"
      ),
    subtitle = TeX(
      "Where $K = 20$, $G_0 \\sim$ gamma(1, 2) and $\\alpha \\sim$ gamma(1,1)"
    ),
    y = TeX("Log of Mixture Weights: $\\{\\pi_k\\}_{k = 1}^K$"),
    x = TeX("Cluster Parameters: $\\{\\lambda_k\\}_{k = 1}^K$")
  )  + theme_bw() + scale_fill_viridis_d(option = "magma")  + theme(
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    plot.background = element_rect(fill = "white", colour = "white"),
    plot.title = element_text(margin = margin(b = -3.5, unit = "pt")),
    plot.subtitle = element_text(margin = margin(b = -5, unit = "pt")),
    legend.position = "none",
    legend.title = element_blank(),
    axis.ticks.length = unit(-2, "mm"),
    legend.text = element_text(size = 8),
    legend.margin = margin(t = 0, unit = "mm", l = -5)
  ) + scale_y_continuous(n.breaks = 10) +
  scale_x_continuous(n.breaks = 10)

print(p1) # trbl
ggsave(
  "final_project/dirch_appx.png",
  plot = p1,
  width = 7,
  height = 5
)
```

Data Processing Code

```
library(dplyr)
library(lubridate)
library(readxl)
library(tidyr)

#######################
### Data Processing ###
#######################

# NOTE: the aggregation is over a very large dataset, so runtime is slow

cat("Reading Data... \n")
# read Chicago Crash data from excel using `readxl`
df <- read_excel("final_project/crash_dates_and_damages.xlsx")

cat("Processing Dates...\n")
df <- df %>%
  # drop weirdly formatted excel dates
  filter(!grepl("^\\d+\\.\\d+$", CRASH_DATE)) %>%
```

```r
  # standrdize remaining dates to same timezone
  mutate(CRASH_DATE = mdy_hms(CRASH_DATE, tz = "UTC", quiet = TRUE)) %>%
  # remove NAs
  drop_na(CRASH_DATE) %>%
  # standardize formatting to MDY
  mutate(CRASH_DATE = format(CRASH_DATE, "%m/%d/%Y"))

cat("Aggregating... \n")
# here, we aggregate severity counts weekly
aggregated_data <- lapply(damage_levels, function(damage_level) {
  # we filter through each damage level
  df_filtered <- df %>% filter(DAMAGE == damage_level)
  # and aggregate by week
  weekly_aggregation <- df_filtered %>%
    mutate(Week = floor_date(as.Date(CRASH_DATE, format = "%m/%d/%Y"), "week")) %>%
    group_by(Week) %>%
    summarise(Crash_Count = n(), .groups = 'drop') %>%
    arrange(Week)
  # use NNI for poorly-captured 2014, 2015 data
  cat("Imputing ", damage_level, "...\n")
  # get 2016 weeks
  weeks_2016 <- weekly_aggregation %>%
    filter(year(Week) == 2016) %>%
    pull(Week)
  # get exact week if present, else nearest week by euclidean distance
  weekly_aggregation <- weekly_aggregation %>%
    rowwise() %>%
    mutate(
      Nearest_2016_Week = if(year(Week) %in% c(2014, 2015)) {
        weeks_2016[which.min(abs(difftime(Week, weeks_2016, units = "weeks")))]
      } else {
        Week
      }
    ) %>%
    left_join(weekly_aggregation %>% filter(year(Week) == 2016) %>%
                select(Week, Crash_Count), by = c("Nearest_2016_Week" = "Week")) %>%
    mutate(
      # Use 2016's Crash_Count for nearest week and add to scaled 2014 count if present
      Crash_Count_Adjusted = if_else(year(Week) == 2014, Crash_Count.y + Crash_Count.x / max(Crash_Coun
    ) %>%
    select(Week, Crash_Count_Adjusted)

  return(weekly_aggregation)
})

# the data frame we use is then the aggregated weeks & counts
outDF = data.frame(
  crash_time = c(
    aggregated_data[[1]]$Week,
    aggregated_data[[2]]$Week,
    aggregated_data[[3]]$Week
  ),
  crash_count =  c(
```

```
    aggregated_data[[1]]$Crash_Count_Adjusted,
    aggregated_data[[2]]$Crash_Count_Adjusted,
    aggregated_data[[3]]$Crash_Count_Adjusted
  )
)

# which we write to a csv
cat("Writing to File... \n")
write.csv(outDF, "final_project/cleaned_crash_data.csv", row.names = FALSE)
```

$$
\begin{aligned}
\{\pi_k\}_{k=1}^{K} &\sim \text{GEM}(\alpha), \text{ stick-breaking process} \\
\{\theta_k\}_{k=1}^{K} &\sim \mathbb{G}_0, \text{ base measure} \\
z_i &\sim \text{categorical}(\{1,2,\dots,K\},\{\pi_k\}), \text{ cluster assignment } i \in [1,N] \\
y_i \mid z_i, \{\theta_k\}_1^K &\sim \text{F}(x_i\,;\,\theta_{z_i}), \text{ likelihood given cluster}
\end{aligned}
$$

Infinite Dirichlet Process Mixture Model

$$
\begin{aligned}
\{\pi_k\}_{k=1}^{\infty} &\sim \text{GEM}(\alpha) \\
\{\langle \alpha_k, \beta_k \rangle\}_{k=1}^{\infty} &\sim \mathbb{G}_0 \\
y_i \mid \{\pi_k\}, \{\langle \alpha_k, \beta_k \rangle\} &\sim \sum_{k=1}^{\infty} \pi_k \text{beta}(y_i \mid \langle \alpha_k, \beta_k \rangle)
\end{aligned}
$$

Where the MCMC implementation is facilitated by the `dirichletprocess` package, created by (Markwick 2023).

Finite-approximated infinite mixture DPMM, similar

$$
\begin{aligned}
\{\pi_k\}_{k=1}^{K} &\sim \text{GEM}(\alpha), \text{ stick-breaking process} \\
\{\theta_k\}_{k=1}^{K} &\sim \mathbb{G}_0, \text{ base measure} \\
z_i &\sim \text{categorical}(\{1,2,\dots,K\},\{\pi_k\}), \text{ cluster assignment } i \in [1,N] \\
y_i \mid z_i, \{\theta_k\}_1^K &\sim \text{F}(x_i\,;\,\theta_{z_i}), \text{ likelihood given cluster}
\end{aligned}
$$

$$
\begin{aligned}
\{\pi_k\}_{k=1}^{\infty} &\sim \text{GEM}(\alpha), \text{ stick-breaking process} \\
\{\theta_k\}_{k=1}^{\infty} &\sim \mathbb{G}_0, \text{ base measure}, \\
F &= \prod_{i=1}^{N} \left(\sum_{k=1}^{\infty} \pi_k N(x_i \mid \theta_k) \right), \text{ likelihood from normal kernel}
\end{aligned}
$$

Sources

(Jones 2021)

Billingsley, Patrick. 2012. *Probability and Measure, Anniversary Edition.* Wiley.

CPD. 2024. "Chicago Traffic Crashes - Chicago Police Dept." Kaggle. https://doi.org/10.34740/KAGGLE/DSV/7339559.

Demirbas, Seckin, and Andrew Rechnitzer. 2023. "An Introduction to Mathematical Proof : MATH 220." Free web and pdf textbook. https://personal.math.ubc.ca/~PLP/.

Ferguson, Thomas. 1973. "Bayesian Analysis of Some Nonparametric Problems." *Annals of Statistics* 1 (2): 209–30. https://doi.org/10.1214/aos/1176342360.

Hannah, Lauren A. 2011. "Dirichlet Process Mixtures of Generalized Linear Models." *Journal of Machine Learning Research* 12: 1923–53. https://www.jmlr.org/papers/volume12/hannah11a/hannah11a.pdf.

Ishwaran, Hemant, and Lancelot F. James. 2001. "Gibbs Sampling Methods for Stick-Breaking Priors." *Journal of the American Statistical Association* 96 (453): 161–73. https://doi.org/10.1198/016214501750332758.

Jones, Andy. 2021. "Dirichlet Process Mixture Models." Online. https://andrewcharlesjones.github.io/journal/dpmm.html.

Markwick, Dean. 2023. *Dirichletprocess: An 'r' Package for Dirichlet Process Mixture Models*. https://dm13450.github.io/dirichletprocess/.

Neal, Radford M. 2000. "Markov Chain Sampling Methods for Dirichlet Process Mixture Models." *Journal of Computational and Graphical Statistics* 9 (2): 249–65. https://doi.org/10.1080/10618600.2000.10474879.

Rudin, Walter. 1986. *Real and Complex Analysis*. 3rd ed. McGraw-Hill.

Sethuraman, Jayaram. 1994. "A Constructive Definition of Dirichlet Priors." *Statistica Sinica*.

StackOverflow. 2023. "How to Use Custom Functions in Mutate (Dplyr)." Stack Overflow. https://stackoverflow.com/questions/44730774/how-to-use-custom-functions-in-mutate-dplyr.

Xing, Eric P. 2014. "Hierarchical Dirichlet Processes." Carnegie Mellon University; Online. https://www.cs.cmu.edu/~epxing/Class/10708-14/scribe_notes/scribe_note_lecture20.pdf.