# Supplementary Materials for Introducing Variational Inference in Statistics and Data Science Curriculum

**Abstract**

The supplementary materials include: 1) Details of the class activity on probabilistic model for count data with variational inference, introduced in Section 3 in the main text; 2) The manual of the `R` `shiny` app we have developed for the module, mentioned in Section 3 in the main text; 3) Detais of the guided `R` logistic regression lab with U.S. women labor participation sample data, presented in Section 4.1 in the main text; and 4) Details of the guided `R` lab of the LDA application to a sample of the Associated Press newspaper articles with variational inference, presented in Section 4.2 in the main text.

*Keywords:* Active learning, Bayesian statistics, probabilistic models, statistical computing, variational inference

# 1 Class Activity: Probabilistic Model for Count Data with Variational Inference

The goal of this activity is to illustrate variational inference on a simple example of Gamma-Poisson conjugate model, which is a popular model for count data.

## 1.1 A Motivating Example

Our task is to estimate the average number of active users of a popular massively multiplier online role-playing game (mmorpg) playing between the peak evening hours 7 pm and 10 pm. This information can help the game developers in allocating server resources and optimizing user experience. To make this estimate, we will consider the following counts (in thousands) of active players collected during the peak evening hours over a two-week period past month.

| | **Sun** | **Mon** | **Tue** | **Wed** | **Thu** | **Fri** | **Sat** |
|---|---|---|---|---|---|---|---|
| **Week 1** | 50 | 47 | 46 | 52 | 49 | 55 | 53 |
| **Week 2** | 48 | 45 | 51 | 50 | 53 | 46 | 47 |

## 1.2 Overview of the Gamma-Poisson Model

**Sampling density:**

Suppose that $\boldsymbol{y} = (y_1, \ldots, y_n)$ represent the observed counts in $n$ time intervals where the counts are independent, then each $y_i$ follows a Poisson distribution with rate $\theta > 0$. Namely,

$$y_i \mid \theta \sim \text{Poisson}(\theta)$$

- $\mathbb{E}(y_i \mid \theta) = \theta$
- $\mathbb{V}ar(y_i \mid \theta) = \theta$

**Prior distribution:**

$$\theta \sim \text{Gamma}(\alpha, \beta)$$

- $\alpha > 0$ is the shape parameter
- $\beta > 0$ is the rate parameter
- $\mathbb{E}(\theta) = \frac{\alpha}{\beta}$
- $\mathbb{V}ar(\theta) = \frac{\alpha}{\beta^2}$

**Posterior distribution:**

$$\theta \mid y_1, \ldots, y_n \sim \text{Gamma}(\alpha + \sum_{i=1}^{n} y_i, \beta + n)$$

## 1.3 Exact Inference with the Gamma-Poisson Model

We will start by selecting a prior distribution for the unknown average number of active users. Suppose that we elicit an expert's advice on the matter, and they tell us that a similar mmorpg has typically about 50,000 users during peak hours. However, they are not too sure about that, so the interval between 45,000 and 55,000 users should have a reasonably high probability. This reasoning leads to a $\text{Gamma}(100, 2)$ as a reasonable prior for the average number of active users.

**Task 1: Explain the reasoning behind using Gamma$(100, 2)$ as the prior distribution.**

**Task 2: Use the information above to find the exact posterior distribution for the average number of active users.**

**Task 3: What are the mean and standard deviation of the posterior distribution that you just obtained? What is your recommendation about the typical number of active users playing the mmorpg between the peak evening hours 7pm and 10pm?**

### 1.4 Variational Inference with the Gamma-Poisson Model

Variational inference approximates the (unknown) posterior distribution of a parameter by a simple family of distributions. In this case, we will try to approximate the posterior distribution of the mmorpg's average number of active users between the peak hours $\theta$ by a log-normal distribution with mean $\mu$ and standard deviation $\sigma$. Log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed. It also happens to be a popular variational family for non-negative parameters as it is amenable to autodifferentiation. Since we know exactly how the posterior distribution for Gamma-Poisson model looks like, we will be able to check the fidelity of the variational approximation. Use the accompanying applet titled *Variational Inference with Gamma-Poisson Model for count data* to complete the following task.

**Task 4: Use the sliders in the applet to manually find the member of a log-normal variational family that well approximates the posterior distribution of $\theta$. What is your strategy?**

**Task 5: Compare your approximation with a neighbor. Whose approximation is closer to the exact posterior distribution of $\theta$? How are you deciding?**

**Task 6: Check the *Fit a variational approximation* box in the applet to find the variational approximation using the gradient ascent algorithm. How close was the variational approximation that you found manually to the one found here?**

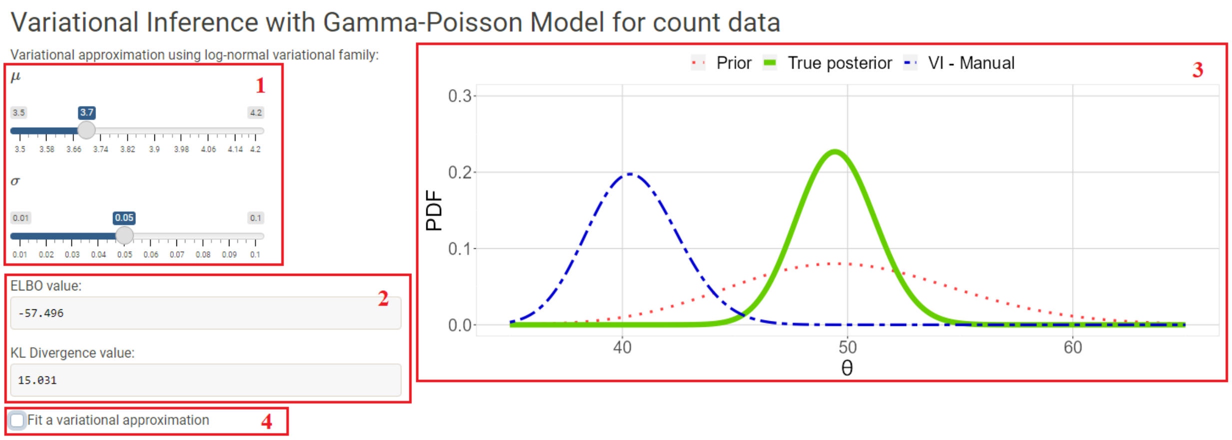


Figure 1: The applet is based on the class activity presented in Section 1 of the supplementary materials. The applet visual before checking the “Fit a variational approximation“ checkbox is displayed.

# 2 Manual of the `R shiny` app

This document describes the elements of `R Shiny` applet that accompanies the “Probabilistic Model for Count Data with Variational Inference” class activity. Note that the numbering in Section 2.1 and Section 2.2 corresponds to the numbered boxes in Figure 1 and Figure 2.

## 2.1 Manual Search for Variational Approximation

1. Sliders to control the mean $\mu$ and the standard deviation $\sigma$ of log-normal variational family.

2. The ELBO and KL divergence values for variational approximation based on the mean and standard deviations selected in box 1.

3. A plot that displays the true $\text{Gamma}(792, 100)$ posterior distribution, the $\text{Gamma}(100, 2)$ prior distribution, and the variational approximation based on the selection in box 1.

4. A checkbox to display the results of ELBO maximization via gradient ascent algorithm. The resulting variational approximation is plotted in box 3.

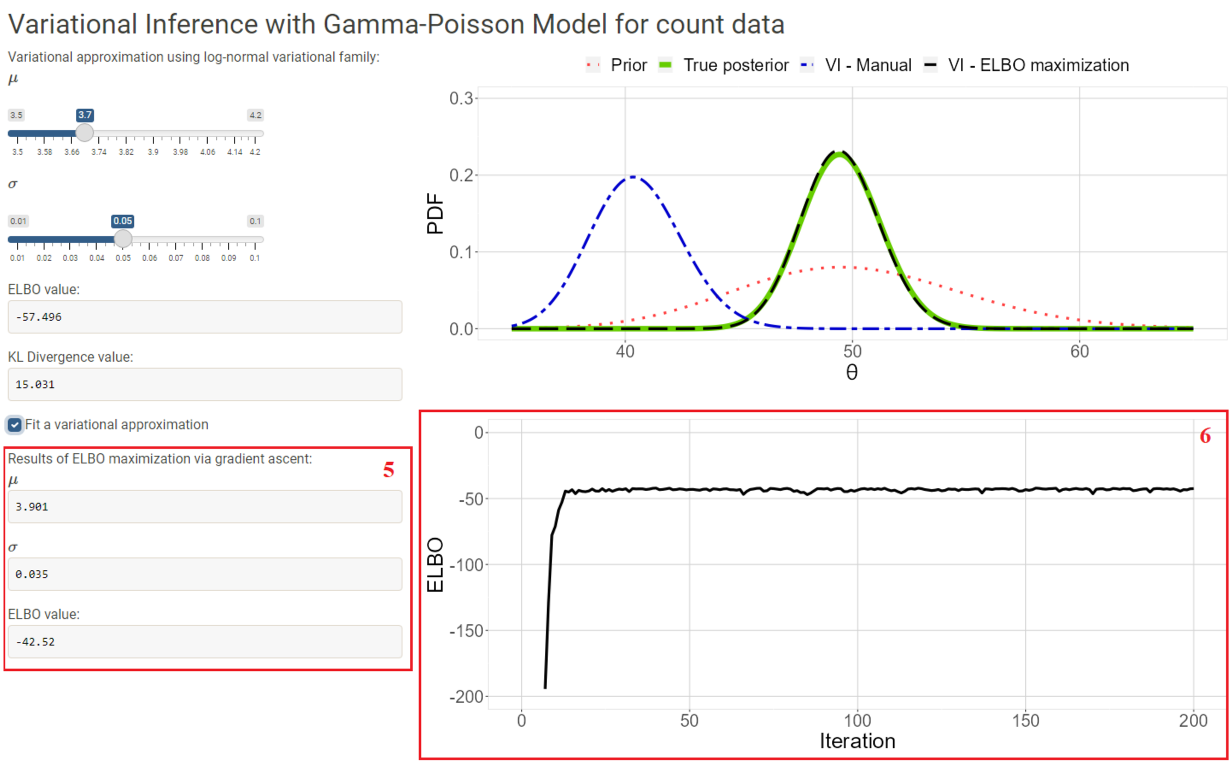


Figure 2: The applet visual after checking the "Fit a variational approximation" checkbox is displayed.

## 2.2 Variational Approximation Based on ELBO Maximization

5. The resulting mean $\mu$, standard deviation $\sigma$, and ELBO values of variational approximation based on ELBO maximization.

6. A plot depicting ELBO values for each iteration of the gradient ascent algorithm.

# 3 Lab on Logistic Regression

The goal of this lab is to gain practical experience with variational inference on a case study of U.S. women labor participation with logistic regression model and implement the model in `R`. To do so, we consider a sample data from the University of Michigan Panel Study of Income Dynamics (PSID) which is the longest running longitudinal household survey in the world. The survey dates back to 1968 and contains information on over 18,000 individuals living in 5,000 families in the United States. The survey of these individuals and their descendants has been collected continuously and includes data on income, wealth, employment status, health, marriage, and hundreds of other variables. Our interest is in analyzing a PSID sample of 753 observations from 1976 (Mroz (1987)). The PSID 1976 survey is particularly interesting since it interviewed wives in households directly in the previous year. You can load the dataset `PSID` with the following R command.

```
PSID <- read.csv("https://raw.githubusercontent.com/monika76five/ProbBayes/master/R%20C
```

This PISD sample contains two variables: Family income exclusive of wife's income (in \$1000) and wife's labor participation (yes or no). The goal of the lab is predicting a wife's labor participation status (response variable) from the family income exclusive of her income (predictor variable) using logistic regression. We refer interested readers to Albert & Hu (2019) Section 11.4 for an in-depth analysis of Bayesian logistic regression applied to the same prediction task.

## 3.1 Overview of the Logistic Regression Model and Stan Script

Logistic regression model is a supervised learning algorithm for binary classification. It is therefore well suitable for analysis of a binary response such as labor participation. Namely, the logistic regression model assumes that a binary response $y_i$ follows a Bernoulli distribution with probability of success $p_i$:

$$y_i \mid p_i \sim \text{Bernoulli}(p_i).$$

To relate a predictor $x_i$ to the response $y_i$, logistic regression typically considers the natural logarithm of odds $p_i/(1-p_i)$ (also known as logit) to be a linear function of the predictor variables $x_i$:

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta x_i, \tag{1}$$

with $\alpha$ and $\beta$ being regression coefficients. Note that it is a bit more challenging to interpret the coefficients in the logistic regression than in standard linear regression as $\alpha$ and $\beta$ are directly related to the log odds $p_i/(1-p_i)$, instead of $p_i$. For example, $e^\alpha$ is the odds when the value of predictor $x_i$ is , whereas the quantity $e^\beta$ refers to the change in odds per unit increase in $x_i$.

Lastly, by rearranging the terms in Equation (1), one can express the probability of success $p_i$ as

$$p_i = \frac{e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}}.$$

In the Bayesian framework, one proceeds to prior specification of regression coefficients $(\alpha, \beta)$ and posterior inference through MCMC. For illustration, we consider independent normal priors for the regression coefficients $\alpha \sim \text{Normal}(\mu_0, \sigma_0)$ and $\beta \sim \text{Normal}(\mu_1, \sigma_1)$, where $(\mu_0, \mu_1)$ and $(\sigma_0, \sigma_1)$ are the prior means and standard deviations for the regression coefficients respectively.

Below, we include the `Stan` script for the logistic regression model of PSID data.

```
data {
  int<lower=0> N;
  vector[N] x;
  int<lower=0,upper=1> y[N];
}
parameters {
  real alpha; // intercept
  real beta; // slope
}
model {
  y ~ bernoulli_logit(alpha + beta * x);
  // priors
  alpha ~ normal(0, 5);
  beta ~ normal(0, 5);
}
```

## 3.2 Variational Inference with the Logictic Regression Model

We are now ready to fit the logistic regression model using variational inference capabilities of the `cmdstanr` package. The following code achieves the goal:

```
Logistic_model_cmd <- cmdstan_model(stan_file = "Logistic.stan")

data = list(N = dim(PSID)[1],
            x = PSID$FamilyIncome,
            y = PSID$Participation
)

vi_fit <- Logistic_model_cmd$variational(data = data,
                                          seed = 1,
                                          output_samples = 5000,
                                          eval_elbo = 1,
                                          grad_samples = 15,
                                          elbo_samples = 15,
                                          algorithm = "meanfield",
                                          output_dir = NULL,
                                          iter = 1000,
                                          adapt_iter = 20,
                                          save_latent_dynamics=TRUE,
                                          tol_rel_obj = 10^-4)
```

The "Logistic.stan" file contains the `Stan` script for the logistic regression model provided in Section 3.1. We recommend the usage of the `R` help to get familiar with the `variational()` method of the `cmdstan_model()` function. The variable `vi_fit` contains the results of variational approximation of the logistic regression parameters. For example, one can obtain the distribution of $\beta$ with `vi_fit$summary("beta")`.

Finally, to access the ELBO values, use the following:

```
vi_diag <- utils::read.csv(vi_fit$latent_dynamics_files()[1],
                           comment.char = "#")
ELBO <- data.frame(Iteration = vi_diag[,1], ELBO = vi_diag[,3])
```

**Task 1: Use a graphical display to show the posterior interval estimates for the probability of labor participation of a married woman who has a family income exclusive of her own income ranging from $10,000 to $70,000 with $10,000 increments.**

**Task 2: Obtain an MCMC fit the logistic regression model and compare the MCMC approximation with the variational approximation. What do you observe?**

**Task 3: Create a dataset that contains 50 replicates of the original PSID dataset (37,650 observations in total). Obtain both MCMC and variational fit for the regression model and compare the times that it takes to obtain each of the approximations.**

All necessary R code for fitting the logistic regression model to the PSID sample, including the graphical displays shown in the main text, is included in a separate R script file called Logistic_LAB.R available as a part of the supplementary materials. We also include a printout of the R script below for interested readers.

```
library(cmdstanr)
# Checking integrity of installation of cmdstanr
check_cmdstan_toolchain()
install_cmdstan(cores = 2)
cmdstan_path()
cmdstan_version()

# Auxiliary packages
library(tidyverse)
```

```r
## Get data
PSID <- read.csv("https://raw.githubusercontent.com/monika76five/ProbBayes/master/R%20C

## Input for stan model
data = list(N = dim(PSID)[1],
            x = PSID$FamilyIncome,
            y = PSID$Participation
)

## VI fit
Logistic_model_cmd <- cmdstan_model(stan_file = "Logistic.stan")
Logistic_model_cmd$print()

vi_fit <- Logistic_model_cmd$variational(data = data,
                                         seed = 1,
                                         output_samples = 5000,
                                         eval_elbo = 1,
                                         grad_samples = 15,
                                         elbo_samples = 15,
                                         algorithm = "meanfield",
                                         output_dir = NULL,
                                         iter = 1000,
                                         adapt_iter = 20,
                                         save_latent_dynamics=TRUE,
                                         tol_rel_obj = 10^-4)

## Plotting ELBO
vb_diag <- utils::read.csv(vi_fit$latent_dynamics_files()[1],
                           comment.char = "#")
```

```r
ELBO = data.frame(Iteration = vb_diag[,1], ELBO = vb_diag[,3])

ggplot(data = ELBO, aes(x = Iteration, y = ELBO)) + geom_line(lwd=1.5) +
  theme(text = element_text(size = 20),
        panel.background = element_rect(fill = "transparent",
                                        color = "lightgrey"),
        panel.grid.major = element_line(colour = "lightgrey")) +
  xlim(0,110)

## Accessing parameters
vi_fit$summary("alpha")
vi_fit$summary("beta")

## Posterior interval estimates for the probability of labor participation
prob_interval <- function(x, post){
  lp <- post[, 1] + x * post[, 2]
  quantile(exp(lp) / (1 + exp(lp)),
           c(.05, .50, .95))
}

out <- sapply(seq(10, 70, by = 10),
              prob_interval, vi_fit$draws()[,3:4])

df_out <- data.frame(Income = seq(10, 70, by = 10),
                     Low = out[1, ],
                     M = out[2, ],
                     Hi = out[3,  ])

ggplot(df_out) +
  geom_line(aes(x = Income, y = M), lwd=1.5) +
```

```r
  geom_segment(aes(x = Income, y = Low,
                   xend = Income, yend = Hi), size = 2) +
  ylab("Prob(Participate)") +
  ylim(0, 1) +
  theme(text = element_text(size = 20),
        panel.background = element_rect(fill = "transparent",
                                        color = "lightgrey"),
        panel.grid.major = element_line(colour = "lightgrey"))

## MCMC fit
mcmc_fit <- Logistic_model_cmd$sample(
  data = data,
  seed = 1,
  chains = 1,
  iter_warmup = 5000,
  iter_sampling = 5000
)

mcmc_fit$summary()

## VI/MCMC comparison
df <- data.frame(Method = c(rep('VI', 5000), rep('MCMC', 5000) ),
                 value = c(vi_fit$draws("alpha"),
                           mcmc_fit$draws("alpha")))

ggplot(df, aes(x=value, fill=Method)) +
  geom_histogram( color='#e9ecef', alpha=0.6, position='identity') +
  xlab("alpha")

df <- data.frame(Method = c(rep('VI', 5000), rep('MCMC', 5000) ),
```

```
                value = c(vi_fit$draws("beta"),
                          mcmc_fit$draws("beta")))

ggplot(df, aes(x=value, fill=Method)) +
  geom_histogram( color='#e9ecef', alpha=0.6, position='identity') +
  xlab("beta")

## Speed comparison
# VI
PSID_rep <- PSID[rep(seq_len(nrow(PSID)), 50), ]

data = list(N = dim(PSID_rep)[1],
            x = PSID_rep$FamilyIncome,
            y = PSID_rep$Participation
)

vi_fit <- Logistic_model_cmd$variational(data = data,
                                         seed = 1,
                                         output_samples = 5000,
                                         eval_elbo = 1,
                                         grad_samples = 15,
                                         elbo_samples = 15,
                                         algorithm = "meanfield",
                                         output_dir = NULL,
                                         iter = 1000,
                                         adapt_iter = 20,
                                         save_latent_dynamics=TRUE,
                                         tol_rel_obj = 10^-4)


# MCMC
```

```r
mcmc_fit <- Logistic_model_cmd$sample(
  data = data,
  seed = 1,
  chains = 1,
  iter_warmup = 5000,
  iter_sampling = 5000
)
```

# 4 Lab on Document Clustering

The goal of this lab is to apply variational inference in a more advanced case study based on Latent Dirichlet Allocation (LDA) and implement the model in `R` applied to a dataset of documents. To do so, we consider a collection of 2246 Associated Press newspaper articles to be clustered using the LDA model. The dataset is part of the `topicmodels R` package. You can load the dataset `AssociatedPress` with the following R command.

```r
data("AssociatedPress", package = "topicmodels")
```

## 4.1 Overview of the LDA Model and Stan Script

The LDA is a mixed-membership clustering model, commonly used for document clustering. LDA models each document to have a mixture of topics, where each word in the document is drawn from a topic based on the mixing proportions. Specifically, the LDA model assumes $K$ topics for $M$ documents made up of words drawn from $V$ distinct words. For document $m$, a topic distribution $\boldsymbol{\theta_m}$ is drawn over $K$ topics from a Dirichlet distribution,

$$\boldsymbol{\theta}_m \sim \text{Dirichlet}(\boldsymbol{\alpha}), \tag{2}$$

where $\sum_{k=1}^{K} \theta_{m,k} = 1$ $(0 \le \theta_{m,k} \le 1)$ and $\boldsymbol{\alpha}$ is the prior a vector of length $K$ with positive values.

Each of the $N_m$ words $\{w_{m,1}, \ldots, w_{m,N_m}\}$ in document $m$ is then generated independently conditional on $\boldsymbol{\theta}_m$. To do so, first, the topic $z_{m,n}$ for word $w_{m,n}$ in document $m$ is

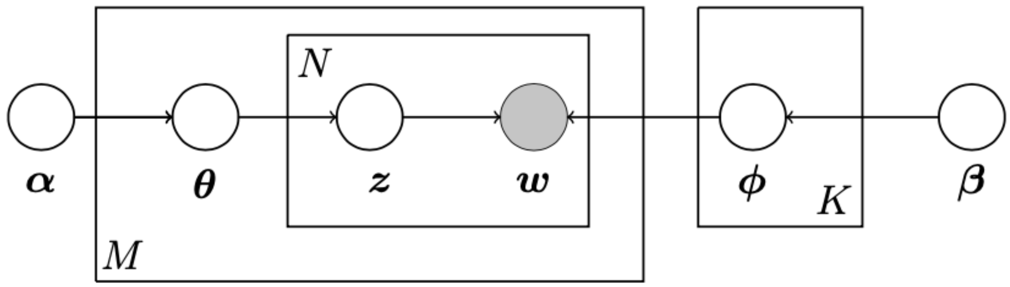


Figure 3: Graphical model representation of LDA. The outer box represents the documents. On the left, the inner box represents the topics and words within each document. On the right, the box represents the topics.

drawn from

$$z_{m,n} \sim \text{categorical}(\boldsymbol{\theta}_m), \tag{3}$$

where $\boldsymbol{\theta}_m$ is the document-specific topic-distribution defined in Equation (2).

Next, the word $w_{m,n}$ in document $m$ is drawn from

$$w_{m,n} \sim \text{categorical}(\boldsymbol{\phi}_{z[m,n]}), \tag{4}$$

which is the word distribution for topic $z_{m,n}$. Note that z[m, n] in Equation (4) refers to $z_{m,n}$.

Lastly, a Dirichlet prior is given to distributions $\boldsymbol{\phi}_k$ over words for topic $k$ as

$$\boldsymbol{\phi}_k \sim \text{Dirichlet}(\boldsymbol{\beta}), \tag{5}$$

where $\boldsymbol{\beta}$ is the prior a vector of length $V$ (i.e., the total number of words) with positive values. Figure 3 shows a graphical model representation of LDA.

Below, we include the `Stan` script for the LDA model provided by Stan Development Team (2022). Note that `Stan` supports the calculation of marginal distributions over the continuous parameters by summing out the discrete parameters in mixture models (Stan Development Team 2022). This process corresponds to the `gamma` parameter in the `Stan` script below. We refer interested readers to Stan Development Team (2022) for further details.

```
data {
  int<lower=2> K;               // number of topics
  int<lower=2> V;               // number of words
  int<lower=1> M;               // number of docs
  int<lower=1> N;               // total word instances
  int<lower=1, upper=V> w[N];    // word n
  int<lower=1, upper=M> doc[N];  // doc ID for word n
  vector<lower=0>[K] alpha;     // topic prior vector of length K
  vector<lower=0>[V] beta;      // word prior vector of length V
}

parameters {
  simplex[K] theta[M];   // topic distribution for doc m
  simplex[V] phi[K];     // word distribution for topic k
}

model {
  for (m in 1:M)
    theta[m] ~ dirichlet(alpha);
  for (k in 1:K)
    phi[k] ~ dirichlet(beta);
  for (n in 1:N) {
    real gamma[K];
    for (k in 1:K)
      gamma[k] = log(theta[doc[n], k]) + log(phi[k, w[n]]);
    target += log_sum_exp(gamma);  // likelihood;
  }
}
```

## 4.2 Variational Inference with the LDA model

For demonstration purposes, we shall start with a two-topic LDA model (i.e., $K = 2$). Before that, we recommend removing the words from `AssociatedPress` datasets that are rare using the function `removeSparseTerms()` from the `tm` package. These words have a minimal effect on the LDA parameter estimation. Nevertheless, they increase the computational cost of variational inference and therefore should be removed using the following R command.

```
dtm <- removeSparseTerms(AssociatedPress, 0.95)
```

We are now ready to fit the LDA model using variational inference capabilities of the `cmdstanr` package. The following code achieves the goal:

```
LDA_model_cmd <- cmdstan_model(stan_file = "LDA.stan")

N_TOPICS <- 2
data <- list(K = N_TOPICS,
             V = dim(dtm)[2],
             M = dim(dtm)[1],
             N = sum(dtm$v),
             w = rep(dtm$j,dtm$v),
             doc = rep(dtm$i,dtm$v),
             #according to Griffiths and Steyvers(2004)
             alpha = rep(50/N_TOPICS,N_TOPICS),
             beta = rep(1,dim(dtm)[2])
)

vi_fit <- LDA_model_cmd$variational(data = data,
                                    seed = 1,
                                    output_samples = 1000,
                                    eval_elbo = 1,
```

```
                                         grad_samples = 10,
                                         elbo_samples = 10,
                                         algorithm = "meanfield",
                                         output_dir = NULL,
                                         iter = 1000,
                                         adapt_iter = 20,
                                         save_latent_dynamics=TRUE,
                                         tol_rel_obj = 10^-4)
```

The "LDA.stan" file contains the `Stan` script for the LDA model provided in Section 4.1. We recommend the usage of the `R` help to get familiar with the `variational()` method of the `cmdstan_model()` function. The variable `vi_fit` contains the results of variational approximation of the LDA parameters. For example, one can obtain the word distributions for the each of the topics with `vi_fit$summary("phi")`.

Finally, to access the ELBO values, use the following:

```
vi_diag <- utils::read.csv(vi_fit$latent_dynamics_files()[1],
                           comment.char = "#")
ELBO <- data.frame(Iteration = vi_diag[,1], ELBO = vi_diag[,3])
```

**Task 1: Use a graphical display to show the 10 most common words for each of the two-topics and their probabilities.**

**Task 2: Use the function `wordcloud()` from the `wordcloud` package and display the most common words for each of the topics as a world clowd. What kinds of articles do these topics represent?**

**Task 3: Fit a three-topic LDA model, display the most common words for each of the topics. How do the results differ from the two-topic LDA?**

**Task 4 (Advanced): Use the three-topic LDA model and diplay the topic prevalence among the 2246 Associated Press articles. That is, show what proportions of articles fall under each topic.**

All necessary R code for fitting the LDA model to the Associated Press sample, including the graphical displays shown in the main text, is included in a separate R script file called LDA_LAB.R available as a part of the supplementary materials. We also include a printout of the R script below for interested readers.

```
library(cmdstanr)

# Checking integrity of installation of cmdstanr
check_cmdstan_toolchain()
install_cmdstan(cores = 2)
cmdstan_path()
cmdstan_version()

# Auxiliary packages
library(tm)
library(tidyverse)
library(tidytext)
library(topicmodels)
```

```
## Get data
data("AssociatedPress", package = "topicmodels")

## Removing rare words from the vocabulary
dtm <- removeSparseTerms(AssociatedPress, 0.95)
dim(dtm)

## Input for stan model
N_TOPICS <- 2

data <- list(K = N_TOPICS,
             V = dim(dtm)[2],
             M = dim(dtm)[1],
             N = sum(dtm$v),
             w = rep(dtm$j,dtm$v),
             doc = rep(dtm$i,dtm$v),
             #according to Griffiths and Steyvers(2004)
             alpha = rep(50/N_TOPICS,N_TOPICS),
             beta = rep(1,dim(dtm)[2])
)

### VB fit
LDA_model_cmd <- cmdstan_model(stan_file = "LDA.stan")
LDA_model_cmd$print()

vb_fit <- LDA_model_cmd$variational(data = data,
                                    seed = 1,
                                    output_samples = 1000,
                                    eval_elbo = 1,
                                    grad_samples = 10,
```

```
                                    elbo_samples = 10,
                                    algorithm = "meanfield",
                                    output_dir = NULL,
                                    iter = 1000,
                                    adapt_iter = 20,
                                    save_latent_dynamics=TRUE,
                                    tol_rel_obj = 10^-4)

# Plotting ELBO
vb_diag <- utils::read.csv(vb_fit$latent_dynamics_files()[1],
                           comment.char = "#")
ELBO <- data.frame(Iteration = vb_diag[,1],
                  ELBO = vb_diag[,3])

ggplot(data = ELBO, aes(x = Iteration, y = ELBO)) + geom_line(lwd=1.5) +
  theme(text = element_text(size = 20),
        panel.background = element_rect(fill = "transparent",
                                        color = "lightgrey"),
        panel.grid.major = element_line(colour = "lightgrey")) +
  xlim(0,110)

## Accessing parameters
vb_fit$summary("theta") # dim: M-by-K
vb_fit$summary("phi") # dim: K-by-V

## Word distribution per topic
V <- dim(dtm)[2]
odd_rows <- rep(c(1,0), times = V)
Topic1 <- vb_fit$summary("phi")[odd_rows == 1,]
Topic2 <- vb_fit$summary("phi")[odd_rows == 0,]
```

```
word_probs <- data.frame(Topic = c(rep("Topic 1", V),
                                   rep("Topic 2", V)),
                         Word = rep(dtm$dimnames$Terms,N_TOPICS),
                         Probability = c(Topic1$mean, Topic2$mean))

# Selecting top 10 words per topic
top_words <- word_probs %>% group_by(Topic) %>% top_n(10) %>%
  ungroup() %>% arrange(Topic, -Probability)

top_words %>%
  mutate(Word = reorder_within(Word, Probability, Topic)) %>%
  ggplot(aes(Probability, Word, fill = factor(Topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ Topic, scales = "free") +
  scale_y_reordered() + theme(text = element_text(size = 15)) + xlim(0,0.025) +
  xlab("Word distributions ( \u03d5 )")

# Word Cloud display
#install.packages("wordcloud")
library(wordcloud)

top_words <- word_probs %>% group_by(Topic) %>% top_n(20) %>%
  ungroup() %>% arrange(Topic, -Probability)

mycolors <- brewer.pal(8, "Dark2")
wordcloud(top_words %>% filter(Topic == "Topic 1") %>% .$Word ,
          top_words %>% filter(Topic == "Topic 1") %>% .$Probability,
          random.order = FALSE,
          color = mycolors)
```

```r
mycolors <- brewer.pal(8, "Dark2")
wordcloud(top_words %>% filter(Topic == "Topic 2") %>% .$Word ,
          top_words %>% filter(Topic == "Topic 2") %>% .$Probability,
          random.order = FALSE,
          color = mycolors)
```

# References


Albert, J. & Hu, J. (2019), *Probability and Bayesian Modeling*, 1 edn, Chapman and Hall/CRC.

Mroz, Thomas, A. (1987), 'The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions', *Econometrica* **55**, 765–799.

Stan Development Team (2022), *Stan Modeling Language User's Guide and Reference Manual, Version 2.31*.