# TopicTimes2

## DCT

- proportions that change conditional on new information,
- the dynamic diseaes probabilities within a topic in a pinrcipeld fashion
- a rate of prgression through these topics that is conditional on some covariate

Recall

- Traditional LDA uses Dirichlet-distributed $\theta$ for topic proportions.
- Dynamic models use a logistic normal distribution for document-specific topic proportions.
- This change allows capturing uncertainty over time with mean $\alpha$.
- The evolution of $\alpha_t$ follows a Gaussian process, offering temporal dynamics.

For a topic model with $K$ topics and $V$ terms:

- Let $\beta_{t,k}$ be the V-vector of natural parameters for topic $k$ at time $t$.
- The natural parameter for the multinomial is the log odds: $\beta_i = \log(\frac{\pi_i}{\pi_V})$
- Dirichlet distributions are common but not ideal for sequential data.
- We use a state-space model with Gaussian noise for temporal evolution of $\beta_{t,k}$.

Sequential topic modeling involves:

1. Drawing topics $\beta_t$ from a Gaussian distribution with mean $\beta_{t-1}$.
2. Drawing $\alpha_t$ for topic proportions, evolving as a Gaussian process.
3. For each document:
    - Draw topic mixture $n \sim \mathcal{N}(\alpha_t, a^2 I)$.
    - For each word, draw from a multinomial distribution parameterized by $\pi(\beta_{t,z})$.

1. **Define the simulation parameters**:
    - Number of topics (KK) and number of documents (DD).
    - The vocabulary size (VV) for each topic.
    - The number of time steps (TT) to simulate the evolution of topics over time.
2. **Simulate correlated topic proportions**:
    - Use a multivariate normal distribution to generate correlated topic proportions, , for each document.
    - Each document's topic proportion vector d d will be drawn from N( ,Σ)N( ,Σ), where is the mean topic distribution and ΣΣ is the covariance matrix that models the correlation between topics.
3. **Model dynamic disease probabilities within a topic**:
    - For each topic, simulate the change in word probabilities over time, which could represent the evolution of disease states.
    - This could involve a random walk or another time-series model for the probabilities associated with each word in the vocabulary of a topic.
4. **Model progression through topics**:
    - The progression rate through topics can be modeled as a function of a covariate, such as age or another relevant variable.
    - This could be done by adjusting the mean over time for each document based on the covariate.
5. **Generate documents based on the evolving topics**:
    - At each time step, for each document, draw topic assignments and words based on the current state of the topic proportions and word probabilities.
6. **Incorporate new information**:
    - When new diagnoses (information) arrive, update the topic proportions for the affected documents accordingly.
7. **Implement in R**:
    - Write the R code that uses the above steps to generate synthetic data.
    - Use loops or apply functions to iterate over documents and time steps.

This outline provides a conceptual framework for the simulation. Implementing it would require substantial programming effort and a deep understanding of the statistical methods involved. The actual R code would be quite lengthy and beyond the scope of a single response, but this plan gives you a roadmap to begin coding.

```
K <- 10 # Number of topics
D <- 100 # Number of documents
V <- 500 # Vocabulary size
T <- 50 # Number of time steps

# Objective 2: Simulate correlated topic proportions
# Assuming the existence of a function that generates correlated topic proportions
simulate_correlated_proportions <- function(mu, Sigma) {
  # Implement the function to draw from a multivariate normal distribution
}

# Objective 3: Model dynamic disease probabilities within a topic
simulate_dynamic_word_probs <- function(beta, time_steps) {
  # Implement the function to simulate changes in word probabilities over time
}

# Objective 4: Model progression through topics
progression_rate <- function(covariate, time) {
  # Define the progression rate as a function of some covariate
}

# Objectives 5 & 6: Generate documents & Update topic proportions with new information
generate_documents <- function(D, V, T, progression_rate, update_info) {
  documents <- list()
  for (t in 1:T) {
    for (d in 1:D) {
      # Draw topic proportions theta_d,t
      theta_d_t <- simulate_correlated_proportions(mu, Sigma)
      # Update theta_d_t with new information if available
      if (update_info$available) {
        theta_d_t <- update_topic_proportions(theta_d_t, update_info$data)
      }
      document <- c()
      for (n in 1:V) {
        # Draw topic assignment Z and word W for each word in the document
        Z <- sample(1:K, 1, prob = theta_d_t)
        W <- sample(1:V, 1, prob = beta[Z,])
```

```
        document <- c(document, W)
      }
      documents[[paste0('Document_', d, '_Time_', t)]] <- document
    }
  }
  return(documents)
}

# Update function for topic proportions given new information
update_topic_proportions <- function(theta, new_data) {
  # Bayesian updating of theta using new_data
  # This is a placeholder for the update logic
  updated_theta <- theta # Update logic goes here
  return(updated_theta)
}

# Example: Running the simulation with a covariate influencing progression rate
covariate <- runif(D) # Random covariate, for example
documents <- generate_documents(D, V, T, progression_rate, update_info)
```