In general, we are looking to analyze what is a conceptually simple concept: given two random variables X and Y that have some relation to each other, what is the expected value of X and how can we define a joint probability density function for X and Y , given Y = ψ. Being able to do this will allow us to make inferences and learn about the behavior of X given what we know about Y .

For Latent Dirichlet Allocation (LDA), we are given a set of text documents and then are attempting to determine the topic of a given document based on the words in the document. For each topic, we derive a probability model. Then, using Bayes' rule, we are able to determine which topic has the highest probability of belonging to any particular document. To do this, we are looking to calculate

$$P(W,Z,\theta,\varphi|\alpha,\beta)$$

where we have M = number of documents, N = number of words per document, V = number of unique words, K = the number of topics, W = vector with each word in a document represented by a number from 1 to V , Z = identity of topic of a word from 1 to K, α= the likelihood of a given topic in a given document, β = the likelihood of a word occurring in a topic, θ is a random variable from the Dirichlet distribution with parameter α and φ is a random variable from the Dirichlet distribution with parameter β.

However, P (Z, θ, φ|W, α, β) cannot be done in closed form. To make this calculation easier, we first integrate out θ and φ, which we do because they are Dirichlet random variables with known probability distribution functions. This process is called collapsing, and is useful in helping derive the inference algorithm to determine the overall topic of a document. Then, after more simplifications utilizing the Dirichlet distribution function and the Gamma function, we are able to conclude that

$$P(z_k = j|\mathbf{Z}_{-\mathbf{k}}, \mathbf{W}) \propto (\mathbf{n}_{\mathbf{m}}^{\mathbf{k},-(\mathbf{m},\mathbf{n})} + \alpha_{\mathbf{k}}) \frac{\mathbf{n}_{\mathbf{v}}^{\mathbf{k},-(\mathbf{m},\mathbf{n})} + \beta_{\mathbf{v}}}{\sum_{\mathbf{r}-\mathbf{1}}^{\mathbf{V}} \mathbf{n}_{\mathbf{r}}^{(\mathbf{k},-(\mathbf{m},\mathbf{n}))} + \beta_{\mathbf{r}}}$$

Which in words means that we can multiply the probability of a word occurring in a topic with the probability of that topic occurring in a document to determine the probability of a document having a given topic.

However, this is still difficult to compute in closed form. Instead, we run Monte Carlo simulation using Markov chains and a Gibbs sampler. We are sampling the Z variable to make an estimate as to its current assignment, with one matrix of MxN, representing all of the topic allocations. The code would go into each of the (i,j) and computes the probability of Z belonging to any K. We would also use a matrix of size MxK to keep track of the probability of topics occurring in a document and a matrix of size KxV to keep track of the probability of a word occurring in a topic.

The Gibbs sampler then is calculating from P(Z0|W = document,Z\Z0) via Monte Carlo simulation. Then, the code comes down to multiplying the corresponding entries of the MxK and KxV , divided by the corresponding entry in the MxN matrix. Once we have calculated all of the Zi's, we can use the values to then go back and compute the φ and θ.

For the time complexity of the LDA algorithm, it is in O(M*V*K) as defined above. While it's not exponential, it's not great either.