

=====

<CC> Pseudocode

=====

read_data

```
// Allocations
corpus->words[doc_length]
corpus->counts[doc_length]
parse the documents sequentially: access stride 1.
```

run_em

```
// Allocations
var_gamma[num_docs][NTOPICS]
phi[max_length][NTOPICS]
```

new_lda_model

```
model->log_prob_w[num_topics][num_terms]
```

new_lda_suffstats

```
lda_suffstats->class_total[num_topics]
lda_suffstats->class_word[][num_topics][num_terms]
```

```
// Initialisations
```

corpus_initialize_ss

```
for k = 0 : num_topics
  for i = 0 : NUM_INIT
    for n = 0 : doc_length
      ss->class_word[k][doc->words[n]] += doc->counts[n];
    for n = 0 : model->num_terms
      ss->class_word[k][n] += 1.0;
      ss->class_total[k] = ss->class_total[k] + ss->class_word[k][n];
```

lda_mle

```
for k = 0 : model->num_topics
  for w = 0 : model->num_terms
    model->log_prob_w[k][w] = log(ss->class_word[k][w]) - log(ss->class_total[k])
```

```
while !converged
```

zero_initialize_ss

```
for k = 0 : model->num_topics
  ss->class_total[k] = 0
  for w = 0 : model->num_terms
    ss->class_word[k][w] = 0
```

```
for d = 0 : corpus->num_docs
```

doc_e_step

lda_inference

```
for k = 0 : model->num_topics
  var_gamma[k] = model->alpha + (doc->total/((double) model->num_topics));
  digamma_gam[k] = digamma(var_gamma[k])
  for (n = 0 : doc->length
    phi[n][k] = 1.0/model->num_topics
  while !converged
    for n = 0 : doc->length
      for k = 0 : model->num_topics
        oldphi[k] = phi[n][k]
```

```

// Shity access
phi[n][k] = digamma_gam[k] + model->log_prob_w[k][doc->words[n]]
if (k > 0)
    phisum = log_sum(phisum, phi[n][k])
else
    phisum = phi[n][k]

```

```

for k = 0 : model->num_topics
    phi[n][k] = exp(phi[n][k] - phisum)
    var_gamma[k] = var_gamma[k] + doc->counts[n]*(phi[n][k] - oldphi[k])
    digamma_gam[k] = digamma(var_gamma[k])

```

compute_likelihood

```

for k = 0 : model->num_topics
    dig[k] = digamma(var_gamma[k])
    var_gamma_sum += var_gamma[k]
for k = 0 : model->num_topics
    likelihood += (model->alpha - 1)*(dig[k] - digsum)
                + lgamma(var_gamma[k])
                - (var_gamma[k] - 1)*(dig[k] - digsum)

for n = 0 : doc->length
    if (phi[n][k] > 0)
        likelihood += doc->counts[n]*
                    (phi[n][k]*((dig[k] - digsum) - log(phi[n][k]))
                    + model->log_prob_w[k][doc->words[n]])

```

```

for k = 0 : model->num_topics
    gamma_sum += gamma[k];
    ss->alpha_suffstats += digamma(gamma[k])
for n = 0 : doc->length
    for k = 0 : model->num_topics
        ss->class_word[k][doc->words[n]] += doc->counts[n]*phi[n][k]
        ss->class_total[k] += doc->counts[n]*phi[n][k]

```

lda_mle

```

for k = 0 : model->num_topics
    for w = 0 : model->num_terms
        model->log_prob_w[k][w] = log(ss->class_word[k][w]) - log(ss->class_total[k])

```