Appendix

Scalability

In Figure 7-Left we plot run times for the experiments in Figures 3-5, where we can see that DC is approximately 2.5 times slower than standard LDA.

We also plot run times for DC with different number of topics (Figure 7-Center) and different number of documents (Figure 7-Right), and as can be seen it scales linearly with the number of topics and sublinearly with the number of documents.

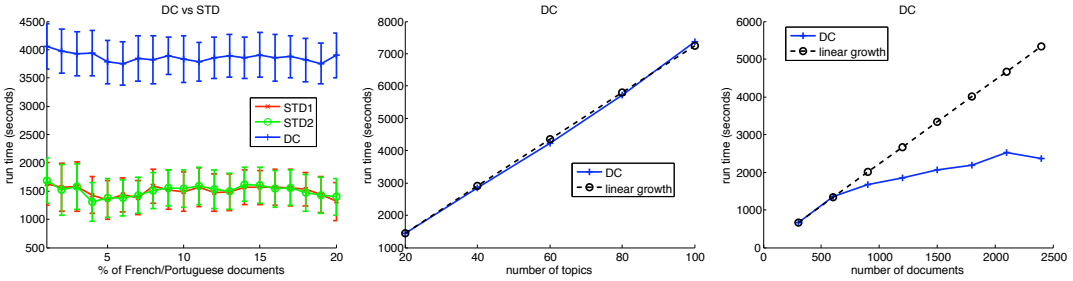

Figure 7: Left: run times for the experiments in Figures 3-5. Center: run times versus number of topics for downstream conditioning. Right: run times versus number of documents for downstream conditioning.

Fast sampling

Following [19] we decompose eq. 6 in three terms:

$$p(z_{mn} = k|\text{rest}) \propto \underbrace{\beta_{kv} \frac{\alpha_k}{n^{\text{K}}_{k-} + \bar{\beta}_k}}_{:=A_k} + \underbrace{\beta_{kv} \frac{n^{\text{KM}}_{km-}}{n^{\text{K}}_{k-} + \bar{\beta}_k}}_{:=B_k} + \underbrace{\frac{n^{\text{KV}}_{kv_{mn}-} \left[n^{\text{KM}}_{km-} + \alpha_k\right]}{n^{\text{K}}_{k-} + \bar{\beta}_k}}_{:=C_k}$$

Unlike in the generic case [4], some of the normalizations are now dependent on the *word* in question (via the β smoother). It is therefore not possible to apply the sampler of [19] directly, since all terms now depend on both the topics and the words via β_{kv}. We modify it as follows: $A := \sum_k A_k$ only depends on w_{mn} in a *multiplicative* fashion via β_{kv} and it is constant throughout the document otherwise. In $B := \sum_k B_k$ only two terms need updating whenever we reassign a word to a new topic (and we have a new multiplicative constant for different words via β_{kv}). Hence, the only term that needs full recomputing for each word is $C := \sum_k C_k$. Fortunately, this is nonzero for only a small number of topics for a given word.