Probabilistic LSA (pLSA)

- Matrix factorization is a very practical approach to document retrieval, however it isn't principled in a statistical sense. At a high level, we start with an extremely sparse term-document matrix and ask, how can we measure similarity between documents? To do this we compute a projection onto a K dimensional space where the data is represented by a dense(r) matrix, and compute distances on it.
- pLSA uses a latent variable approach to model terms and documents as a joint distribution $p(w, d)$ by factoring it using a set of latent variables Z:

$$p(d, w) = \sum_{z \in Z} p(z)p(d|z)p(w|z)$$

- The standard MLE method to fit latent variable models is the EM algorithm (general form is shown here, pLSA actually uses a more involved EM-based approach):

E step: $$p(z|d, w) = \frac{p(z)p(d|z)p(w|z)}{\sum_{z' \in Z} p(z')p(d|z')p(w|z')}$$

M step:
$$p(w|z) \propto \sum_{d \in D} f_{d,w} p(z|d, w)$$
$$p(d|z) \propto \sum_{w \in W} f_{d,w} p(z|d, w)$$
$$p(z) \propto \sum_{d \in D} \sum_{w \in W} f_{d,w} p(z|d, w)$$

[1] Hoffman, 2000

Practical issues with pLSA

- There are a few problems that arise when using pLSA in practice:
  - There is no probabilistic interpretation at the document level; each document is represented as a list of numbers and there is no generative model for them.
  - Parameters grows linearly with the size of the corpus, which leads to overfitting
  - No natural way to assign probability to a document outside of the training set
  - Unlike NMF, there is no clear way to impose sparsity on our latent representation

[1] Blei, 2003