VIP Topic Modelling Team

I. Formulae/Algorithms

A. pLSI

With pLSI we are modelling:

- Document as a mixture of topics, i.e., $P(z|d)$
- Topic as a distribution over words, i.e., $P(w|z)$

We are given a corpus $\mathcal{D}$ consisting of N documents $d_i, i \in \{1, ..., N\}$ (or d for short), which consists of a set of M unique words (i.e., vocabulary) $\mathcal{W}$ consisting of $w_j, j \in \{1, ..., M\}$ (or w for short) where each word may appear more than once.

Through our algorithm, we derive a set of K topics $\mathcal{Z}$ consisting of $z_k, k \in \{1, ..., K\}$ (or z for short).

The algorithm is mostly in agreement with [1], but we also use [2] and [3] as reference.

We derive the following distributions:

- $P(z|d, w)$
- $P(w|z)$
- $P(z|d)$

1) Inputs:

- K: Number of topics
- $n(\cdot, \cdot)$: 2D array of documents and words, where each row represents one document and the columns are number of occurrences each word corresponding to the (unique) word for that column.

2) $P(z|d, w)$:

$$P(z|d, w) = \frac{P(w, z, d)}{P(d, w)} \tag{1}$$

Numerator:

$$\begin{aligned} &P(w, z, d) \\ &\quad = P(w|z, d)P(z|d)P(d) \text{ (chain rule)} \\ &\quad = P(w|z)P(z|d)P(d) \text{ (conditional dependence assumption)} \end{aligned} \tag{2}$$

Denominator:

$$\begin{aligned} &P(d, w) \\ &\quad = \sum_{z' \in \mathcal{Z}} P(w, z', d) \\ &\quad = \sum_{z' \in \mathcal{Z}} P(w|z')P(z'|d)P(d) \end{aligned} \tag{3}$$

So we have:

$$\begin{aligned} P(z|d, w) &= \frac{P(w|z)P(z|d)P(d)}{\sum_{z' \in \mathcal{Z}} P(w|z')P(z'|d)P(d)} \\ &= \frac{P(w|z)P(z|d)}{\sum_{z' \in \mathcal{Z}} P(w|z')P(z'|d)} \end{aligned} \tag{4}$$

3) $P(w|z)$: From equation (4) in [2]:

$$P(w|z) \propto \sum_{d \in \mathcal{D}} n(d, w)P(z|d, w) \tag{5}$$

Since this is a probability distribution over words, we can easily derive:

$$P(w|z) = \frac{\sum_{d \in \mathcal{D}} n(d, w)P(z|d, w)}{\sum_{w' \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(d, w')P(z|d, w')} \tag{6}$$

4) $P(z|d)$:

$$\begin{aligned} P(z|d) &= \frac{P(z,d)}{P(d)} \\ &= \frac{\sum_{w\in\mathcal{W}} P(z|d,w)P(d,w)}{P(d)} \\ &= \frac{\sum_{w\in\mathcal{W}} P(z|d,w)P(w|d)P(d)}{P(d)} \\ &= \sum_{w\in\mathcal{W}} P(z|d,w)P(w|d) \end{aligned} \tag{7}$$

$$P(w|d) = \frac{n(d,w)}{\sum_{w'\in\mathcal{W}} n(d,w')} \tag{8}$$

Note: for each document, computing

$$P(w|d) \propto n(d,w) \tag{9}$$

and normalizing over the word dimension achieves the same result (during the M step we need to normalize to ensure a valid distribution anyway):

$$P(z|d) \propto \sum_{w\in\mathcal{W}} n(d,w)P(z|d,w) \tag{10}$$

5) Log likelihood: Likelihood:

$$L = \prod_{d\in\mathcal{D},w\in\mathcal{W}} P(w,d)^{n(d,w)} \tag{11}$$

Since we don't have $P(w,d)$, we use $P(w|d)$ in its place:

$$\begin{aligned} \log L' &= \sum_{d\in\mathcal{D},w\in\mathcal{W}} n(d,w)\log P(w|d) \\ &= \sum_{d\in\mathcal{D},w\in\mathcal{W}} n(d,w)\log[\sum_{z\in\mathcal{Z}} P(w|z)P(z|d)] \text{ (see (3))} \end{aligned} \tag{12}$$

B. Clustering Metrics

During training, we keep track of clustering scores in addition to log likelihood. We derive the clusterings based on the probability distributions derived during training (e.g., $P(w|z)$, $P(z|d)$). Ref: [4].

1) Topic clustering: One way to cluster topics is based on the probability distributions $P(w|z)$ following these steps:

1) For each topic, sort the words by their probability in descending order.
2) Choose top words, based on fixed count (e.g., $(vocabsize)/(\#topics)$), or with Top P (e.g., 90%).

We will use a fixed count for each topic cluster:

$$c = (vocabsize)/(\#topics) \tag{13}$$

for I-B3 and I-B4 such that there exist bijective mappings from clusters to vocabulary.

2) V-measure: V-measure is computed as follows:

$$v = \frac{(1+\beta)\times homogeneity \times completeness}{\beta\times homogeneity + completeness} \tag{14}$$

We use a beta value of 1.0.

3) Completeness (proxy): Completeness measures how much of the data points from each true class are assigned to the same cluster. In other words, it evaluates how well all data points from a specific true class are grouped together in the same predicted cluster. Since this measure requires another clustering (or the ground truth clusters in addition to our derived clusters), we instead derive a proxy measure as follows:

$$completeness = (total\ \#words\ covered\ by\ all\ topic\ clusters)/(vocabulary\ size) \tag{15}$$

4) Homogeneity (proxy): Homogeneity measures how well all data points within a predicted cluster belong to the same true class. It evaluates the consistency of the predicted clusters, aiming to ensure that each cluster contains as many points as possible from a single true class. Since this measure requires another clustering (or the ground truth clusters in addition to our derived clusters), we instead derive a proxy measure as follows:

$$homogeneity = \sum_{z \in \mathcal{Z}} \frac{1}{|\mathcal{Z}|} \frac{(topic\ score)}{(\#words\ in\ topic\ z's\ cluster)} \tag{16}$$

We define two distinct $(topic\ score)$s:

- binary: # words unique to the current topic cluster.
- reciprocal:

$$\sum_{word\ w\ in\ current\ topic\ cluster} (1 - \frac{\#occurrences\ of\ w\ in\ all\ other\ topic\ clusters}{|\mathcal{Z}| - 1}) \tag{17}$$

REFERENCES

[1] Z. Zhang, "Plsa (probabilistic latent semantic analysis)," https://github.com/laserwave/plsa.

[2] T. Hofmann *et al.*, "Probabilistic latent semantic analysis." in *UAI*, vol. 99, 1999, pp. 289–296.

[3] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 50–57. [Online]. Available: https://doi.org/10.1145/312624.312649

[4] S. learn developers, "Clustering — scikit-learn documentation," 2025, accessed: 2025-04-08. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html