# Advanced Probabilistic Machine Learning and Applications

### Tutorial 2: Categorical Mixture Model (CMM) + EM

The problem we tackle in this tutorial is finding clusters in Twitter data, i.e. grouping tweets with similar content. For this purpose, we first describe a mixture model for categorical data. Secondly, we use the expectation-maximization (EM) algorithm to find the maximum likelihood (ML) estimate of the parameters of our model. Then, we implement the algorithm in Python and we test it using Twitter data. Finally, we show the results in a Jupyter Notebook.

### Intro: Dataset and Model specifications

Through this document we will use the following notation:

- $K$ : number of mixture components, i.e., we interpret as topics/clusters.
- $N$ : number of documents, i.e. tweets.
- $I$ : dictionary.
- $|I|$ : number of words in $I$.
- $\Theta = \{\pi, \{\theta_k\}\}$ : set of all the parameters of the model.
- $[x_i = i]$ : function that takes the value 1 if $x = i$ and 0 otherwise.

**About Data set:** The dataset $\{\mathbf{x}_n\}_{n=1}^N$ consist of $N$ documents (tweets) generated by $U$ users. Each of them have been cleaned and processed, i.e., lematization, lowerization and stemming, using a dictionary, denoted by $I$. We represent each document as $\mathbf{x}_n = (x_{n1}, \dots, x_{nW_n})$, that is a vector of $W_n$ words. Each word $x_{nj} \in \{1, \dots, |I|\}$ is represented with its position in the dictionary.

**Model**: For the data described above, consider the following **probabilistic mixture model**:

$$p\left(\{\mathbf{x}_n\}_{n=1}^N|\Theta\right) = \prod_{n=1}^N \sum_{k=1}^K \pi_k p(\mathbf{x}_n|\theta_k) \quad \text{where} \quad p(\mathbf{x}_n|\theta_k) = \prod_{j=1}^{W_n} Cat(x_{nj}|\theta_k) \tag{1}$$

where $\pi \in R^K$ are the mixing proportions (which must be positive and sum to one) and each $\theta_k = (\theta_{k1}, \dots, \theta_{k|I|})$ represents the probabilities of the words in the dictionary for a given topic $k$. Thus, the parameter $\theta_{km}$ represents the probability of the word $i$ in the topic (cluster) $k$.
**Material:** Copy the Jupyter notebook available in the Github repository `https://github.com/APMLA-2021/APMLA-2021_material/tree/main/L2` and complete the exercises proposed below.

### Exercise 1: Derive the EM-Algorithm for the CMM

Given the dataset and the probabilistic model described in the previous section:

1. Write down the expression for the log joint distribution of the latent and observed variables, i.e., $\log p\left(\{\mathbf{x}_n\}_{n=1}^N|\Theta\right)$.
2. Compute the closed-form expression for the E-step, i.e., $Q(\Theta, \Theta^{old})$.
   Hint: $Q(\Theta, \Theta^{old}) = \sum_{n=1}^N \sum_{k=1}^K p(z_n = k|\mathbf{x}_n, \Theta) \log p(\mathbf{x}_n, z_n = k|\Theta^{old}) \approx \log p\left(\{\mathbf{x}_n\}_{n=1}^N|\Theta\right)$
3. M step: Derive the expression of the MLE for the model parameters $\Theta = \{\pi, \{\theta_k\}\}$.

### Exercise 2: Data exploration task on Twitter data (for CMM)

You can find a Python implementation of the EM algorithm for the CMM in `https://github.com/APMLA-2021/APMLA-2021_material/tree/main/L2` . Get familiar with it and afterwards complete the following tasks in the Jupyter notebook:

1. Fix $K = 5$ and try different initializations.
2. Cross validate the number of components $K \in 2, 3, 5, 10$ according to the Akaike Information Criterion (AIC). You will need to implement the formula for the AIC. Hint: $\text{AIC} = 2|\Theta| - 2\log p\left(\{\mathbf{x}_n\}_{n=1}^N|\Theta\right)$
3. For the optimal value of $K$ according to AIC, show
   i. the (approximated) log-likelihood evolution per iteration,
   ii. the 10 most representative words for each topic using a cloud of words, and
   iii. the 10 most relevant documents for each topic.

### Exercise 3: Data exploration task on Pizza data (GMM)

Implement the GMM for the Pizza dataset.