Clustering with Multinomial Outcomes

Shiro Kuriwaki*

October 2019

1 Data Generating Process

Setup Index individuals by $i \in \{1, ..., N\}$ and the universe of races excluding the top of the ticket as $j \in \{1, ..., D\}$. The data we observe is a length-D vector of votes $\mathbf{Y}_i$. Y_{ij} is a discrete response value, $Y_{ij} \in \{0, ..., L\}$. For now, let's use $L = 1$ so that each vote y_{ij} be a binary variable for splitting their ticket or not. $Y_{ij} = 1$ would mean voter i splitting their ticket in some office j, with reference to a top of the ticket office like the President or Governor.

Parameters Individuals are endowed with a cluster (or type) $k \in 1 : K$, which is drawn from

$$Z_i \sim \text{Cat}(\boldsymbol{\theta}),$$

Where the length-K simplex $\boldsymbol{\theta}$ is called the mixing proportion.

Let $\mu_{k,j} \in [0, 1]$ be the parameter that governs each observation through the cluster assignment. Therefore $\boldsymbol{\mu}$ is a $\{K \times D \times (L + 1)\}$ array, where

$$\Pr\left(Y_{ij} = \ell \mid Z_i = k\right) = \mu_{kj\ell}.$$

In other words, for each individual who belongs in type k, we can their observed vector $\mathbf{Y}_i \mid Z_i = k$ is governed by a length-D parameter vector $\boldsymbol{\mu}_k$. Therefore, we can express the joint density as follows.

$$\Pr\left(\mathbf{Y}_i \mid Z_i = k, \boldsymbol{\theta}\right) = \prod_{j=1}^{D} \text{Cat}(Y_{ij} | \boldsymbol{\mu}_k) = \prod_{j=1}^{D} \prod_{\ell=1}^{L} \mu_{jk\ell}^{\mathbf{1}(Y_{ij}=\ell)} \tag{1.1}$$

In our $L = 1$ case, the Categorical reduces to a Bernoulli and we can get rid of the ℓ index altogether to let $\mu_{jk} \equiv \mu_{jk,(\ell=1)} = 1 - \mu_{jk,(\ell=0)}$ and so

*Thanks to Shusei Eshima, Sooahn Shin, and Soichiro Yamauchi for their help.

$$\Pr\left(\mathbf{Y}_i \mid Z_i = k, \boldsymbol{\theta}\right) = \prod_{j=1}^{D} \mu_{jk}^{Y_{ij}} (1 - \mu_{jk})^{1 - Y_{ij}} \tag{1.2}$$

The benefit of this modeling exercise over that from a naive sample of $N \times D$ Bernoullis is that we have captured the correlations between variables.[1]

2 EM

This mixture model lends itself to clustering analysis such as K-means. Although we can estimate this model in a Bayesian fashion by setting a prior for $\boldsymbol{\theta}$ and $\boldsymbol{\mu}$, the Stan program cannot reliably estimate clustering models like this one by MCMC because of label-switching and multimodality.

Because the model is simple enough, we can derive an algorithm to obtain the global solution for the parameters.[2] An EM implementation makes it possible to handle extensions, such as systematic missing data, multinomial outcomes, and covariates.

Complete Likelihood If we knew the cluster assignment, we would be able to write the complete likelihood ($\mathbf{L}_{\text{comp}}$) and the complete log-likelihood ($\mathcal{L}_{\text{comp}}$),

$$\begin{aligned}
\mathbf{L}_{\text{comp}}(\boldsymbol{\mu}, \boldsymbol{\theta} | \mathbf{Y}, \mathbf{Z}) &= \prod_{i=1}^{N} \prod_{j=1}^{D} \prod_{k=1}^{K} \Pr(Z_i = k) \prod_{\ell=1}^{L} \Pr(Y_{ij} = 1 | Z_i = k, \boldsymbol{\mu})^{\mathbf{1}(Y_{ij} = \ell | Z_i = k)} \\
\mathcal{L}_{\text{comp}}(\boldsymbol{\mu}, \boldsymbol{\theta} | \mathbf{Y}, \mathbf{Z}) &= \sum_{i=1}^{N} \sum_{j=1}^{D} \sum_{k=1}^{K} \sum_{\ell=0}^{2} \mathbf{1}\{Y_{ij} = \ell, Z_i = k\} \Big\{ \log \Pr(Y_{ij} = \ell | Z_i = k, \boldsymbol{\mu}) \Big\} \\
&\quad + \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbf{1}\{Z_i = k\} \log \Pr(Z_i = k | \boldsymbol{\theta})
\end{aligned} \tag{2.1}$$

We first take expectations over the latent variable Z_i,

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_{\text{comp}}\right] &= \sum_{i=1}^{N} \sum_{j=1}^{D} \sum_{k=1}^{K} \sum_{\ell=0}^{2} \mathbf{1}\{Y_{ij} = \ell\} \mathbb{E}\left[\mathbf{1}\{Z_i = k\}\right] \Big\{ \underbrace{\log \Pr(Y_{ij} = \ell | Z_i = k, \boldsymbol{\mu})}_{\equiv \log \boldsymbol{\mu}_{kj\ell}} \Big\} \\
&\quad + \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{E}\left[\mathbf{1}\{Z_i = k\}\right] \underbrace{\log \Pr(Z_i = k | \boldsymbol{\theta})}_{= \log \boldsymbol{\theta}_k}
\end{aligned} \tag{2.2}$$

[1] That dependency can be expressed as $\mathbb{E}(\mathbf{y}) = \sum_{k=1}^{K} \theta_k \boldsymbol{\mu}_k$ and $\text{Cov}(\mathbf{y}) = \sum_k \theta_k (\boldsymbol{\Sigma}_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top) - \mathbb{E}(\mathbf{y}) \mathbb{E}(\mathbf{y})^\top$, where $\Sigma_k = \text{diag}(\mu_{jk}(1 - \mu_{jk}))$.

[2] Thanks to Soichiro Yamauchi for deriving this algorithm in the original iteration.

Let's define this unkown quantity as

$$\zeta_{ik} \equiv \mathbb{E}\left[\mathbf{1}\left(Z_i = k\right)\right].$$

Then the E-step can be the normalized version of the posterior probability marginalized by the mixing proportion,

$$\widehat{\zeta}_{ij} \propto \boldsymbol{\theta}_k \prod_{j=1}^{D} \underbrace{\prod_{\ell=0}^{2} \left(\boldsymbol{\mu}_{kj,\ell}\right)^{\mathbf{1}(Y_{ij}=\ell)}}_{\boldsymbol{\mu}_{kj,Y_{ij}}} \tag{2.3}$$

The M-step is derived by taking the derivatives of $\mathbb{E}\left[\mathcal{L}_{\text{comp}}\right]$ with respect to the model parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$. This leads to a MLE-like M-step

$$\widehat{\theta}_k = \frac{1}{N} \sum_{i=1}^{N} \zeta_{ik} \tag{2.4}$$

$$\widehat{\mu}_{jk\ell} = \frac{\sum_{i=1}^{N} \mathbf{1}\left(Y_{ij} = \ell\right) \zeta_{ik}}{\sum_{i=1}^{N} \zeta_{ik}} \tag{2.5}$$

EM Implementation We first need to set initial values for $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$. I do this by letting $\boldsymbol{\theta}^{(0)} = \left(\frac{1}{K}, \ldots \frac{1}{K}\right)$, randomly assigning an initial cluster assignment $Z_i' \sim \text{Cat}(\boldsymbol{\theta}^{(0)})$, and setting the initial $\boldsymbol{\mu}$ by the sample means of the data within those initial assignments, $\mu_{jk}^{(0)} = \frac{\sum_{i=1}^{N} \mathbf{1}(Y_{ij}=1)\mathbf{1}\left(Z_i'=k\right)}{\sum_{i=1}^{N} \mathbf{1}\left(Z_i'=k\right)}$.

Then we iterate as follows.

For each voter i, compute the probability that they belong in cluster k (E-step):

$$\zeta_{ik} \leftarrow \frac{\boldsymbol{\theta}_k \prod_{j=1}^{D} \boldsymbol{\mu}_{kj,Y_{ij}}}{\sum_{k'=1}^{K} \boldsymbol{\theta}_{k'} \prod_{j=1}^{D} \boldsymbol{\mu}_{k'j,Y_{ij}}} \tag{2.6}$$

Then given those cluster probabilities, update the parameters with the MLE (M-step)

$$\widehat{\theta}_k \leftarrow \frac{1}{N} \sum_{i=1}^{N} \widehat{\zeta}_{ik} \tag{2.7}$$

$$\widehat{\mu}_{jk1} \leftarrow \frac{\sum_{i=1}^{N} \mathbf{1}\left(Y_{ij} = 1\right) \widehat{\zeta}_{ik}}{\sum_{i=1}^{N} \widehat{\zeta}_{ik}} \tag{2.8}$$

$$\widehat{\mu}_{jk0} = 1 - \widehat{\mu}_{jk1} \tag{2.9}$$

We repeat this until convergence.

Convergence We evaluate covergence by the observed log likelihood,

$$\begin{aligned}\mathbf{L}_{\text{obs}} &= \prod_{i=1}^{N}\sum_{k=1}^{K}\theta_k f_k(\mathbf{Y}_i|\boldsymbol{\mu}_k) \\ &= \prod_{i=1}^{N}\sum_{k=1}^{K}\theta_k \prod_{j=1}^{D}\boldsymbol{\mu}_{k'j,Y_{ij}}\end{aligned}$$

So the log-likelihood is

$$\mathcal{L}_{\text{obs}} = \sum_{i=1}^{N}\left[\prod_{k=1}^{K}\left\{\log\theta_k + \sum_{j=1}^{D}\sum_{\ell=1}^{L}\mathbf{1}\left(Y_{ij}=\ell\right)\log(\mu_{kj\ell})\right\}\right] \tag{2.10}$$

Calculating (eq. 2.10) is computationally intensive, so a quick way to check convergence is to track the maximum of the change in parameters which are all on the probability scale, i.e. $\max\left\{|\widehat{\theta}_1^{(t+1)} - \widehat{\theta}_1^{(t)}|, ..., |\widehat{\mu}_{K,D}^{(t+1)} - \widehat{\mu}_{K,D}^{(t)}|\right\}$

Speed-Ups Because this EM algorithm deals with discrete data, the algorithm needs only sufficient statistics. In our setting the unique number of voting profiles is much smaller than the number of observations, because vote vectors follow a systematic pattern and most votes are straight-ticket votes. Therefore, we can re-format the dataset so that each row is a unique combination.