

Clustering Voter Types with Multinomial Outcomes

Shiro Kuriwaki*

October 2019

1 Data Generating Process

Setup Index individuals by $i \in \{1, \dots, N\}$ and the universe of races excluding the top of the ticket as $j \in \{1, \dots, D\}$. The data we observe is a length- D vector of votes \mathbf{Y}_i . Y_{ij} is a discrete response value, $Y_{ij} \in \{0, \dots, L\}$.

In the simplest case $L = 1$, we code each vote y_{ij} an indicator 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. When $L = 2$, we can consider three outcomes: abstention $Y_{ij} = 0$, split $Y_{ij} = 1$, or straight $Y_{ij} = 2$.

Parameters There are two sets of parameters: $\boldsymbol{\mu}$, the propensity for a given outcome for a given type of voter in a given office; and $\boldsymbol{\theta}$, the mixing proportions of each type. Individuals are endowed with a cluster (or type) $k \in \{1, \dots, K\}$, which is drawn from a distribution governed by length- K simplex $\boldsymbol{\theta}$ (the mixing proportion).

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

Set $\mu_{k,j} \in [0, 1]$ be the parameter that governs the outcome of each type. Therefore $\boldsymbol{\mu}$ is a $\{K \times D \times (L + 1)\}$ array, where

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

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

$$\Pr(\mathbf{Y}_i \mid Z_i = k, \boldsymbol{\theta}) = \prod_{j=1}^D \text{Cat}(Y_{ij} \mid \boldsymbol{\mu}_k) = \prod_{j=1}^D \prod_{\ell=0}^L \mu_{kj\ell}^{\mathbf{1}(Y_{ij}=\ell)} \quad (1.1)$$

*Thanks to Shusei Eshima, Max Goplerud, Soohun Shin, and Soichiro Yamauchi for their generous time and help.

The loop over ℓ simply represents the categorical distribution. In the binary case, the Categorical reduces to a Bernoulli:

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

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 Clustering as an Unobserved Variable Problem: 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 log-likelihood ($\mathcal{L}_{\text{comp}}$). First start with the joint probability of the outcome data and the cluster assignment:

$$\begin{aligned} \Pr(\mathbf{Y}, \mathbf{Z} \mid \boldsymbol{\mu}, \boldsymbol{\theta}) &= \Pr(\mathbf{Y} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\theta}) \Pr(\mathbf{Z} \mid \boldsymbol{\theta}) \\ &= \prod_{i=1}^N \prod_{j=1}^D \Pr(Y_{ij} \mid \mathbf{Z}, \boldsymbol{\mu}) \prod_{i=1}^N \Pr(Z_i \mid \boldsymbol{\theta}) \\ &= \prod_{i=1}^N \prod_{j=1}^D \prod_{k=1}^K \left\{ \prod_{\ell=0}^L \Pr(Y_{ij} = \ell \mid Z_i = k) \right\}^{\mathbf{1}(Z_i=k)} \prod_{i=1}^N \prod_{k=1}^K \Pr(Z_i = k \mid \boldsymbol{\theta})^{\mathbf{1}(Z_i=k)} \\ &= \prod_{i=1}^N \prod_{j=1}^D \prod_{k=1}^K \prod_{\ell=0}^L \Pr(Y_{ij} = \ell \mid Z_i = k, \boldsymbol{\mu})^{\mathbf{1}(Y_{ij}=\ell, Z_i=k)} \prod_{i=1}^N \prod_{k=1}^K \Pr(Z_i = k \mid \boldsymbol{\theta})^{\mathbf{1}(Z_i=k)} \end{aligned}$$

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 $\boldsymbol{\Sigma}_k = \text{diag}(\mu_{kj}(1 - \mu_{kj}))$.

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

Therefore, the complete log-likelihood is computed by taking the log of this:

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

We first take expectations over the latent variable Z_i ,

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

Let's define this unknown quantity as

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

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

$$\hat{\zeta}_{ij} \propto \boldsymbol{\theta}_k \prod_{j=1}^D \underbrace{\prod_{\ell=0}^L (\boldsymbol{\mu}_{kj\ell})^{\mathbf{1}(Y_{ij}=\ell)}}_{\boldsymbol{\mu}_{kj, Y_{ij}}} \quad (2.3)$$

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

$$\hat{\theta}_k = \frac{1}{N} \sum_{i=1}^N \zeta_{ik} \quad (2.4)$$

$$\hat{\mu}_{kj\ell} = \frac{\sum_{i=1}^N \mathbf{1}(Y_{ij} = \ell) \zeta_{ik}}{\sum_{i=1}^N \zeta_{ik}} \quad (2.5)$$

EM Implementation We first need to set initial values for $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$. I do this by letting $\boldsymbol{\theta}^{(0)} = (\frac{1}{K}, \dots, \frac{1}{K})$, 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, $\mu_{kj}^{(0)} = \frac{\sum_{i=1}^N \mathbf{1}(Y_{ij}=1) \mathbf{1}(Z'_i=k)}{\sum_{i=1}^N \mathbf{1}(Z'_i=k)}$.

Then we iterate as follows. For each voter i , compute the probability that they belong in

cluster k (E-step):

$$\zeta_{ik} \leftarrow \frac{\theta_k \prod_{j=1}^D \mu_{kj, Y_{ij}}}{\sum_{k'=1}^K \theta_{k'} \prod_{j=1}^D \mu_{k'j, Y_{ij}}} \quad (2.6)$$

Then given those type probabilities, in the M-step. We can show from a Lagrangian maximization problem that the new parameters are MLEs. For updating θ_k , we take the simple average of $\hat{\zeta}_{ik}$ across all i . For updating $\hat{\mu}_{kj\ell}$, for each k and ℓ we take the sample proportion of the occurrence of $Y_{ij} = \ell$, but weighted by $\hat{\zeta}_{ik}$:

$$\text{for each } k, \text{ update: } \hat{\theta}_k \leftarrow \frac{1}{N} \sum_{i=1}^N \hat{\zeta}_{ik} \quad (2.7)$$

$$\text{for each } k, j, \ell, \text{ update: } \hat{\mu}_{kj\ell} \leftarrow \frac{\sum_{i=1}^N \mathbf{1}(Y_{ij} = \ell) \hat{\zeta}_{ik}}{\sum_{i=1}^N \hat{\zeta}_{ik}} \quad (2.8)$$

We repeat this until convergence.

Evaluating Convergence We evaluate convergence by the observed log likelihood,

$$\mathbf{L}_{\text{obs}} = \prod_{i=1}^N \sum_{k=1}^K \theta_k \prod_{j=1}^D \mu_{kj, Y_{ij}}$$

So the log-likelihood is

$$\mathcal{L}_{\text{obs}} = \sum_{i=1}^N \log \left\{ \sum_{k=1}^K \theta_k \prod_{j=1}^D \mu_{kj, Y_{ij}} \right\} \quad (2.9)$$

Calculating eq. (2.9) 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. $\max \left\{ |\hat{\theta}_1^{(t+1)} - \hat{\theta}_1^{(t)}|, \dots, |\hat{\mu}_{K,D}^{(t+1)} - \hat{\mu}_{K,D}^{(t)}| \right\}$

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.

Let $u \in \{1, \dots, U\}$ index the unique voting profiles, and n_u be the number of such profiles in the data. We re-cycle the objects \mathbf{Y} and $\boldsymbol{\zeta}$ so that each row indexes profiles rather than voters.

We repeat the EM algorithm described earlier. For each profile u , compute the probability that it belong in type k :

$$\text{for each } u, k, \text{ update: } \hat{\zeta}_{uk} \leftarrow \frac{\theta_k \prod_{j=1}^D \mu_{kj, Y_{uj}}}{\sum_{k'=1}^K \theta_{k'} \prod_{j=1}^D \mu_{k'j, Y_{uj}}} \quad (2.10)$$

Then given those type probabilities, update with

$$\text{for each } k, \text{ update: } \hat{\theta}_k \leftarrow \frac{1}{N} \sum_{u=1}^U n_u \hat{\zeta}_{uk} \quad (2.11)$$

$$\text{for each } k, j, \ell, \text{ update: } \hat{\mu}_{kj\ell} \leftarrow \frac{\sum_{u=1}^U n_u \mathbf{1}(Y_{uj} = \ell) \hat{\zeta}_{uk}}{\sum_{u=1}^U n_u \hat{\zeta}_{uk}} \quad (2.12)$$

And the log-likelihood will also only require looping through the profiles:

So the log-likelihood is

$$\mathcal{L}_{\text{obs}} = \sum_{u=1}^U \log n_u + \sum_{u=1}^U \log \left\{ \sum_{k=1}^K \theta_k \prod_{j=1}^D \mu_{kj, Y_{uj}} \right\} \quad (2.13)$$

3 Modeling Uncontested Races

A majority of elections for state and local offices are uncontested, which means that a voter technically votes in a choice but does not have the option to vote for one of the candidates. These qualitatively-different settings for another layer of modeling.

This setup uses the trichotomous outcome $\ell \in \{0, 1, 2\}$ where $Y_{ij} = 0$ indicates *abstention*, $Y_{ij} = 1$ indicates ticket *splitting* (appearing like they are voting for their less preferred party as expressed in the top of the ticket) and $Y_{ij} = 2$ indicates *straight* (co-party) voting.

Categories of uncontestedness Now we introduce a new layer: voter i for a given office j is in one of three settings, denoted by $M_{ij} \in \{1, 2, 3\}$. Unlike the cluster Z_i , that status is exactly observed in the data. Denote $M_{ij} = 3$ to mean vote j for voter i falls in the *contested* case, so the voter has all three options on the menu. Denote $M_{ij} = 2$ as the case when only the *preferred party's* candidate is in the contest, so the voter only has options $Y_{ij} \in \{0, 2\}$. Finally denote $M_{ij} = 1$ as the case when only the *opposed party's* candidate is in the contest, so the voter only has the option to abstain or “reluctantly” vote for the less favored option by splitting: $Y_i \in \{0, 1\}$.

Moving from a Multinomial model to a logit model To express the choice probability for option ℓ for office j among voters of type k , let us introduce another parameter ψ which

represents the intensity of preference for option $\ell \in \{1, 2\}$ relative to $\ell = 0$ (abstention). We set the baseline for abstention to be 0, i.e. $\psi_{kj,(\ell=0)} = 0 \forall k, j$. Then the log-likelihood can be expressed in three parts. Formally, let $V_{lk}(Y_{ij} = \ell)$ quantify the utility a voter gets for voting for option ℓ .

$$\begin{cases} \text{For } \ell = 0 : & \psi_{kj0} \stackrel{\text{set}}{=} 0 \\ \text{For } \ell \in \{1, 2\} : & \psi_{kj\ell} \equiv V_{kj}(Y_{ij} = \ell \mid Z_i = k) - \psi_{kj\ell} \end{cases}$$

Of course, how much utility translates into vote choice is an empirical question, but for our purposes these are latent quantity anyways, so we will simply assume that ψ is an intercept which exclusively shapes the category.

Parameterization To show the essence of this modeling choice, the below shows the case when there are no clusters $K = 1$ and no latent heterogeneity across individuals for simplicity. Then the indices i, k drop and we get

$$\begin{aligned} \mu_{j\ell} = & \mathbf{1}(M_j = 1) \frac{\exp \psi_{j\ell}}{\sum_{\ell' \in \{0,1\}} \exp \psi_{j\ell'}} \\ & + \mathbf{1}(M_j = 2) \frac{\exp \psi_{j\ell}}{\sum_{\ell' \in \{0,2\}} \exp \psi_{j\ell'}} \\ & + \mathbf{1}(M_j = 3) \frac{\exp \psi_{j\ell}}{\sum_{\ell' \in \{0,1,2\}} \exp \psi_{j\ell'}} \end{aligned} \quad (3.1)$$

Now we add the clustering and individual indices back in for the real data, effectively another layer to account for the fact that individuals are both of a type (Z_i) and each separate office is also of a missingness type M_{ij} :

$$\begin{aligned} \mu_{kj\ell} = & \sum_{k=1}^K \mathbf{1}(Z_i = k, M_{ij} = 1) \frac{\exp \psi_{kj\ell}}{\sum_{\ell' \in \{0,1\}} \exp \psi_{kj\ell'}} \\ & + \sum_{k=1}^K \mathbf{1}(Z_i = k, M_{ij} = 2) \frac{\exp \psi_{kj\ell}}{\sum_{\ell' \in \{0,2\}} \exp \psi_{kj\ell'}} \\ & + \sum_{k=1}^K \mathbf{1}(Z_i = k, M_{ij} = 3) \frac{\exp \psi_{kj\ell}}{\sum_{\ell' \in \{0,1,2\}} \exp \psi_{kj\ell'}} \end{aligned} \quad (3.2)$$

Analog to Multinomial Logit Because $\exp(\psi_{kj\ell}) = 1$ for $\ell = 0$, which exists in all three components, each component is analogous to a simple multinomial logit. In the first two cases, since we consider only two possibilities, it reduces to a simple intercept-only logit. Also notice

that we use the same set of parameters ψ_{kj} regardless of M_{ij} . This represents the well-known independence of irrelevant alternatives (IIA) assumption in multinomial logit. The choice probabilities when one option is not on the “menu” is assumed to follow the same type of decision rule as the ratio between the existing options. We can therefore think of equation 3.2 as representing that the relevant parameter μ for each individual takes one of the types depending on the (known) class of menu options, without introducing more parameters into the model.

EM Estimation We can use this new representation of the parameter μ in the EM algorithm. The three components in equation 3.2 can be cast as a multinomial logit. R packages of multinomial logit typically presume IIA if an outcome value is missing and implicitly do the kind of three-way subsetting as in eq. 3.2. The required data would be of the “long” form

Voter	ζ_{i1}	\cdots	ζ_{iK}	Office	M_i	Option ℓ	$Y_{ij} = \ell$
$i = 1$	0.12	\cdots	0.05	$j = 1$	3	0	FALSE
$i = 1$	0.12	\cdots	0.05	$j = 1$	3	1	FALSE
$i = 1$	0.12	\cdots	0.05	$j = 1$	3	2	TRUE
$i = 1$	0.12	\cdots	0.05	$j = 2$	2	0	FALSE
$i = 1$	0.12	\cdots	0.05	$j = 2$	2	1	NA
$i = 1$	0.12	\cdots	0.05	$j = 2$	2	2	TRUE
$i = 1$	0.12	\cdots	0.05	$j = 3$	1	0	FALSE
$i = 1$	0.12	\cdots	0.05	$j = 3$	1	1	TRUE
$i = 1$	0.12	\cdots	0.05	$j = 3$	1	2	NA
$i = 2$	0.01	\cdots	0.80	$j = 1$	3	0	FALSE
$i = 2$	0.01	\cdots	0.80	$j = 1$	3	1	FALSE
$i = 2$	0.01	\cdots	0.80	$j = 1$	3	2	TRUE

Here again, we’ve set aside the clustered nature. With that simplification, this data would allow for the simple syntax `m <- mlogit(Y ~ 1, data = votes)`.

We will estimate this regression separately for each $k \in \{1, \dots, K\}$, using the estimates of $\hat{\zeta}_{ik}$, which represents the posterior probability of voter i being in cluster k , as the weights of the logit:

$$\text{for each } k, \text{ update: } \hat{\theta}_k \leftarrow \frac{1}{N} \sum_{i=1}^N \hat{\zeta}_{ik} \quad (3.3)$$

$$\text{for each } k, j, \ell, \text{ update: } \hat{\mu}_{kj\ell} \leftarrow \text{coef}(\text{mlogit}(Y[[j]] \sim 1, \text{data}, \text{weights} = \text{zeta_k}))[[\ell]] \quad (3.4)$$

We then read off the two coefficients (relative to baseline), which will determine the value for $\mu_{kj\ell}$.