Probabilistic K-means with Multinomial Outcomes

Shiro Kuriwaki[*]

October 2019

1 Data Generating Process

(Note: This first setup basically follows the Murphy text (2017), section 11.2.2. Then it tries to implement it in stan.)

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 which can be 1 of M possibilities.

Likelihood For now, let's use $M = 2$ 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.

Let $\mu_{z[i],j} \in [0, 1]$ be the parameter that governs each y_{ij}. i.e., $\Pr(y_{ij} = 1) = \mu_{z[i],j}$. However, we won't estimate a μ for each of the N individuals; we will be estimating only K sets of length-D vectors $\boldsymbol{\mu}_k$.

Individual voters come from one of K different clusters. Individual's cluster membership is denoted $z[i]$. Therefore, we can express the joint density as follows. By referring to the set of parameters generically as $\boldsymbol{\theta}$,

$$p(\mathbf{y}_i \mid z_i = k, \boldsymbol{\theta}) = \prod_{j=1}^{D} \text{Bern}(y_{ij}|\boldsymbol{\mu}_k) = \prod_{j=1}^{D} \mu_{jk}^{y_{ij}} (1 - \mu_{jk})^{1-y_{ij}} \tag{1.1}$$

On the log scale,

$$\log p(\mathbf{y}_i \mid z_i = k, \boldsymbol{\theta}) = \sum_{j=1}^{D} (y_{ij} \log \mu_{jk} + (1 - y_{ij}) \log \mu_{jk}) \tag{1.2}$$

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

Cluster membership The random variable z_i comes from a discrete distribution. We put a discrete prior for this,

$$p(z_i) = \text{Cat}(\boldsymbol{\theta}) \tag{1.3}$$

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

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]

Cluster probabilities From this likelihood we can extract the posterior probability a point belongs to a cluster, $p(z_i = k \mid \mathbf{y}_i, \boldsymbol{\theta})$. Some call this the *responsibility* of cluster k for point i. It can be computed by Bayes rule as:

$$\begin{aligned} p(z_i = k \mid \mathbf{y}_i, \boldsymbol{\theta}) &= \frac{p(z_i = k \mid \boldsymbol{\theta}) \cdot p(\mathbf{y}_i \mid z_i = k, \boldsymbol{\theta})}{\sum_{k'=1}^{K} p(\mathbf{y}_i \mid z_i = k', \boldsymbol{\theta}) \cdot p(z_i = k' \mid \boldsymbol{\theta})} && (1.4) \\ &\propto p(z_i = k \mid \boldsymbol{\theta}) \cdot p(\mathbf{y}_i \mid z_i = k, \boldsymbol{\theta}) && (1.5) \\ &= \theta_k \cdot \prod_{j=1}^{D} \mu_{jk}^{y_{ij}} (1 - \mu_{jk})^{1 - y_{ij}} && (1.6) \end{aligned}$$

On the log scale, the normalized probability is computed as follows.

$$\begin{aligned} &\log p(z_i = k \mid \mathbf{y}_i, \boldsymbol{\theta}) \\ =&\log p(z_i = k \mid \boldsymbol{\theta}) + \log p(\mathbf{y}_i \mid z_i = k, \boldsymbol{\theta}) - \text{log\_sum\_exp}_{k'=1}^{K} \left(\log p(z_i = k' \mid \boldsymbol{\theta}) + \log p(\mathbf{y}_i \mid z_i = k', \boldsymbol{\theta})\right) \end{aligned}$$

[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 $\Sigma_k = \text{diag}(\mu_{jk}(1 - \mu_{jk}))$.

2 EM algorithm

We can estimate the key parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ by EM.[2]

The complete log-likelihood can be writen as

$$\mathbb{L}_{\text{comp}}(\boldsymbol{\mu}, \boldsymbol{\theta} | \mathbf{Y}, \mathbf{Z}) \prod_{i=}^{N} \tag{2.1}$$

$$\tag{2.2}$$

3 Stan Code

The code below tries to adapt the Stan user manual on Soft-K means and Summing out the responsibility mixture. My current attempt at the Stan code:

```
data {
  int<lower=1> K;                // number of clusters
  int<lower=1> D;                // number of offices
  int<lower=1> N;                // number of voters
  int<lower=0, upper=1> y[N, D]; // data
  vector<lower=0>[K] alpha;      // hyperparameter for cluster prevalence
}

transformed data {
  real<upper=0> neg_log_K;
  neg_log_K = -log(K);
}


parameters {
  simplex[K] pi_mixture;          // mixing proportions
  real<lower=0,upper=1> mu[K, D]; // Bernoulli probability
}

model {
  // prior
  pi_mixture ~ dirichlet(alpha);
  for (k in 1:K) {
    for (j in 1:D) {
      // something with mode at a low pr
      mu[k, j] ~ beta(2, 5);
    }
```

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

```
  }

  // likelihood
  for (n in 1:N) {
    for(j in 1:D) {
      // initiate with z ~ Cat(1/K, ... 1/K)
      vector[K] lps = rep_vector(neg_log_K, K);
      for (k in 1:K) {
        // sum all possible values of
        lps[k] += bernoulli_lpmf(y[n, j] | mu[k, j]) + log(pi_mixture[k]);
      }
      target += log_sum_exp(lps);
    }
  }



}
```

The **Appendix** includes Old Iterations and specifications I will incorporate later.

A Adding covariates

Later on, we will model θ_{jk} as a function of v_j, covariates of candidate j.

$$\theta_{jk} = \frac{\exp(v_j^\top \beta_{jk})}{\sum_{j'} \exp(v_{j'}^\top \beta_{j'k})}$$

For now, we will model three attributes of the candidate: whether the candidate is an incumbent, and whether the candidate is in an open-seat.

For example, assume we are talking about Republican voter:

		Republican Candidate		
j	Race	Name	Incumbent	Open-seat
1	HD 15	Samuel Rivers	1	0
2	HD 94	Con Chellis	0	1
3	HD 99	Nancy Mace	1	0
4	HD 110	William Cogswell	1	0
5	HD 112	Mike Sottile	1	0
6	HD 114	Lin Bennett	1	0
7	HD 115	Peter McCoy	1	0
8	HD 117	Bill Crosby	1	0
9	HD 119	Paul Sizemore	0	0

B Model 2: Kosuke

We model the outcome as coming from a categorical distribution:

$$(y_{ij} \mid Z_i = k) \sim \text{Categorical}(\theta_{k,j})$$

For example, suppose that there are three types of voters. They each have a given value of $\theta_{k,j}$:

$$\begin{cases} \text{Always straight} & \theta_{k,j} = (0.97, 0.01, 0.02) \text{ if } k = 1 \\ \text{Always split} & \theta_{k,j} = (0.01, 0.97, 0.02) \text{ if } k = 2 \\ \text{Random} & \theta_{k,j} = (0.49, 0.49, 0.02) \text{ if } k = 3 \end{cases}$$

For simplicity, let's assume this holds regardless of the office, i.e. $\theta_{k,j} = \theta_{k,j'}\ \forall j \neq j'$.

The cluster is also a categorical variable,

$$Z_i \sim \text{Categorical}(\psi),$$
$$\psi \sim \text{Dirichlet}(\alpha)$$

The length-K simplex ψ is called the **mixing proportion**. This is a low-dimensional quantity of interest.

From substantive background knowledge, our prior is that most voters are straight ticket voters, so we can set the hyperparameter to

$$\alpha = (2.0, 1.5, 1.0)$$

C Model 2: Soichiro

A simpler model may be to model the office type as a categorical variable separately. Let this variable be W. The main attribute of this variable is its level of office. Our main interest is whether some offices exhibit more split ticketting than others, so the simplest setup is to let

$$w_j \in \{0, 1\}.$$

where the values indicate

0	low propensity to generate split ticket
1	high propensity to generate split ticket

Later on, we can add a variable for candidate level incumbency.

Then, the simplex governing the parameter can be indexed as

$$(Y_{ij} | Z_i = z, W_j = w) \sim \text{Categorical}(\theta_{z,w})$$

The values of W_j for offices $j \in \{1 : J\}$ can be modeled as a categorical variable, or more simply a Bernoulli

$$W \sim \text{Bern}(\theta)$$

where $\theta \in [0, 1]$.