Supplementary Appendix for "Detecting Election Fraud from Irregularities in the Distribution of Vote-Shares"

Contents

A Precinct-Level Distributions of Vote-Shares **1**

B Population-Level Distributions of Vote-Shares **5**

C Estimating Latent Densities **7**

D Calibrating the RKD Algorithm **9**

E Software **10**

A. PRECINCT-LEVEL DISTRIBUTIONS OF VOTE-SHARES

Let $0 < N$ be the number of voters in a precinct, let $0 < T \leq N$ denote the number of voters who turn out to vote and let $0 < V \leq T$ denote the number of votes for a given candidate/party. Finally, let $R = V/T$ denote the proportion of votes for the given candidate/party and let $\mathcal{G} = \{(T, V) \mid V \leq T\}$ denote the sample space.

Proposition 1 (Jointly uniform case)**.** *Assume that* $\Pr\{(T, V) = (t, v)\} = 1/|\mathcal{G}|$ *for all* $(t, v) \in \mathcal{G}$*. For any irreducible fraction* $k/m \in (0, 1)$*,*

$$\Pr\left\{R = \frac{k}{m}\right\} = \frac{2\lfloor N/m \rfloor}{N(N+1)},$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x.

Proof. The random variable $R = k/m$, if and only if $V = ka$ and $T = ma$ for some $a \in \{1, 2, ...\}$. Since $T \leq N$, $a \leq \lfloor N/m \rfloor$. $R = k/m$ for each $(T, V) = \{k/m, 2k/2m, ..., \lfloor N/m \rfloor k/\lfloor N/m \rfloor m\}$; hence, there are $\lfloor N/m \rfloor$ pairs of $(T, V) \in \mathcal{G}$ yielding $R = k/m$. Since each pair (T, V) is equally likely, $\Pr\{R = k/m\} = \lfloor N/m \rfloor / |\mathcal{G}|$. Given that $|\{1, ..., N\} \times \{1, ..., N\}| = N^2$, we have $|\mathcal{G}| = (N^2 + N)/2$. □

Proposition 2 (Conditionally uniform case)**.** *Assume that* $\Pr\{T = t\} = 1/N$ *for all* t *and* $\Pr\{V = v | T = t\} = 1/t$*. For any irreducible fraction* $k/m \in (0, 1)$*,*

$$\Pr\left\{R = \frac{k}{m}\right\} = \frac{1}{Nm} \sum_{a=1}^{\lfloor N/m \rfloor} \frac{1}{a},$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x.

Proof. The random variable $R = k/m$, if and only if $V = ka$ and $T = ma$ for some $a \in \{1, 2, ...\}$. Since $T \leq N$, $a \leq \lfloor N/m \rfloor$; hence,

$$\Pr\left\{R = \frac{k}{m}\right\} = \sum_{a=1}^{\lfloor N/m \rfloor} \Pr\{T = ma \cap V = ka\}, \tag{1}$$

$$= \sum_{a=1}^{\lfloor N/m \rfloor} \Pr\{V = ka | T = ma\} \Pr\{T = ma\}, \tag{2}$$

$$= \sum_{a=1}^{\lfloor N/m \rfloor} \frac{1}{ma} \frac{1}{N} = \frac{1}{Nm} \sum_{a=1}^{\lfloor N/m \rfloor} \frac{1}{a}. \tag{3}$$

□

Proposition 3 (Mixture of binomials)**.** *Assume that*

$$T \sim \textit{Binomial}\ (N, p_t), \tag{4}$$

$$V | T = t \sim \textit{Binomial}\ (t, p_v). \tag{5}$$

For any irreducible fraction $k/m \in (0,1)$,

$$\Pr\left\{R = \frac{k}{m}\right\} = \sum_{a=1}^{\lfloor N/m \rfloor} \frac{N!(1-p_t)^N}{(N-ma)!(ma-ka)!ka!}\left[\frac{p_t(1-p_v)}{1-p_t}\right]^{ma}\left[\frac{p_v}{1-p_v}\right]^{ka},$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x.

Proof. The random variable $R = k/m$, if and only if $V = ka$ and $T = ma$ for some $a \in \{1, 2, ...\}$. Since $T \leq N$, $a \leq \lfloor N/m \rfloor$; hence,

$$\Pr\left\{R = \frac{k}{m}\right\} = \sum_{a=1}^{\lfloor N/m \rfloor} \Pr\{T = ma \cap V = ka\}, \tag{6}$$

$$= \sum_{a=1}^{\lfloor N/m \rfloor} \Pr\{V = ka | T = ma\} \Pr\{T = ma\}, \tag{7}$$

$$= \sum_{a=1}^{\lfloor N/m \rfloor} \binom{N}{ma} p_t^{ma}(1-p_t)^{N-ma}\binom{ma}{ka}p_v^{ka}(1-p_v)^{ma-ka}, \tag{8}$$

which simplifies to the desired equation.

□

Figure 1 shows precinct-level distributions under four different generative models. First, is the conditionally uniform model where $T \sim \text{Uniform}\{1, ..., n\}$ and $V \sim \text{Uniform}\{0, ..., t\}$. Second is the binomial model with the expected turnout equal to $.5n$ and the expected support equal to $.59t$, so that the expected vote-share of the party is 0.59. Third, the generative model where the turnout and support are drawn from the beta-binomial distributions, with the probability success for being 0.5 for turnout and 0.59 for support. Fourth, the turnout and support are drawn from the hypergeometric distributions. Both of the hypergeometric distributions are parameterized so that the expected vote-share is 0.59. In all cases number of voters in the precinct is set to 1000 ($n = 1000$). We see that in all three cases where

the expected vote-share of the party is 0.59 (figures 2-4), the largest mass in the probability distribution is at 0.6.

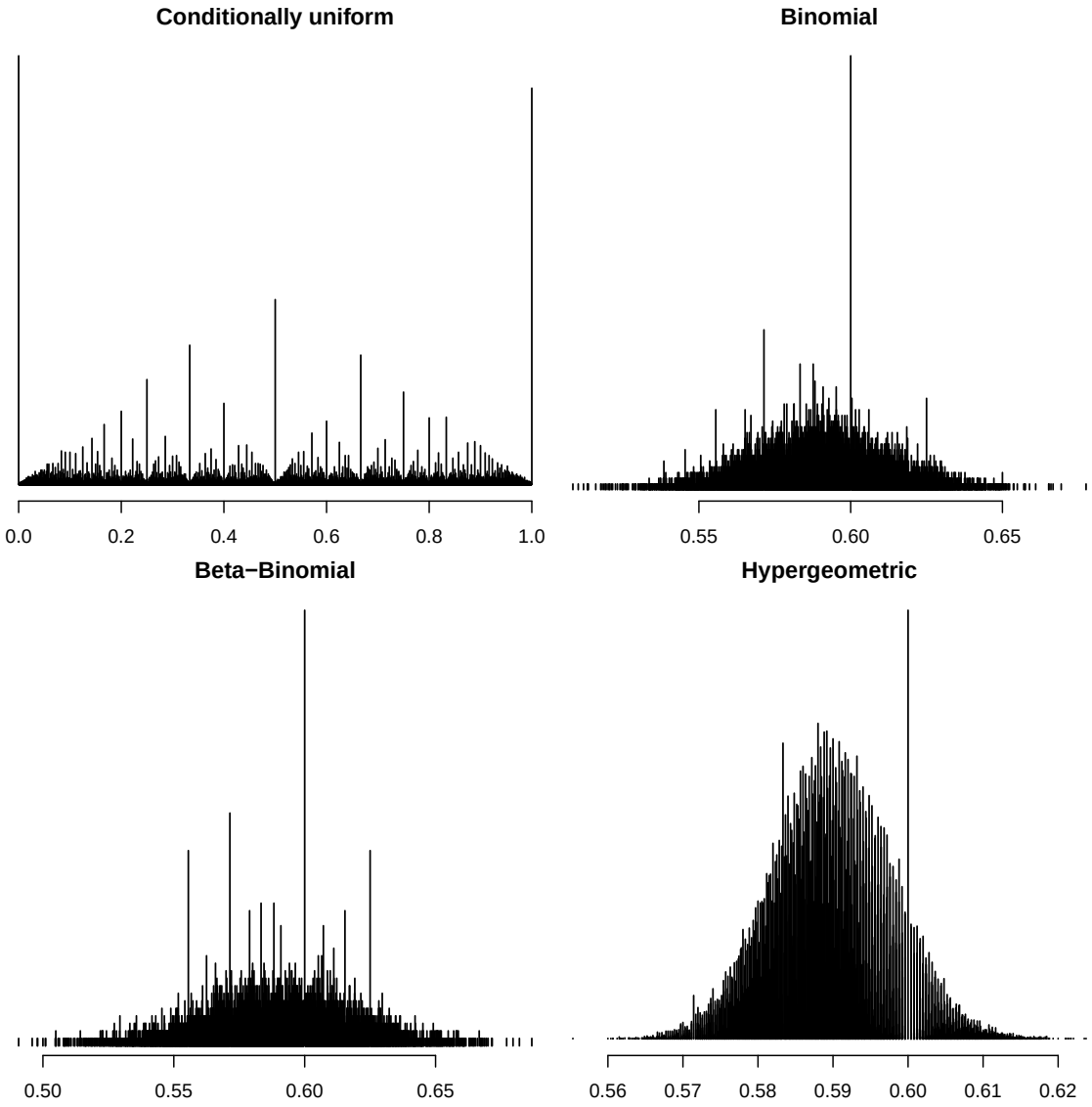

Figure 1: Precinct-level distributions of vote-shares under four different generative models.

B. POPULATION-LEVEL DISTRIBUTIONS OF VOTE-SHARES

Consider the binomial mixtures model, where we allow the turnout rates t^* and support rates v^* to vary across electoral units. For each unit $i = 1, ..., N$, we have

$$\begin{aligned} t_i &\sim \text{Binomial}(n_i, t_i^*), \\ v_i &\sim \text{Binomial}(t_i, v_i^*), \\ t_i^* &\sim P_{t^*}, \\ v_i^* &\sim P_{v^*}. \end{aligned}$$

The distribution of the irreducible fractions across the population of electoral units for the binomial case can be computed as follows:

$$\Pr\left\{R = \frac{k}{m}\right\} = \int_{v^*} \int_{t^*} \sum_{a=1}^{\lfloor n/m \rfloor} \frac{n!(1-t^*)^n}{(n-ma)!(ma-ka)!ka!} \left[\frac{t^*(1-v^*)}{1-t^*}\right]^{ma} \left[\frac{v^*}{1-v^*}\right]^{ka} dP_{t^*} dP_{v^*}$$

For given distribution P_{t^*} and P_{v^*}, the above integral can be evaluated using Monte Carlo methods. Identical computations can be performed for generative precinct-level models other than the binomial (e.g., beta-binomial, uniform).

To study the behavior of the population-level distributions of vote-shares, Figure 2 shows four such distributions for different levels of over-dispersion in the generative precinct-level distributions. The turnout and support are drawn from beta-binomial distributions with means given by t_i^* and v_i^*, respectively. In all cases, we assume that $t_i^* \sim \text{Beta}(2, 2)$ and $v_i^* \sim \text{Beta}(2, 1)$ (this roughly approximates the case where the average turnout is 50% and the average support for the party is 66%). The number of voters in each precinct is drawn from the uniform distribution on $\{500, ..., 1500\}$.

Across the four figures, I very the degree of over-dispersion in the generative distributions. When there is no over-dispersion, we have the standard binomial model. We see that the population-level distributions are spiky even without over-dispersion

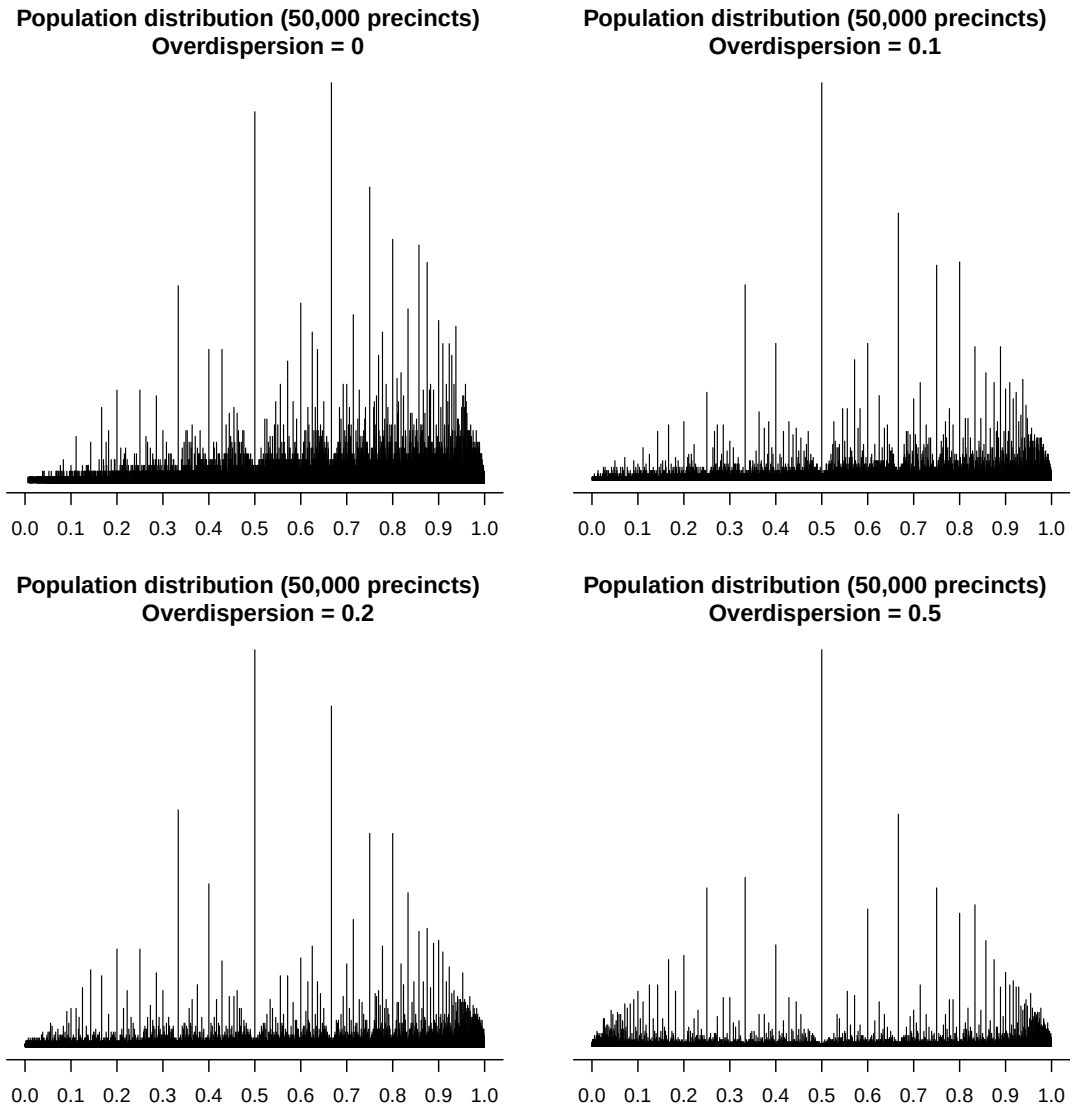

Figure 2: Population-level distributions of vote-shares, when precinct-level turnout and votes follow the binomial distribution and the binomial probabilities across the population follow beta distributions.

in the generative distributions (first figure). However, as the over-dispersion in the generative distributions increases, the mass points at low fractions become more and more prominent. Remarkably, the data were simulated so that the expected vote-

share across the precincts is 0.6, but we see the largest mass spikes at 0.5 for even mildly over-dispersed data.

C. ESTIMATING LATENT DENSITIES

The latent turnout rates $\{t_i^*\}_{i=1}^N$ and the latent support rates $\{v_i^*\}_{i=1}^N$ are drawn independently from distributions P_{t^*} and P_{v^*}, respectively. We approximate the latter continuous distributions with a finite mixture of beta distributions. The complete model for the data at hand is, for each $i = 1, ..., N$,

$$T_i|n_i \sim \text{Binomial}(n_i, t_i^*) \tag{9}$$

$$V_i|t_i \sim \text{Binomial}(t_i, v_i^*) \tag{10}$$

$$t_i^* \sim \sum_{\ell=1}^{L_t} \pi_\ell \text{Beta}\left(\boldsymbol{\theta}_\ell^{(t)}\right) \tag{11}$$

$$v_i^* \sim \sum_{\ell=1}^{L_v} \pi_\ell \text{Beta}\left(\boldsymbol{\theta}_\ell^{(v)}\right), \tag{12}$$

where $\boldsymbol{\theta}_\ell^{(t)} = (\theta_1^{(t)}, \theta_2^{(t)})_\ell$ represents the two shape parameters of the beta distribution of the turnout rates for the mixture component ℓ. Conditionally on the observation i belonging to the mixture component ℓ, we can specify the marginal distributions and turnout (which do not belong on the unknown parameters t_i^* and v_i^*), for all $\ell = 1, ..., L$,

$$T_i|n_i, i \in \ell \sim \text{Beta-Binomial}\left(n_i; \boldsymbol{\theta}_\ell^{(t)}\right) \tag{13}$$

$$V_i|t_i, i \in \ell \sim \text{Beta-Binomial}\left(t_i; \boldsymbol{\theta}_\ell^{(v)}\right). \tag{14}$$

The stochastic process in the above model allows the population of precincts to be heterogeneous in terms of the distributions from which their latent turnout and

support rates are drawn. Note that although the parameters $\{t_i^*\}_{i=1}^N$ and $\{v_i^*\}_{i=1}^N$ can be estimated, we do not need them as all the necessary information about the population distribution is contained in the hyper-parameters $\boldsymbol{\theta}_\ell^{(t)}$ for $\ell = 1, ..., L_t$ and $\boldsymbol{\theta}_\ell^{(v)}$ for $\ell = 1, ..., L_v$.

The population hyper-parameters can be estimated relatively straightforwardly using the Expectation-Maximization (EM) algorithm (Casella and Berger, 2002, Ch. 7). However, the initial values for the EM algorithm have to be chosen judiciously. The following initialization routine works very well. Since the target of the estimation is the latent distributions of turnout and support rates, a good starting point is to select the starting values for the negative binomial parameters $\boldsymbol{\theta}$ and the component weights $\pi_1, ..., \pi_L$, which minimize the square distance between the kernel density estimate of the latent distribution (which will not be smooth) and the smooth density approximated via the mixtures of beta distributions model above. Formally, let $P_L(x; \boldsymbol{\theta}, \boldsymbol{\pi})$ denote the density parameterized by the beta-mixture model with L components and evaluated at point x. The initial values are found by solving

$$\min_{\boldsymbol{\theta}, \boldsymbol{\pi}} \frac{1}{K} \sum_{i=1}^{K} \left(\hat{f}_h(x_i) - P_L(x_i; \boldsymbol{\theta}, \boldsymbol{\pi}) \right)^2,$$

subject to $\sum_{\ell=1}^{L} \pi_\ell = 1$.

For a given L, the EM algorithm is iterated from these initial values until the correlation between the densities computed at two successive iterations is above 0.99. Given the initialization procedure above, the EM algorithm typically converges in as few as fives steps. The number of components L is chosen via the following procedure: estimate the density functions for $L = 1, 2, ...$ and stop when the Pearson correlation between the density $P_{L+1}(x_i; \boldsymbol{\theta}, \boldsymbol{\pi})$ estimated at points $\{x_1, ..., x_K\}$ and $P_L(x_i; \boldsymbol{\theta}, \boldsymbol{\pi})$ is at least 0.99. Intuitively, this simply means that we stop increasing the complexity of the underlying model when they yield highly similar density functions. For the data analyzed here, most of the time the optimal L is between 1 and 3.

D. CALIBRATING THE RKD ALGORITHM

When using the RKD algorithm, three parameters have to be preset – the number of resamples (M), the size of the grid on which the kernel density is estimated (K), and the bandwidth for kernel density estimation (h). The choice of these parameters involves certain types of trade-offs.

When choosing the number of resamples M, there is a trade-off between how 'conservative' or 'liberal' the resulting estimates will be: a too small M will result in too small estimates of fraud, whereas too large M might result in too small estimates of fraud. Choosing the size of the grid results in a different type of trade-off between the computational speed and precision of the resulting estimates. As the size of the grid K increases, the estimates become more precise, but this also means that on each iteration one has to estimate the density at many more points (and store the estimates in memory to calculate the upper envelope across all samples). Finally, choosing the bandwidth h is more straightforward: when the bandwidth is too large, the density estimates will not have spikes and one will not be able to detect fraud. Generally, at least for the kernel density estimation used here, reducing the size of h does not cause an increase in the computation cost.

To calibrate parameters of the RKD algorithm, I used simulated data and varied the three RKD parameters to see which combination yields accurate estimates while retaining computational efficiency. The calibration exercise is conducted on two synthetic datasets: the true amount of fraud is 1% and 3%, respectively.

The results are given in Table 1. We see that as long as the bandwidth parameter is not too large (less than or equal to 0.001), we see very small differences between the estimates. In all cases, the estimates are quote close to their true value and vary only within few decimal points. Based on these results, I set $M = 1000$, $K = 1001$ and $h = 0.001$ in the analyses of the paper.

resamples	V2	V3	V4	V5
True level of fraud is 1 %				
500	501	0.01	0.30	(0.07, 0.37)
500	501	0.001	0.88	(0.72, 0.95)
1000	1001	0.001	0.98	(0.91, 1.02)
2000	1001	1e-04	1.01	(0.95, 1.05)
1000	1001	1e-04	1.02	(0.94, 1.06)
1000	2001	1e-04	1.05	(1.01, 1.08)
True level of fraud is 3 %				
500	501	0.01	1.99	(1.52, 2.43)
500	501	0.001	2.93	(2.91, 2.96)
1000	1001	0.001	3.02	(3.00, 3.05)
2000	1001	1e-04	2.95	(2.93, 2.99)
1000	1001	1e-04	2.97	(2.94, 3.05)
1000	2001	1e-04	2.99	(2.98, 3.02)

Table 1: RKD estimates of fraud with 95 percent credible intervals for different estimation parameters.

E. SOFTWARE

The package `spikes` in R software (R Core Team, 2016) implements the RKD algorithm. The main function in the package `spiketest` takes a data frame of election results and returns the estimated level of fraud:

$$\texttt{spikes}(\texttt{data}, \texttt{resamples} = 1000, \texttt{bw} = 0.0001, \texttt{grid} = 1001)$$

The data frame `data` must consist of three columns named as follows: N (representing the number of voters), t (the number of people who turned out to vote), and v (the number of people who supported the party whose vote-shares are being analyzed. The function `spiketest` is a wrapper function, implementing the two steps of the algorithm: it estimates the densities of latent turnout and support rates and then resamples data from the estimated densities and compares them with the density of

the observed data.

The package also contains a function

$$\texttt{confInt}(\texttt{out}, \texttt{boots} = 100),$$

which estimates the credible intervals for the fraud estimate. It uses the Bayesian approach (with flat priors) and samples the densities of turnout and support from the poster and then executes the resampling procedure conditional on the draw from the posterior. The parameter `boots` specifies the number of bootstraps from which the credible intervals are estimated. Note that since each bootstrap requires to resample the data M times, setting `boots` to a large number is computationally extremely costly. The default value of 100 boots yields a fairly good estimate of the credible interval, which does not change by much for a larger number of samples from the posterior density. See description of the package on CRAN for further details.

REFERENCES

Casella, George and Roger S. Berger. 2002. *Statistical Inference*. Duxbury: Thompson Learning.

R Core Team. 2016. "R: A Language and Environment for Statistical Computing." *https://www.R-project.org/* .