# Notes for Statistics Lab 52568 - 2020/21

Or Zuk

## 1 Israel Election Dataset - Analysis and Methods

### 1.1 Introduction

We describe here the dataset for elections for the 23nd Knesset in Israel (March 2020), with votes per city. (**Remark:** by city here we mean any 'yeshuv' which can be a city, village, kibutz etc.).

We also describe the data analysis and tools used to answer different questions about the data.

#### 1.1.1 Notations:

- Our dataset is a matrix $\boldsymbol{N} \in \mathbb{R}_{\boldsymbol{C} \times \boldsymbol{K}}$ where $K$ is the number of parties and $C$ is the number of cities. $n_{ij}$ is the number of voters for party $j$ in city $i$. In addition, we have the following:
- Let $n_{i\bullet} = \sum_{j=1}^{K} n_{ij}$ be the total number of legal votes ('kolot ksherim') in city $i$. $\tilde{n}_{i\bullet}$ is the total number of eligible voters in city $i$ ('baalei zhut bhira'). From these, we can calculate the voting turnout at city $i$: $v_i = \frac{n_{i\bullet}}{\tilde{n}_{i\bullet}}$. (In the data file you are given both $\tilde{n}_{i\bullet}$ and the voting turnout, but you need to re-calculate $v_i$ without 'kolot psulim').
- Similarly the number of total votes for party $j$ is $n_{\bullet j} = \sum_{i=1}^{C} n_{ij}$ .
- Let $n = \sum_{i=1}^{C} n_{i\bullet} = \sum_{i=1}^{C} \sum_{j=1}^{K} n_{ij}$ be the total votes across all cities ('kolot ksherim'). Similarly, let $\tilde{n} = \sum_{i=1}^{C} \tilde{n}_{i\bullet}$ be the total number of eligible votes in Israel ('baalei zhut bhira').
- We also define $\tilde{n}_{ij}$ - the total number of supporters for party $j$ in city $i$ - that is, how many individual would have voted for party $j$ if all eligible voters in city $i$ were forced to vote. In contrast to the previous quantities, this quantity is not observed and cannot be computed directly from the data.
- Let also $z_i$ is the number of bad votes 'psulim' in city $i$.

## 1.2 Computing Parties Vote Share

1. The fraction of votes for party $j$ in the elections is $p_j \equiv \frac{n_{\bullet j}}{n}$. The vector $p = (p_1, ..., p_K)$ represents the share of votes for each party, such that $s_i$, the number of seats in the parliament for each party, is approximately $s_i \approx 120 p_i$ (the exact relationship is much more complicated, and includes rounding, thresholding small parties due to 'ahuz hasima', the Badder-Offer law etc.).

2. Since the elections are meant to represent the opinions of all citizens in the country, voting turnout may be an issue as it can distort the actual preferences of the citizens - that is, if the turnout for the potential voters of party $i$ is much larger than the turnout of the potential voters of party $v_j$, then the share of votes $p_i$ may be much higher for the first party compared to $p_j$ for the second, even if in the general population the situation is reversed.

3. A natural question which we would like to answer is: can we infer from the elections results the actual preferences in the population? A followup question is: if every citizen in the country would have voted, would we see a significantly different result in the elections?

4. Denote by $\tilde{n}_{\bullet j}$ the (unknown) total number of votes for party $j$ if every citizen actually voted. Similarly, denote by $q_j$ the (unknown) share of votes for the party in this situation, given by $q_j \equiv \frac{\tilde{n}_{\bullet j}}{n}$.

5. Our goal will be to estimate the $q_j$ values from the election results.

## 1.3 A Statistical Model for Voting

We assume that each person in Israel decides in advance which party he/she prefers. Then, on election day, people from city $i$ who prefer party $j$ vote

with probability $v_{ij}$. Therefore, the number of actual voters for party $j$ in village $i$ is $n_{ij} \sim Binom(\tilde{n}_{ij}, v_{ij})$. Both $\tilde{n}_{ij}$ and $v_{ij}$ are unknown parameters, and we will try to estimate them from the data in order to make a correction for the total number of votes. The problem is that the number of unknown parameters is $\approx 2K \times C$ which is on the order of the data size, and we have no hope of estimating the parameters reliably. Therefore, we need to make additional assumptions in order to estimate parameters.

**Exercise 1.** For each of the following quantities, determine if it is (i) observale in the data, (ii) can be computed from the observable data, or (iii) unobservable (and can only be estimated):

$n_{ij}, \tilde{n}_{ij}, n_{i\bullet}, \tilde{n}_{i\bullet}, n_{\bullet j}, \tilde{n}_{\bullet j}, p_j, q_j$

## 1.4 Simulation Study

Our goal is to estimate the unknown $q_j$ values (proportion of potential voters for party $j$ in the population) from the observed $p_j$ values (parties' proportion

in the election). For the real data, we don't know how good will our estimates be.

However, we can make different assumptions on the voting probabilities of individuals, and evaluate the performance of different corrections under these assumptions in a simulation study. The high-level description for a simulation study is as follows:

1. Choose values for the real numbers of voters $\tilde{n}_{ij}$ and voting probabilities $v_{ij}$ , and compute the parties proportions $q_j$ from the $\tilde{n}_{ij}$ values.

2. Simulate (many times) the observed number of voters in the election $n_{ij}$ using $n_{ij} \sim Binom(\tilde{n}_{ij}, v_{ij})$

3. Apply a correction (see next section) to get estimators $\hat{\tilde{n}}_{ij}$ and sub-sequentially estimators $\hat{q}_j$ for the population proportions

4. Compare the true values $q_j$ to the estimated values $\hat{q}_j$ : Compute the empirical bias, variance and mean-squared error of the estimators $\hat{q}_j$.

Will use **parameter tying** - that is, assume that the value of different parameters is the same. We can suggest the following options:

1. Constant voting turnout per city: $v_{ij} = v_i$. ($C$ parameters in total).
2. Constant voting turnout per party: $v_{ij} = u_j$. ($K$ parameters in total).
3. An additive/multiplicative model: $v_{ij} = u_j + v_i$ or $v_{ij} = u_j v_i$ . This model will have $K + C$ parameters, still far less than the data size ($K \times C$).

Note: in all of these models we do parameter tying for the $v_{ij}$ parameters. We don't consider here what to do with the $\tilde{n}_{ij}$ parameters. This will be clearer when we estimate the parameters.

## 1.5 Estimating total votes

We propose here different estimators for the votes distribution if everybody voted. The estimators differ in their assumptions, computation and statistical properties.

1. We can first do the following simple correction: if in city $i$ the voting turnout was $v_i$, this means that every vote actually counted in this city represents not one but $v_i^{-1}$ votes from the population of the city. We can thus give weights to the votes in each city. We get the following estimator for the votes in a city: First, compute the $v_i$ values:

$$v_i = \frac{n_{i\bullet}}{\tilde{n}_{i\bullet}} \tag{1}$$

Then, we can use the $v_i$ values to compute the correction:

$$\hat{\tilde{n}}_{ij} = \frac{n_{ij}}{v_i} \tag{2}$$

$$\hat{\tilde{n}}_{\bullet j} = \sum_{i=1}^{C} \hat{\tilde{n}}_{ij} = \sum_{i=1}^{C} n_{ij} v_i^{-1} \tag{3}$$

$$\hat{q}_j = \frac{\hat{\tilde{n}}_{\bullet j}}{\sum_{k=1}^{K} \tilde{n}_{\bullet k}} = \frac{\hat{\tilde{n}}_{\bullet j}}{\tilde{n}} = \frac{\sum_{i=1}^{C} n_{ij} v_i^{-1}}{\sum_{i=1}^{C} \sum_{k=1}^{K} n_{ik} v_i^{-1}}. \tag{4}$$

This estimator adjusts the voting in each city according to the voting turnout. We used this estimator in class, and the results were shown in lab 2.

A main problem with this adjustment is that it assumes that all voters in a city are equally likely to vote. But what if the voters of a certain party are more/less likely to vote?

**Exercise 2.** Consider an election with two ballots and three parties. Let the results be given in the following table:

| Party: | Greens | Reds | Blues | bzb |
|---|---|---|---|---|
| Ballot1 | 100 | 200 | 300 | 800 |
| Ballot2 | 50 | 250 | 200 | 1000 |

Compute $p_j$ and $\hat{q}_j$ using eq. (4) for all three parties.

**Exercise 3.** Let $p_j, q_j$ and $\hat{q}_j$ be the elections frequency, the frequency in the population, and the corrected frequency for party $j$. Give numerical examples for election results with the following three scenarios: $p_j < \hat{q}_j < q_j$, $p_j < q_j < \hat{q}_j$ and $q_j < p_j < \hat{q}_j$.

2. To develop the next estimator, we will assume that the voter turnout for each **party** is a **constant** (rather than for each **city**). Let $u_j$ be the voter turnout for party $j$. Then we have:

$$\tilde{n}_{\bullet j} = n_{\bullet j} u_j^{-1} \tag{5}$$

and therefore, if we knew the $u_j$ values, we could use the estimator:

$$\hat{q}_j = \frac{n_{\bullet j} u_j^{-1}}{\sum_{k=1}^{K} n_{\bullet k} u_k^{-1}} \tag{6}$$

We next need to estimate the $u_j$s from the data. After we do so, we can just plug in the estimators $\hat{u}_j^{-1}$ into the above equation to get:

$$\hat{q}_j = \frac{n_{\bullet j} \hat{u}_j^{-1}}{\sum_{k=1}^{K} n_{\bullet k} \hat{u}_k^{-1}}. \tag{7}$$

How would we estimate the parties voting turnout? The problem is that we cannot repeat the computation we did before for $v_i$. If we take Equation (1), the analogous equation for $u_j$ would be:

$$u_i = \frac{n_{\bullet j}}{\tilde{n}_{\bullet j}} \tag{8}$$

But, in contrast to the observable $\tilde{n}_{i\bullet}$ (total number of eligible voters in city $i$), we don't know $\tilde{n}_{\bullet j}$ (total potential number of voters for party $j$). Instead, we will develop a different method for estimating the $u_j$ values and for computing the correction, described next.

The idea is conceptually simple: if in cities where a party is strong we see higher voting turnouts, then the voting turnout for the voters of this parties is high (and the same for lower turnouts indicating a lower turnout for the party). To translate this idea into mathematical formulation, we would like the cities voting turnout $v_i$ to be explained by the parties voting turnouts $u_j$. That is:

$$v_i \approx \frac{n_{i\bullet}}{\sum_{j=1}^{K} n_{ij} u_j^{-1}}. \tag{9}$$

Since $v_i = \frac{n_{i\bullet}}{\tilde{n}_{i\bullet}}$, we can require $\tilde{n}_{i\bullet} \approx \sum_{j=1}^{K} n_{ij} u_j^{-1}$ for each city $i$. Summing over the cities, we can formulate a least-squares problem:

$$(\hat{u}_1^{-1}, ..., \hat{u}_K^{-1}) = argmin_{u_1^{-1},...,u_K^{-1}} \sum_{i=1}^{n} (\sum_{j=1}^{K} n_{ij} u_j^{-1} - \tilde{n}_{i\bullet})^2 \tag{10}$$

That is, the inverse turnout parameters $u_j^{-1}$ can be obtained as the least squares solution of a linear regression problem with design matrix $N$ and outcome vector $y = \tilde{n_{|\bullet}}$ where $\tilde{n_{|\bullet}} = (\tilde{n_{1\bullet}}, ..., \tilde{n_{C\bullet}})$ . The least-squares solution is therefore:

$$\hat{u}^{-1} = [N^T N]^{-1} N^T \tilde{n_{|\bullet}} \tag{11}$$

and our estimator for $q$ is

$$\hat{q}_j = \frac{\sum_{i=1}^{C} n_{ij} \hat{u}_j^{-1}}{\sum_{i=1}^{C} \sum_{j'=1}^{K} n_{ij'} \hat{u}_{j'}^{-1}}. \tag{12}$$

We can write the estimator in a vector form using matrix and vector operations as follows:

$$\hat{q} = \frac{\mathbf{1}_C^T N diag([N^T N]^{-1} N^T \tilde{n_{|\bullet}})}{\left\| \mathbf{1}_C^T N diag([N^T N]^{-1} N^T \tilde{n_{|\bullet}}) \right\|_1}. \tag{13}$$

where $\mathbf{1}_C^T$ is a constant row vector of ones of length $C$, $diag(v)$ for a vector $v$ is a diagonal square matrix with the values of $v$ on the diagonal, and

$|| \cdot ||_1$ is the $L_1$-norm: $||x||_1 = \sum_i |x_i|$ . The estimator we get here is different from the one in eq. (4).

**Question:** How does it perform with respect to the actual election results compared to the previous estimator? we can run it on the real data and look if the results make sense.

**Question:** Do you see any problems with this estimator? what are the issues that are problematic and/or can be improved?

Alternatively, we may not want to fit the actual number of voters in each city, but rather the proportion how voted in each city

$$(\hat{u}_1^{-1}, ..., \hat{u}_K^{-1}) = argmin_{u_1^{-1},..,u_K^{-1}} \sum_{i=1}^{C} (\sum_{j=1}^{K} \frac{n_{ij}}{n_{i\bullet}} u_j^{-1} - \frac{\tilde{n}_{i\bullet}}{n_{i\bullet}})^2 \tag{14}$$

3. Ideally, we would like to know the parameters $v_{ij}$ representing the voting turnout for the voters of party $j$ at city $i$. Then, if we have good estimators $\hat{v}_{ij}^{-1}$we can simply use the estimated votes $\hat{\tilde{n}}_{ij} = n_{ij}\hat{v}_{ij}^{-1}$ to estimate $q$. The problem is that there are too many such parameters $(K \times C)$, which are equal to the number of observation. In fact, any division of the votes for people who didn't vote between the parties will give a valid $v_{ij}$- for example, assuming that all the voters who didn't vote in the entire country would have voted for the Pirates party.

   Our first estimator essentially assumed that turnout is constant across parties, that is: $v_{ij} = v_i$. The next estimator assumed that turnout is constant across cities (for the same part), that is: $v_{ij} = u_j$ - that is, the turnout for a party varies between cities but in the same way for all parties. We can offer a richer model which combines the two. Let $u$ be the parties turnout vector and let $v$ be a cities turnout vector. We can assume for example an additive model $v_{ij} = v_i + u_j$ or a multiplicative model $v_{ij} = v_i u_j$ . This gives a model with $K + C$ parameters which we can try to estimate with the $C \times K$ observations. However, since the linear regression form in eq. (10) is in terms of the $u_j^{-1}$(which stand for the $v_{ij}^{-1}$), it is mathematically more convenient to use the parameterization: $v_{ij}^{-1} = v_i^{-1} + u_j^{-1}$,or $v_{ij} = \frac{1}{1/v_i + 1/u_j}$ i.e. $v_{ij}$is half the harmonic mean of $v_i$ and $u_j$.

   We can again use least-squares for estimation.

$$\tilde{n_{i\bullet}} \approx \sum_{j=1}^{K} n_{ij}(v_i^{-1} + u_j^{-1}) \tag{15}$$

   and we get the following least-squares optimization problem:

$$(\hat{u}^{-1}, \hat{v}^{-1}) = argmin_{u^{-1},v^{-1}} \sum_{i=1}^{C} (\sum_{j=1}^{K} n_{ij}(v_i^{-1} + u_j^{-1}) - \tilde{n_{i\bullet}})^2 \tag{16}$$

While this is a convenient linear regression problem, there is a problem here: we have more unknowns $(K + C)$ than equations $(C)$. To overcome this problem, we can use regularization (e.g. Ridge regression), or formulate a different optimization problem and different estimators. It is not clear how best to solve this problem.

## 1.6 Principal Component Analysis

Principal Component Analysis (PCA) is a method for data transformation and dimensionality reduction. We will use this method for visualization of parties and ballots in the elections.

Suppose that we have data points $x_1, .., x_n \in \mathbb{R}^{\mathrm{p}}$ where typically the dimension $p$ is large. We would like to represent the points in a compact manner, where we may allow ourselves to lose some information (precision) about the points.

For example, we may describe the points using one vector, namely their average: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \in \mathbb{R}^{\mathrm{p}}$. The average is a summary of the data points using one point, which can be thought of as a zero-dimensional subspace. We can similarly summarize the points using a one-dimensional space (a line), a two-dimensional space (a plane) and so on. In general, we would like to represent the points using a linear subspace $B \subset \mathbb{R}^{\mathrm{p}}$ with $dim(B) = k \ll p$. The projection of each vector $x_i$ onto $B$ is denoted $T_B(x_i)$,

$$T_B(x) = \arg\min_{x' \in B} ||x - x'||_2^2 \tag{17}$$

*Remark* 4. In the above we assume that $B$ is a *linear* sub-space of dimension $k$, that is, there is a basis $b_1, ..., b_k \in \mathbb{R}^p$ such that $B = \{\sum_{j=1}^{k} \alpha_j b_j \ : \ \alpha_1, ..., \alpha_k \in \mathbb{R}\}$. In particular, setting all $\alpha_j$'s to zero we get $0 \in B$ as indeed every linear subspace contains the origin. However, our data points may be far away from the origin, hence we may want to represent them using an *affine* sub-space of dimension $k$, defined as: $B = \{b_0 + \sum_{j=1}^{k} \alpha_j b_j \ : \ \alpha_1, ..., \alpha_k \in \mathbb{R}\}$. Here $b_0$ is the offset of the subspace, and we need $k+1$ vectors to describe an affine sub-space of dimension $k$. To simplify the discussion, in PCA it is customary to subtract the mean from all vectors as a pre-processing step, and then perform PCA on a set of vectors with mean zero, and represent this set using a linear subspace. That is, we will find the linear subspace $B$ best describing the shifted vectors: $x_1 - \bar{x}, ..., x_n - \bar{x}$. From now on, we will assume that the vectors $x_1, ..., x_n$ are given after the mean was already subtracted.

In $PCA$, our goal is to find the subspace $B$ minimizing the distance to all points, i.e. the distance between points and their projection:

$$B^* = argmin_{B \subset \mathbb{R}^{\mathrm{p}}, dim(B)=k} \sum_{i=1}^{n} ||x_i - T_B(x_i)||^2 \tag{18}$$

The solution to the above problem is given using the Singular Value Decomposition (SVD) of the matrix $X$, and we don't present here the details.

### 1.6.1 Changing the dimension $k$

We can perform PCA and solve the optimization problem (18) for multiple values of $k$. It turns out that there is a relationship between the optimal sub-spaces $B^*$ for different values of $k$ and they are *nested:*

**Proposition 5.** *Let $B^{*(1)}, B^{*(2)}, ..., B^{*(p)}$ be the PCA solutions to problem (18) for $k = 1, .., p$ respectively. Then, we have:*

$$B^{*(1)} \subset B^{*(2)} \subset .. \subset B^{*(p)} \subset \mathbb{R}^p. \tag{19}$$

*Remark* 6. When the dimension $p$ is larger than the number of points $n$, there is no point in performing PCA for $k \geq n$. The reason is that $n$ points can always fit perfectly a sub-space of dimension $n-1$. For example, we can always find a line passing through two points, a plane passing through 3 points etc. Therefore, we will be interested in doing PCA only for dimensions $k \leq min(p, n)$. When $n < p$ we will have $B^{*(n)} = B^{*(n+1)} = ... = B^{*(p)}$.

**Corollary 7.** *We can find the optimal PCA subspace in steps. First, we find the one-dimensional subspace $B^{*(1)}$. We can represent this sub-space using a basis comprised of a single vector $b_1$ such that $B^{*(1)} = span(b_1)$. Next, to find $B^{*(2)}$ we need only to find a vector $b_2$ independent of $b_1$ such that $B^{*(2)} = span(b_1, b_2)$. We keep adding vectors to the basis in the same manner, where for dimension $k$ we add a vector $b_k$ independent on $b_1, ..., b_{k-1}$ and have $B^{*(k)} = span(b_1, ..., b_k)$.*

*Remark* 8. In fact, in the above sequential process it is possible to find vectors $b_1, ..., b_k$ that are not only independent, but actually orthogonal. These vectors are then called the Principal Components. They are unique up to normalization (i.e. multiplying each of them by a scalar).

### 1.6.2 Variance reduction

We can define the variance for a set of vectors in $\mathbb{R}^p$:

$$Var(x_1, ..., x_n) = \frac{1}{n} \sum_{i=1}^{n} ||x_i - \bar{x}||^2 \tag{20}$$

where we take here the Euclidean norm of vectors $||v||^2 = \sum_i v_i^2$.

We can also define the variance after projection:

$$Var(T_B(x_1), ..., T_B(x_n)) = \frac{1}{n} \sum_{i=1}^{n} ||T_B(x_i) - T_B(\bar{x})||^2. \tag{21}$$

The projection always reduces (in the weak sense) the variance, that is, for every $k$ and every subspace $B$ we have: $Var(T_B(x_1), ..., T_B(x_n)) \leq Var(x_1, ..., x_n)$.

Moreover, PCA has the following property:

**Proposition 9.** *$B^*$ is the subspace maximizing the variance $Var(T_B(x_1), ..., T_B(x_n))$ over all sub-spaces $B$ of dimension $k$.*

Let $B^{*(k)}$ be the PCA sub-subspace of dimension $k$. Denote $w_k = Var(T_{B^{*(k)}}(x_1), ..., T_{B^{*(k)}}(x_n))$. From the above discussion we conclude that $0 = w_0 \leq w_1 \leq ... \leq w_p = Var(x_1, ..., x_n)$. Define $v_k = w_k - w_{k-1}\ \forall k = 1, .., p$ to be the additional variance obtained when increasing the dimension from $k-1$ to $k$. We then have a variance decomposition: $Var(x_1, ..., x_n) = \sum_{i=1}^{p} v_i$. Because the principal components are orthogonal, $v_i$ is also the variance obtained by projecting the $x_i's$ on $b_i$, i.e. $v_i = Var(T_{b_i}(x_1), ..., T_{b_i}(x_n))$. Moreover, we have the following property:

**Proposition 10.** *The variance obtained by projecting on the principal components are non-increasing, i.e.:* $v_1 \geq v_2 \geq v_3 \geq ... \geq v_p$.

We can also define the relative proportion of the variance: $r_k^2 = \frac{w_k}{w_p}$, a number in $[0, 1]$ describing the proportion of variance of the original data-points captured by their projection on the first $k$ principal components. This value can be interpreted as a goodness of fit measure, in the same manner that $r^2$ is interpreted in linear regression. We can use this value to choose the dimension $k$ to project to in PCA. We would like to choose $k$ as small as possible, but on the other hand having high $r^2$. If we reach a certain $k$ such that the additional variances $v_{k+1}, v_{k+2}, ..$ are small, we can decide to use only $k$ principal components and ignore the rest.

**Exercise 11.** Suppose that $B$ is the PCA sub-space of dimension $k < n, p$ for vectors $x_1, ..., x_n$ having mean zero. Let $T_B$ be the orthogonal projection on $B$ and $T_{B^\perp}$ be the projection on the orthogonal subspace $B^\perp$. Prove that:

$$Var(x_1, ..., x_n) = Var(T_B(x_1), ..., T_B(x_n)) + Var(T_{B^\perp}(x_1), ..., T_{B^\perp}(x_n)). \quad (22)$$

### 1.6.3 Achieving dimensionality reduction with PCA

$T_B$ is a transformation from $\mathbb{R}^{\mathrm{p}}$ to itself, hence it doesn't yet achieve the dimensionality reduction goal. The size of the data matrix containing $T_B(x_1), ..., T_B(x_n)$ is $n \times p$, just as the original data matrix.

However, points in the new subspace $B$ can be represented using only $k$ internal coordinates, instead of $p$, using a basis $b_1, .., b_k$ spanning $B$ (In PCA we find an orthogonal basis.)

Let $T_B(x) = \sum_{i=1}^{k} \alpha_i b_i$. Then, the vector $(\alpha_1, .., \alpha_k)$ represents $x$ in the subspace $B$, and we can define the transformation: $S_B : \mathbb{R}^p \to \mathbb{R}^k$ defined by $S_B(x) = \alpha$.

Thus, in PCA we have three important objects:

1. The Principal components for the sub-space $B^*$, listed as the columns $b_1, .., b_k$ forming an orthonormal basis for $B^*$, and can be listed as columns of a matrix $\boldsymbol{B} \in \mathbb{R}_{\mathrm{n}\times\mathrm{k}}$, i.e. $\boldsymbol{B} = [b_1|b_2|...|b_k]$.

2. The transformation $T_B : \mathbb{R}^{\mathrm{p}} \to \mathbb{R}^{\mathrm{p}}$ projecting each data points $x_i$ onto the subspace $B$.

3. The dimensionality-reduction transformation $S_B : \mathbb{R}^{\mathrm{p}} \to \mathbb{R}^{\mathrm{k}}$, giving the internal coordinates of the transformation $T_B(x_i)$ in the subspace $B$.

All three objects are useful, and it is important not to confuse between them. The projected data $T_B(x_1), ..., T_B(x_n)$ serves as a denoised version of the original data $x_1, ..., x_n$, and the squared distances $(x_i - T_B(x_i))^2$ represent the goodness of fit of the PCA plane. The reduced data $S_B(x_1), ..., S_B(x_n)$ is often used for analysis allowing us reduced computation time compared to the original data (since we get a data matrix of size $n \times k$ instead of $n \times p$), and often also cleaning of noise in the data. Is is also used for *data visualization* , where we often take $k = 2$ or 3, and display the data points in the plane (or three-dimensional space) spanned by the first 2 or 3 principal components, i.e. the transformations $S_B(x_1), .., S_B(x_n)$ are shown. The principal components themselves $b_1, ..b_k$ are used to infer the directions in which the data has the highest variance. They often can be interpreted in terms of the domain from which the data points came. For example, the top principal components can correspond to geographic locations, to political orientation (e.g. left/right, religious/secular) for elections data etc. The coordinates of the principal components represent the importance of the original variances. They are called *the loadings*. That is, $b_{ij}$ represent how important is the original variable $j$ (out of the $p$ variables) in the $i$-th principal components.

## 1.7 Sampling: Using polls (midgam) to predict the election's outcome

Here, we show how to obtain election results from sampling. The goal is to use a subset of the votes and calculate based on this subset an estimator $\hat{p_j}$for the true votes proportion $p_j \equiv \frac{n_{\bullet j}}{n}$ for each party $j$. There are many sampling methods and we will study a few simple ones. Ideally, we would have liked to sample $m$ random individuals out of the $n$ total votes, where the assumption is that we sample individuals with uniform probability and without replacement. That is, a sample of $m$ out of $n$ individuals is in our poll (midgam). Such sampling is what pollsters try to achieve when they conduct polls, e.g. when using online or telephone polls, although in practice there may be many biases to such polling strategies (e.g. some individuals may not be available / not answer the calls, or lie when asked to which party they vote). Nevertheless, we provide a simple mathematical analysis for such polling, assuming that random uniform sampling is performed.

Let $X_{ij}$ be a Bernoulli random variable set to 1 if individual $i$ voted for party $j$ in the poll, and 0 otherwise. Let $I \subset 1, .., n$ with $|I| = m$ be the random set of individuals belonging to the poll. Then, the simplest estimate for the voting share of party $j$ is the proportion of voters for this party in the poll, that is:

$$\hat{p_j} = \frac{1}{m} \sum_{i \in I} X_{ij}. \tag{23}$$

By linearity of expectation, we have:

$$E[\hat{p_j}] = \frac{1}{m}\sum_{i\in I} E[X_{ij}] = \frac{1}{m}\sum_{i\in I} p_j = p_j. \tag{24}$$

Thus, the estimator $\hat{p_j}$ is unbiased. To calculate the variance of this estimator we need to consider the pairwise covariances between the $X_{ij}$ r.vs. However, we can make the a simplifying approximation, by assuming that the individuals are sampled **with replacement.** This assumption, while incorrect, simplifies the calculations and provides an excellent approximation when $m \ll n$ because the probability that the same individual will be sampled twice is negligible. We get under this assumption Independence between the $X_{ij}$'s and therefore:

$$Var[\hat{p_j}] \approx \frac{1}{m^2}\sum_{i\in I} Var[X_{ij}] = \frac{1}{m^2}\sum_{i\in I} p_j(1-p_j) = \frac{p_j(1-p_j)}{m}. \tag{25}$$

### 1.7.1 Sampling Ballots

At the night of the elections, it is customary by T.V. stations to conduct exit polls, where voters are asked to specify their votes in the actual elections. This method avoids or at least reduces several of the biases that arise in telephone polls - the individuals asked are only those that actually voted in the elections, and moreover, they are more likely to report correctly their vote in the poll. A disadvantage of these polls is that it is not practical to sample individuals in this manner, and instead a few ballots are sampled, and all voters in these ballots are asked to report their vote. We consider a set $I \subset 1,..,C$ of $b = |I|$ ballots, sampled randomly and uniformly among all $C$ ballots in the country. The proportion of votes for party $j$ in these ballots is used as an estimator for $p_j$:

$$\hat{p_j} = \frac{\sum_{i\in I} n_{ij}}{\sum_{i\in I}\sum_{j'=1}^{K} n_{ij'}}. \tag{26}$$

In similar to random sampling of individuals, it is also possible to show that the estimator $\hat{p_j}$ above is unbiased, and it is possible to derive an approximate formula for the variance. We omit these derivations here, and will calculate the variance empirically using simulations.

### 1.7.2 Stratified Sampling

When we perform random sampling, it is possible that certain areas and populations are not represented in the sample. To reduce this risk, it is possible to design the sample such that we ensure that different groups are represented. This is called stratified sampling ("midgam schavot"). It is desirable to divide to homogeneous groups, and sample ballots from each such group, i.e. to minimize the diversity within a the groups, and maximize the differences between groups. Formally, consider a stratified sample with $L$ strata ("layers"), $\bigcup_{l=1}^{L} A_l = 1,..,C$ and $A_l \bigcap A_r = \emptyset$, $\forall l \neq r$. We divide the $b$ ballots of our poll into $b_1...,b_L$ ballots

where $b_l$ ballots are used for strata $A_l$, with $\sum_{i=1}^{L} b_i = b$. The set of ballots is thus $I = \bigcup_{i=1}^{L} I_l$ with $I_l \subset A_l$ and $|I_l| = b_l$. We next compute the frequency of each party in each strata:

$$\hat{p_j^{(l)}} = \frac{\sum_{i \in I_l} n_{ij}}{\sum_{i \in I_l} \sum_{j'=1}^{K} n_{ij'}}. \tag{27}$$

Next, our estimator is obtained as a weighted average of the strata estimates:

$$\hat{p_j} = \frac{\sum_{l=1}^{L} N_l \hat{p_j^{(l)}}}{\sum_{l=1}^{L} N_l}, \tag{28}$$

where $N_l$ is the total number of eligible votes in strata $l$, given by: $N_l = \sum_{i \in A_l} \sum_{j=1}^{K} n_{ij}$.

The above estimator can be shown to be unbiased, in similar to the random sampling estimator. If the strata $A_l$ and the number of ballots in each strata $b_l$ are chosen carefully (for example we can consider $L = 10$ socio-economic "eshkolot" since we know that voting patters are different between them), this estimator can have a reduced variance, compared to the random sampling estimator. However, a bad choice can actually increase the variance of the estimator.

### 1.7.3 Shrinkage using previous results

We can combine the information obtained from the sample with other prior information to improve the accuracy of our estimator. For example, if we know that a certain party usually gets around 10% of votes, but in a small poll we have conducted this party got 15% of the votes, we may suspect that this high results occurred by chance or due to unknown biases, and reduce our estimate for this party from 15% to a lower figure. To formalize this intuition, let $r_j$ be the total proportion of votes to party $j$ in previous elections (e.g. in the September 2019 elections in Israel), and let $p_j$be the (unknown) proportion in the upcoming elections (e.g. suppose that we wanted to predict the 2020 elections results in advance). Let $\hat{p_j}$ be our estimator from the poll (for example, a random poll of ballots), and let $0 \leq \alpha \leq 1$ be a scalar. Then, we can define the following estimator:

$$\hat{p_j}(\alpha) = \alpha \hat{p_j} + (1 - \alpha) r_j. \tag{29}$$

Suppose that $\hat{p_j}$ is an unbiased estimator for $p_j$ with variance $\sigma^2$. Then, we can compute the bias and variance of the combined estimator $\hat{p_j}(\alpha)$ :

$$E[\hat{p_j}(\alpha)] - p_j = (1 - \alpha)(r_j - p_j). \tag{30}$$

$$Var[\hat{p_j}(\alpha)] = \alpha^2 \sigma^2. \tag{31}$$

and the mean-squared error is:

$$MSE[\hat{p_j}(\alpha)] = (1 - \alpha)^2 (r_j - p_j)^2 + \alpha^2 \sigma^2. \tag{32}$$

Our goal is to minimize the $MSE$, i.e. the sum of the variance and the squared-bias. As we increase $\alpha$ we reduce the bias of the estimator, but increase the variance. This is called bias-variance trade-off.

It is possible to find the optimal $\alpha$ minimizing the $MSE$ as a function of $\sigma^2, r_j$ and $p_j$. In general this optimal $\alpha$ will be smaller than 1 - that is, using the information from the previous elections can reduce our mean-squared error, although it introduces bias, because it reduces the variance.

**Exercise 12.** Find the $\alpha$ value minimizing the $MSE$ in eq. (32)

**Exercise 13.** Suppose that $\hat{p_j}$ is obtained from a random sample of $m$ votes, and that $\alpha^*$ is the optimal value minimizing the MSE in eq. (32). Now, we double the sample size to $2m$ and get a new estimator $\hat{p'_j}$ for the doubled sample size, and compute $\alpha^{*'}$, the value minimizing the MSE in eq. (32) for $\hat{p'_j}(\alpha)$. Prove or disprove: $\alpha^{*'} \geq \alpha^*$. Prove or disprove: $MSE(\hat{p'_j}(\alpha^{*'})) \geq MSE(\hat{p_j}(\alpha^*))$.

# 2 Transfer of voters between Elections

## 2.1 Markov Chains

We first describe more generally an introduction to Markov chains. Formally, a Discrete-time discrete-space Markov chain on $K$ states is a sequence of random variables $X_1, X_2, ...X_n, ... \in \{1, 2, .., K\}$ (the sequence of $X_i's$ can be finite or infinite), such that for all $i > 1$ it holds that $P(X_i|X_1, ..., X_{i-1}) = P(X_i|X_{i-1})$. The Markov chain is associated with a transition matrix $\boldsymbol{M} \in \mathbb{R}_{K\times K}$ such that $P(X_i = k|X_{i-1} = j) = m_{jk}$, $\forall j, k \in \{1, 2, .., K\}$, $\forall i > 1$. The entries $m_{jk}$ are all non-negative, and in addition, the sum of each row is 1: $\sum_{k=1}^{K} m_{jk} = 1 \, \forall j = 1, .., K$. A matrix satisfying these properties is called a *stochastic matrix*.

**Estimating parameters of the chain form data:** Suppose that we have observations $X_1, ..., X_n$ taken from a Markov chain and we want to estimate the unknown matrix $\boldsymbol{M}$. Each element $m_{jk}$ denotes the probability to move to state $k$, given that the chain is currently at state $j$. Therefore, to estimate $m_{jk}$ we only need to consider occasions where the Markov chain has reached state $j$, and out of all these occasions, count in how many of them the chain then moved to state $k$. Formally, our estimator will be:

$$\hat{m}_{jk} = \frac{\sum_{i=2}^{n} \mathbf{1}_{\{\mathbf{X_{i-1}=j}\}} \mathbf{1}_{\{\boldsymbol{X_i=k}\}}}{\sum_{i=2}^{n} \mathbf{1}_{\{\mathbf{X_{i-1}=j}\}}}. \tag{33}$$

When the denominator is zero (i.e. the Markov chain was never at the state $j$), the estimator $\hat{m}_{jk}$ is undefined.

**Exercise 14.** Suppose that we estimate $m_{jk}$ using the maximum likelihood method. That is, write the likelihood $Like(X_1, ...., X_n; \boldsymbol{M})$ and find the matrix $\hat{\boldsymbol{M}}$ maximizing it. Derive the likelihood and the MLE. Do you get the same

estimator as in eq. (33)? You can assume that $X_1$ is set and not randomized - for example we set $X_1 = 1$.

Suppose now that instead of observing $X_1, ..., X_n$ for one Markov chain, we get to observe the chain values for only two time steps, $X_1$ and $X_2$, but get to observe multiple independent such pairs, i.e. $(X_{11}, X_{12}), ..., (X_{c1}, X_{c2})$ where the pairs $(X_{i1}, X_{i2})$ are independent. (There is still dependency **within** each pair, with $P(X_{i2} = k | X_{i1} = j) = m_{jk}$). In the context of elections, one can think of each pair of random variables $(X_{i1}, X_{i2})$ as representing the votes of a single individual $i$ in two subsequent elections, and such data can for example be obtained by a poll, where each responder in the poll is asked about the parties she voted for in the two elections. For this data, we can estimate $\boldsymbol{M}$ in a similar manner to eq. (33), using the following estimator:

$$\hat{m}_{jk} = \frac{\sum_{i=1}^{c} \mathbf{1}_{\{\mathbf{X_{i1}=j}\}} \mathbf{1}_{\{\boldsymbol{X_{i2}=k}\}}}{\sum_{i=1}^{c} \mathbf{1}_{\{\mathbf{X_{i1}=j}\}}}. \tag{34}$$

While this estimator can be obtained from polling data at the individual level, we would like to get a similar estimator from the more reliable data of the actual elections. The problem is that for the election we have only aggregate data, where we know the total number of votes for each party in each ballot, but do not know how did individuals change their votes between the two elections. The next sections describe how to estimate the matrix $\boldsymbol{M}$ from such aggregate elections data.

## 2.2 Voter Transfer

Consider next two elections (in our case, for example Sep 2019 vs. March 2020). We are interested in analyzing the **changes** between the elections. There are two important differences for our data, compared to the Markov chain scenario from the previous section:

1. We don't observe the values of individuals $(X_{i1}, X_{i2})$, but only the aggregate statistics, which are sums over such $X_i$'s over all individuals $i$ in the same ballot.

2. While in Markov chains the states $\{1, .., K\}$ remain the same, here the states themselves (representing parties) change between the two time points (two elections), because parties can split, merge, disappear etc.

Let $n_{ij}^{(a)}$ be the number of votes for party $j$ in ballot $i$ as before, but for elections $a$, and similarly $n_{ij}^{(b)}$ the number of votes for party $j$ in ballot $i$ for elections $b$. Denote the corresponding matrices by $\boldsymbol{N}^{(a)} \in \mathbb{R}_{C \times K^{(a)}}, \boldsymbol{N^{(b)}} \in \mathbb{R}_{C \times K^{(b)}}$ where $K^{(a)}, K^{(b)}$ are the number of parties in the two elections, and here $C$ is the number of **shared** ballots - ballots we identified as the same in the two elections (we remove the remaining ballots from each election). We will denote by $\boldsymbol{X}$ the random variable representing the vote of a single individual. Specifically, let

$\boldsymbol{X^{(a)}}, \boldsymbol{X^{(b)}}$be the random variables representing the votes of an individual in the two elections, with $\boldsymbol{X^{(a)}} \in \{1, .., K^{(a)}\}, \boldsymbol{X^{(b)}} \in \{1, .., K^{(b)}\}$.

*Remark* 15. The above model does not explicitly address the issue eligible voters ("baalei zchut bhira") that decide not to vote (i.e. not included in the total votes, "kolot ksherim"). In practice, many voters for a certain party in one of the two elections can actually not vote in the other elections, and we would like to model these transitions as well. This is achieved simply by increasing $K^{(a)}, K^{(b)}$ by one to include in the parties list an additional 'party' indicating no voting, with the number of voters defined as the difference between the number of eligible voters and the number of actual votes in each ballot. We assume that this step is performed, if desired, at the beginning, as pre-processing, and ignore it here in order to simplify notation.

We introduce the following variables: $m_{jk} = Pr(\boldsymbol{X^{(b)}} = k | \boldsymbol{X^{(a)}} = j)$ in a matrix $\boldsymbol{M} \in \mathbb{R}_{K^{(a)} \times K^{(b)}}$. The matrix $\boldsymbol{M}$ represents changes in voting preferences (including from voting for a specific party to no voting and vice versa), where $m_{jk}$ is the probability that voter of party $j$ in the first election will vote for party $k$ in the second elections. Our basic model assumes that these probabilities are constant for all voters throughout the country, and that the voting decisions on the second elections of different voters are independent given their votes in the first elections. Our goal is to estimate the matrix $\boldsymbol{M}$ from the results of the two elections. Our model and estimators are taken from analysis done by Harel Cain and Itamar Mushkin, with small modifications.

## 2.3 Estimating $\boldsymbol{M}$

Consider the results for ballot $i$ in the second elections, namely the vector of number of votes for each party: $(n_{i1}^{(b)}, ..., n_{iK^{(b)}}^{(b)})$. We can track back how did the $n_{ik}^{(b)}$ voters of party $k$ voted in the first election. For each party $j$ in the first election, the expected number of voters who shifted to party $k$ in the second elections is $n_{ij}^{(a)} m_{jk}$. We therefore get:

$$E[n_{i1}^{(b)}] = \sum_{j=1}^{K^{(a)}} n_{ij}^{(a)} m_{jk}, \; \forall i = 1, .., C \; \forall k = 1, .., K^{(b)}. \tag{35}$$

We can therefore estimate the $m_{jk}$ values by penalizing deviations of $n_{ik}^{(b)}$ from its expectation over all parties and ballots. Namely, we can solve the following least-squares problem:

$$\hat{\boldsymbol{M}} = argmin_M \sum_{i=1}^{C} \sum_{k=1}^{K^{(b)}} (\sum_{j=1}^{K^{(a)}} n_{ij}^{(a)} m_{jk} - n_{ik}^{(b)})^2. \tag{36}$$

The above minimization problem can be written in matrix form:

$$\hat{\boldsymbol{M}} = argmin_M ||\boldsymbol{N}^{(a)} \boldsymbol{M} - \boldsymbol{N}^{(b)}||_F^2 \tag{37}$$

where $||\cdot||_F^2$ is the squared Frobenius norm of a matrix, defined as the sum of all of its elements squared: $||A||_F^2 \equiv \sum_{i,j} a_{ij}^2$.

Estimating $\boldsymbol{M}$ according to the above least-squares criteria results a multiple linear regression problem (without intercept). The least squares solution is:

$$\hat{\boldsymbol{M}} = [\boldsymbol{N}^{(a)^T}\boldsymbol{N}^{(a)}]^{-1}\boldsymbol{N}^{(a)^T}\boldsymbol{N}^{(b)} \tag{38}$$

which is a simple generalization of the standard least-squares estimator, except that the response $\boldsymbol{N^{(b)}}$ and the parameters $\boldsymbol{M}$ are matrices and not vectors.

To see this, we can divide the $C \times K^{(b)}$ observations in eq. (36) into $K^{(b)}$ distinct groups of size $C$, one for each party in the second elections.

For party $k$, only the parameters $m_{1k}, ..., m_{K^{(a)}k}$ (i.e. the $k$-th column of $\boldsymbol{M}$) are relevant, and we can estimate each column $\boldsymbol{M_{|k}}$ of $\boldsymbol{M}$ separately by solving the following least-squares linear regression problem:

$$\hat{\boldsymbol{M}}_{|k} = argmin_{M_{|k}} \sum_{i=1}^{C}(\sum_{j=1}^{K^{(a)}} n_{ij}^{(a)} m_{jk} - n_{ik}^{(b)})^2. \tag{39}$$

The least squares solution obtained from the design matrix $\boldsymbol{N^{(a)}}$and outcome vector given by the $k$-th column of the second elections matrix, $\boldsymbol{N_{|K}}^{(b)}$ :

$$\hat{\boldsymbol{M}}_{|k} = [\boldsymbol{N}^{(a)^T}\boldsymbol{N}^{(a)}]^{-1}\boldsymbol{N}^{(a)^T}\boldsymbol{N_{|K}}^{(b)}. \tag{40}$$

Concatenating the columns $\boldsymbol{M_{|k}}$ together gives the matrix solution in eq. (38).

### 2.3.1 Improving the Basic Estimator

The above estimator for $\boldsymbol{M}$ suffer from several problems:

1. First, the interpretation of $\boldsymbol{M}$ as a transition probabilities matrix is not consistent with the fact that the total number of eligible voters ('baalei zchut bhira') may change between the two elections. If $\tilde{n}_{i\bullet}^{(a)} \neq \tilde{n}_{i\bullet}^{(b)}$ , we can for example normalize the votes in both elections, and change eq. (36) to be with voting proportions $q_{ij}^{(a)}, q_{ij}^{(b)}$ instead of actual counts $n_{ij}^{(a)}, n_{ij}^{(b)}$. The proportions here are different than in Section 1, because we compute proportions out of the total number of eligible voters ('baalei zhut bhira'), and not out of the number of actual voters ('kolot ksherim').

2. In addition, we can impose constrains on the values of $m_{jk}$which we know are holding in our model. In particular, we know that $m_{jk} \geq 0$, $\forall j, k$ and that $\sum_{k=1}^{K^{(B)}} m_{jk} = 1$, $\forall j = 1, .., K^{(a)}$ because every voter in the first election voted to exactly one party (including 'non-voting') in the second elections. There are in general three ways of imposing additional knowledge on our parameters when estimating them:

(a) Assuming a prior distribution on the parameters values before we see or use the data, and updating our belief about the parameters when we have the data. This is the common theme in *Bayesian Statistics*.

(b) Adding constraints to the optimization problem, resulting in a **constrained** optimization problem.

(c) Using the constraints in a post-processing step on the estimated parameters after we solve the (unconstrained) optimization problem.

We will use mainly strategy (b) above, (to be described next), but will also use (c) and possible combinations between the two approaches.

We can estimate $\boldsymbol{M}$ by solving the **constrained** optimization problem:

$$\hat{\boldsymbol{M}} = argmin_{M:M\in S}||\boldsymbol{N}^{(a)}\boldsymbol{M} - \boldsymbol{N}^{(b)}||_F^2 \tag{41}$$

where $S \subset \mathbb{R}_{K^{(a)}\times K^{(b)}}$ is some subset of matrices satisfying our constraints. For example, we can define $S$ to be the set of all stochastic matrices: $S = \{\boldsymbol{M} : m_{jk} \geq 0,\ \forall j,k, \sum_{k=1}^{K^{(B)}} m_{jk} = 1,\ \forall j = 1,..,K^{(a)}\}$. In general, the problem in eq. (41) does not have a closed from solution (unlike the unconstrained least-squares problem with the solution in eq. (38)), and the difficulty of solving the problem depends on the constrained set $S$.

We start with a simpler set $S$, namely that of *non-negative matrices*: $S = \{\boldsymbol{M} : m_{jk} \geq 0,\ \forall j,k = 1,..,K^{(a)}\}$. Solving problem (41) for this set $S$ is called a Non Negative Least Squares (NNLS) problem, and there are efficient algorithms for this problem.

More generally, our least-squares target function $f(\boldsymbol{M}) = ||\boldsymbol{N}^{(a)}\boldsymbol{M} - \boldsymbol{N}^{(b)}||_F^2$ is convex in the entries of $\boldsymbol{M}$.

**Definition 16.** A function $f : A \to \mathbb{R}$ is called convex if $\forall x, y \in A$ and $\forall \alpha \in [0,1]$ we have:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y) \tag{42}$$

**Exercise 17.** Prove that the least squares cost $||\boldsymbol{N}^{(a)}\boldsymbol{M} - \boldsymbol{N}^{(b)}||_F^2$ is a convex function of $\boldsymbol{M}$ (for constant $\boldsymbol{N}^{(a)}, \boldsymbol{N}^{(b)}$).

It is also desirable to minimize the squared loss over $S$ when $S$ is a convex set:

**Definition 18.** A set $A$ is called a convex set if $\forall x, y \in A$ and $\forall \alpha \in [0,1]$ we also have: $\alpha x + (1-\alpha)y \in A$.

**Exercise 19.** Prove that the set of matrices $S = \boldsymbol{M} : m_{jk} \geq 0,\ \forall j,k, \sum_{k=1}^{K^{(B)}} m_{jk} = 1,\ \forall j = 1,..,K^{(a)}$ is convex.

When $S$ is convex, the resulting problem is called a convex optimization problem, and there are usually efficient algorithms for solving the problem.

**Definition 20.** The problem: $\hat{x} = argmin_A f(x)$ where $A$ is a convex set, and $f$ is a convex function is called a convex optimization problem.

Solving problem (41) for the above $S$ representing normalized and non-negative matrices and more generally for convex sets $S$ can be done for example using the $CVX$ software package.

Alternatively, we can obtain faster solutions by using only some of the constraints or no constraints at all. Then, after we solve the problem, we can modify the obtained estimator to satisfy the constraints - that is, normalize each row to sum to one in order to impose the constrains $\sum_{k=1}^{K^{(B)}} \hat{m}_{jk} = 1$, and/or setting negative estimators $\hat{m}_{jk}$ to zero. However, this heuristic is not guaranteed to find the optimal $\boldsymbol{M}$ satisfying both the non-negativity and sum constraints, i.e. the optimum of the problem in eq. (41).

**Other improvements** An additional heuristic is thresholding small values of $m_{jk}$ to zero, if we believe that many of the small non-zero values are the result of estimation error.

We can also use our statistical model to compute a likelihood function for the data of $\boldsymbol{N}^{(b)}$ and find $\boldsymbol{M}$ maximizing this likelihood, instead of minimizing the squared loss. The likelihood will be based on the multinomial distribution.