



UNIVERSITY OF THE PHILIPPINES

Bachelor of Science in Applied Physics

Crizzia Mielle M. de Castro

Extracting Philippine voting patterns through hyperspectral unmixing

Thesis Adviser:

May T. Lim, Ph.D.

National Institute of Physics

University of the Philippines Diliman

Date of Submission:

June 2020

Thesis Classification:

P

This thesis is not available to the public

*Para sa bayan!
Ad astra per aspera*

Acknowledgments

I would like to acknowledge the help of the following:

Ma'am May for providing me with guidance and new skills. I started out as a junior with barely any experience on research, but now I am graduating carrying with me everything you taught us.

CX Team to the most supportive subgroup I could've asked for, for helping me whenever I get stuck with research, and for always making me laugh.

Werhu, Kenneth, and Kelvin we've come so far. Thank you for all your insights and advice that helped me a ton with research. I'm so proud of us!

The Instrumentation Physics Lab for all the new skills and knowledge I gained as a member.

UP AstroSoc for being the most amazing and supportive second family anyone could ever ask for. You gave me something to get excited about every day. To the stars through difficulties!

Diwata for helping me get out of my comfort zone. I couldn't have asked for better org batch mates. I will forever cherish our memories together, both fun and stressful.

Liz, EJ, and Aiye for making every day in college memorable and fun. Thank you for our daily food trips and card games together. There is never a dull moment with you.

Da and Somido my sisters (though not by blood), for being there for me in my highest, and most importantly, my lowest points. You are my rock, and I love you, always.

Isay and Kosmo my cats, I know you can never actually read this, but know that you offered me a unique kind of companionship no person can ever match.

My Family for the unwavering love and support in everything I do. All my accomplishments pale in comparison to seeing you proud of me. Para sa inyo 'to, Me at Tay.

I would not be here today if not for all of you. Thank you, and this is for you!

I would like to acknowledge the financial support of the following:

DOST-SEI for my scholarship.

CHED-PCARI project DARE (IIID-2016-006) for providing access to resources.

ABSTRACT

EXTRACTING PHILIPPINE VOTING PATTERNS THROUGH HYPERSPECTRAL UNMIXING

Crizzia Mielle M. de Castro
University of the Philippines (2020)

Adviser:
May T. Lim, Ph.D.

Possible voting patterns in elections provide insight on the political and societal behaviors of a community. To extract possible voting patterns in the Philippines, we applied two techniques to the results of the 2013, 2016, and 2019 senate elections: K-means clustering and hyperspectral unmixing. K-means clustering groups the data set into K clusters. Hyperspectral unmixing—a technique commonly applied to satellite images—follows three steps: (1) reducing data dimension, (2) unmixing the pattern spectral signatures, and (3) inverting the spectral signatures to estimate the corresponding weights. We applied K-means clustering to the candidates' cumulative vote distributions. K-means clustering merely split the top and bottom ranking candidates, while hyperspectral unmixing extracted five, six, and six distinct voting archetypes from the 2013, 2016, and 2019 senate elections, respectively. The voting patterns in 2013 are: opposition, pro-RH bill, anti-RH bill, joint campaign, and conservatives. The voting patterns in 2016 are: popular coalitions, provincial politicians, winners, celebrities, TV and written news popularity. The voting patterns in 2019 are: previous senators, conservatives, multimedia popularity, island group association, celebrities, and opposition. Our scheme introduces a new approach to analyzing high-dimensional mixed election results.

PACS: 07.05.Pj Image processing algorithms, 89.75.Kd Pattern formation in complex systems, 78.40.-q Reflection spectra

Table of Contents

Acknowledgments	ii
Abstract	iv
List of Figures	v
1 Introduction	1
1.1 Statement of the problem	2
1.2 Objectives	3
1.3 Significance	3
1.4 Scope and limitations	3
2 Theory	4
2.1 Clustering	4
2.1.1 Principal component analysis	4
2.1.2 K-means clustering	7
2.1.3 Silhouette analysis	8
2.2 Hyperspectral unmixing	10
2.2.1 Dimension reduction	11
2.2.2 Unmixing	14
2.2.3 Inversion	23
3 Methodology	26
3.1 Election dataset	26
3.2 Clustering candidates	27
3.2.1 Dimension reduction using PCA	28
3.2.2 K-means clustering on candidates' cumulative vote distributions	28
3.3 Hyperspectral unmixing on election results	29
3.3.1 Finding dominant archetypes	30
4 Results and Discussion	32
4.1 K-means clustering of candidates	32
4.2 Voting archetypes after hyperspectral unmixing	37
4.3 Comparison with studies using exit polls and pre-election surveys . .	44
5 Conclusions and Recommendations	46

List of Figures

2.1	(a) 2D point (dark circle) projected onto 1D lines u , v , x , and y . Projections are illustrated as broken lines; (b) Variance of projected points along the 1D lines. The variance is highest along u . (figure from Ref. [15])	6
2.2	Illustration of K-means clustering for $K = 2$. The “x” marks the cluster centroids, and different colors represent different clusters.	8
2.3	Visualization of elements involved in calculating the silhouette score $s(i)$. Silhouette analysis compares the distance of i to cluster B (blue lines) and A (red lines).	9
2.4	Comparison of linear and nonlinear unmixing (figure from Ref. [6]) . .	10
2.5	Diagram of a 2-dimensional simplex with three archetypes (vertices). Red points are hyperspectral data, while green points are the archetypes.	14
2.6	Comparison of hyperspectral data (a) with, and (b) without the pure pixel assumption. Data points can be found in the vertices of the simplex in (a), while none in (b).	15
2.7	Illustration of the VCA algorithm. VCA projects S_p along \mathbf{f}_1 and \mathbf{f}_2 , then sets the extremes as the archetypes. (figure from Ref. [24]) . . .	16
2.8	Sample of noisy data where some data points lie outside the true simplex (solid triangle). The incorrect simplex (dashed triangle) estimates archetypes (blue points) far from the true values (green points).	18
3.1	Sample election data for every year	27
3.2	2016 election results per region. Our analysis uses the first two columns to determine each precinct’s region.	30
3.3	Sample shapefiles of Philippine maps. We used (a) for 2013 and 2019, and (b) for 2016.	30
4.1	Reduced features of candidates’ cumulative vote distributions along the first two PCs. PCA reduced the number of features from 60 to 2.	33
4.2	Distortion (overall mean square error) vs number of clusters for K-means clustering per year. The elbow looks to be around $K = 2 - 5$	33
4.3	Silhouette plots of $K = 2, 3, 4, 5$ in K-means clustering per year. The red line denotes the overall mean silhouette score. Cluster silhouette size is proportional to the number of data points in cluster.	35
4.4	Clusters of candidates formed after using K-means clustering for $K = 2, 3, 4, 5$ on candidates’ cumulative votes distribution. The features were reduced from 60 to two using PCA.	36

4.5	Normalized vote intensities of archetypes per year. Archetypes were arranged based on overall mean weight, and candidates were arranged based on total votes.	37
4.6	Dominant archetypes per year based on median weight per province/region. The archetypes in the legend are arranged from highest to lowest overall mean weight.	42
4.7	Dominant archetypes per year based on mean weight per province/region. The archetypes in the legend are arranged from highest to lowest overall mean weight.	43

Chapter 1

Introduction

Post-election analysis often dissects patterns in voting. In the Philippines, voting pattern studies typically use data from surveys: either exit polls or pre-election surveys [1–4]. In exit polls, interviewers approach willing voters after exiting the voting precincts to survey how they voted [2]. In pre-election surveys, interviewers approach random willing respondents months to weeks before voting day [3].

With the switch to electronic voting in Philippine elections, the Commission on Elections (COMELEC) reports precinct-level results to maintain transparency [5]. Machines in counting centers count ballots in the order of their arrival. After passing through the canvassing boards in local government units, the national canvassing board releases real-time partial unofficial results to the public through their website.

Voting precincts compose Philippine election results, similar to how pixels comprise satellite images. Each pixel in a satellite image corresponds to an emission spectrum, whereas each clustered precinct corresponds to a “vote spectrum”. Emission spectra of pixels contain a reflectance intensity vs wavelength profile, while “vote spectra” of clustered precincts contain a votes vs candidate profile. The structural similarities of election results and satellite images validates using satellite image processing techniques on election results.

Satellite images typically contain thousands of pixels with spectral resolutions ranging from tens to hundreds of wavelengths. Recurring patterns in satellite images interest us more than the individual pixels themselves. Thus, satellite images need a fast and efficient method of pattern extraction that can handle vast amounts of image

Part of this work appears in C. M. de Castro and M. Lim. Determining senatorial voting archetypes through hyperspectral unmixing, Proceedings of the Samahang Pisika ng Pilipinas 37, SPP-2019-PB-32 (2019). URL: <https://paperview.spp-online.org/proceedings/article/view/SPP-2019-PB-32>.

data. This is where hyperspectral unmixing comes in.

Hyperspectral unmixing decomposes pixel spectra into separate spectral signatures called archetypes (basic patterns) with a set of corresponding weights [6]. Weights represent how much of each archetype comprises each pixel.

Studies conducted by industries and scientific institutions utilize hyperspectral unmixing to process images and find materials of interest invisible to the naked eye. Hyperspectral unmixing can determine the chemical composition of soil captured by land satellites through mineral mapping [7], environmental surveillance [8], and precision agriculture [9, 10]. Some industrial applications of hyperspectral unmixing include food safety of dairy products based on the presence of mold [11], and assessment of tissue oxygenation in ischemic wounds [12].

A study on the Pareto Task Inference (ParTI) method applied hyperspectral unmixing to infer key biological tasks contained in tissue genes [13]. The study used high dimensional biological datasets such as the gene expression of human breast tumor tissues. The notion of extracting patterns from high dimension data using hyperspectral unmixing prompted us to apply the same methods to the Philippine senate election results.

The Philippines holds senate elections every three years to fill up 12 empty Senate seats [2]. Voters select candidates for all positions, both national and local, during presidential elections every six years. On the other hand, voting excludes the president, vice president, and barangay officials during midterm elections every three years.

1.1 Statement of the problem

Our study uses partial precinct-level election results, instead of election survey results. Our choice of dataset introduces the question of how to approach election results analysis given a coarser resolution. We consider precinct-level results as an ensemble sum of individual votes. In turn, municipality results sum precinct-level votes; province results sum municipality-level votes; and the national results sum province-level votes.

Assuming sufficient mixing of individual votes in the precinct-level, we posit that precinct-level results can be decomposed to its constituent voting patterns. To perform pattern extraction, we implement two techniques: (1) K-means clustering, a

classic approach to pattern extraction from big data, and (2) hyperspectral unmixing, an image processing technique commonly applied to satellite images.

1.2 Objectives

Our study compares two pattern extraction techniques—K-means clustering and hyperspectral unmixing—in obtaining possible voting patterns in precinct-level Philippine senate election results. We also aim to map the dominant voting patterns per province across the Philippines using the pattern weight distributions.

1.3 Significance

Voters choose at most 12 from a pool of 30-60 senate candidates. The impracticality of enumerating all possible voting combinations necessitates using an efficient pattern extraction technique to summarize senate election results into comprehensive archetypes. Our study provides a new approach to find underlying patterns in high-dimension election data by using hyperspectral unmixing.

The possible voting archetypes obtained through hyperspectral unmixing can provide an overview of the political and electoral situation in the Philippines. Instead of looking at all possible voting combinations, hyperspectral unmixing extracts comprehensive archetypes that allow for more manageable voting analysis. Comparing the three years of voting patterns can depict how social and historical context affect Filipino voting tendencies over time.

1.4 Scope and limitations

Philippine automated national elections began in 2010, but we were only able to cover three years due to data availability: 2013 (midterm), 2016 (presidential), and 2019 (midterm). We used partial senate election results (summarized in Table 3.1), and did not consider other government positions besides senators.

Chapter 2

Theory

In this chapter, we discuss the mathematical theories behind clustering (Section 2.1) and hyperspectral unmixing (Section 2.2), the two pattern extraction techniques implemented in our study.

2.1 Clustering

Clustering groups datasets according to intrinsic features to extract data structure and patterns [14]. Since clustering excludes attaching any class labels, the contextual interpretation of formed clusters is left to the user. In this section, we talk about the three general steps of clustering we followed: (1) dimension reduction using Principal Component Analysis (Section 2.1.1), (2) K-means clustering (Section 2.1.2), and (3) determining K using the elbow method and silhouette analysis (Sections 2.1.2 and 2.1.3).

2.1.1 Principal component analysis

Principal Component Analysis (PCA) reduces data dimension while retaining original trends and patterns, making it a useful initial step before clustering [15–17]. To reduce data dimension, PCA first projects the data from the original subspace to a different subspace that minimizes the total distance between the projected and original points—thereby maximizing the variance of the projected points.

PCA expresses the projected data called principal components (PC) as linear combinations of the original data given by

$$\sum_{j=1}^p a_j \mathbf{x}_j = \mathbf{X}\mathbf{a} \quad (2.1)$$

where \mathbf{X} is an $n \times p$ matrix containing the n original data points with p features each, and the vector \mathbf{a} contains the unknown constants a_1, a_2, \dots, a_p [17]. The variance of the projected data is given by

$$\text{var}(\mathbf{X}\mathbf{a}) = \mathbf{a}^T \mathbf{S} \mathbf{a} \quad (2.2)$$

where \mathbf{S} is the covariance matrix of \mathbf{X} . PCA determines the vector \mathbf{a} that maximizes Eqn 2.2 [17]. PCA restricts working with unit-norm vectors to obtain a well-defined solution to \mathbf{a} , meaning $\mathbf{a}^T \mathbf{a} = 1$. Thus, maximizing Eq. 2.2 corresponds to maximizing

$$\mathbf{a}^T \mathbf{S} \mathbf{a} - \lambda (\mathbf{a}^T \mathbf{a} - 1) \quad (2.3)$$

where λ is a Lagrange multiplier. We then equate to $\mathbf{0}$ (null matrix) the first derivative of Eq. 2.3 with respect to \mathbf{a} .

$$\mathbf{S} \mathbf{a} - \lambda \mathbf{a} = 0 \iff \mathbf{S} \mathbf{a} = \lambda \mathbf{a} \quad (2.4)$$

From Eq. 2.4, we can see that \mathbf{a} and λ correspond to the eigenvector and eigenvalue of \mathbf{S} , respectively. By plugging in Eq. 2.4 to Eq. 2.2, we get

$$\text{var}(\mathbf{X}\mathbf{a}) = \mathbf{a}^T \mathbf{S} \mathbf{a} = \mathbf{a}^T \lambda \mathbf{a} = \lambda \mathbf{a}^T \mathbf{a} = \lambda \quad (2.5)$$

The last part holds, because \mathbf{a} is a unit-norm vector. To maximize the variance of the projected data, we need to obtain the largest eigenvalue of the covariance matrix \mathbf{S} . The corresponding eigenvector of the largest eigenvalue correspond to \mathbf{a} . Given \mathbf{a} , we can solve for the first principal component using Eq. 2.1. The succeeding PCs can be solved using the corresponding eigenvectors of the remaining eigenvalues of \mathbf{S} arranged in decreasing order. Explicitly, all the principal components are given by

$$\mathbf{X} \mathbf{a}_k = \sum_{j=1}^p a_{kj} \mathbf{x}_j \quad (2.6)$$

where the maximum number of PCs k equates to the number of features p in the original data. Choosing the number of PCs to use in dimension reduction depends on the variance explained by each PC. Using more PCs to represent the data increases the total variance explained by increments typically much smaller than the previous

PC. Using all PCs explains 100% of the overall variance. To evaluate the variance of each PC, we use

$$\pi_j = \frac{\lambda_j}{\sum_{j=1}^p \lambda_j} \times 100\% \quad (2.7)$$

Before obtaining linear combinations given by Eqn 2.1, the standard PCA approach centers the data using

$$x_{ij}^* = x_{ij} - \bar{x}_j; \quad i = 1, 2, \dots, n \quad (2.8)$$

where \bar{x}_j denotes the mean of the original data points along feature j . The covariance matrix of centered and uncentered data points remains unchanged.

Figure 2.1 shows sample data geometrically projected by PCA onto different dimensions x, y, u , and v . From the sample data, u (blue line) represents the dimension with the highest data variance, so PCA sets u as PC1. The succeeding PCs after PC1 must be uncorrelated and geometrically orthogonal to the previous PCs, so PCA sets v (orange line) as PC2. The number of features or number of data points, whichever is smaller, dictates the maximum number of principal components. In this case, PCA only finds two principal components, because the original data only has two features. Since PC2 (0.22) has a much smaller data variance than PC1 (1.78), we may drop the projected features from PC2 and only consider features along PC1. Thus, PCA reduced the number of dimensions from two to one.

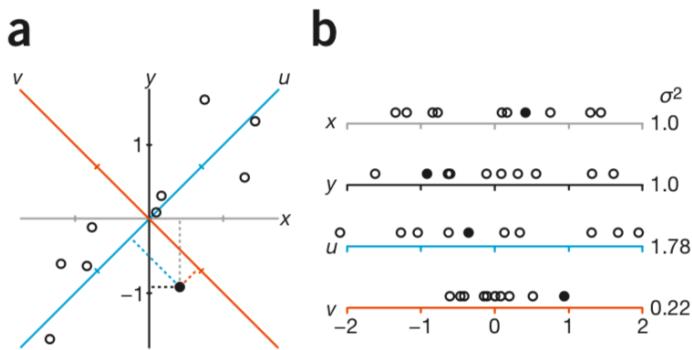


Figure 2.1: (a) 2D point (dark circle) projected onto 1D lines u , v , x , and y . Projections are illustrated as broken lines; (b) Variance of projected points along the 1D lines. The variance is highest along u . (figure from Ref. [15])

We consider certain limitations when interpreting results from PCA: (1) underlying data structures assumed to be linear; (2) non-orthogonal patterns may not be detected; and (3) PCA maximizes variance, but does not necessarily cluster [15].

2.1.2 K-means clustering

K-means clustering groups the observed data points into K (set by the user) clusters in a minimum mean squared error sense [18]. The algorithm implements the following general steps as illustrated in Fig 2.2 [14]:

1. Initialize K cluster classifications using random seed points as cluster centroids.
2. Update classifications by assigning each data point to closest cluster centroid.
3. Calculate new cluster centroids.

The centroid of a cluster c_k is given by

$$\mu_k = \frac{1}{N_k} \sum_{i=0}^{N_k} \mathbf{x}_{ki} \quad (2.9)$$

where N_k is the number of data points assigned to cluster k , and \mathbf{x}_{ki} is the i th data point assigned to cluster k [18]. The centroids don't necessarily have to be one of the observed data points.

K-means clustering iterates the second and third steps until the overall mean squared error (MSE) between the cluster centroid and cluster points reaches a threshold value. The MSE of the distances for a cluster c_k is given by

$$J(c_k) = \sum_{i=0}^{N_k} \|\mathbf{x}_{ki} - \mu_k\|^2 \quad (2.10)$$

where $\|\dots\|$ denotes the distance between two points. K-means clustering stops iterations when the sum of the squared errors over all K clusters given by

$$J(C) = \sum_{k=0}^K \sum_{i=0}^{N_k} \|\mathbf{x}_{ki} - \mu_k\|^2 \quad (2.11)$$

reaches the threshold value.

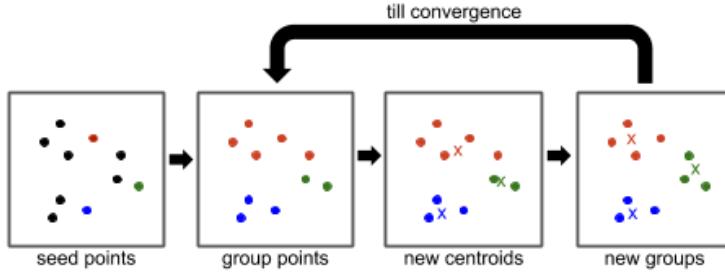


Figure 2.2: Illustration of K-means clustering for $K = 2$. The “x” marks the cluster centroids, and different colors represent different clusters.

The user specifies three parameters in the K-means clustering: number of clusters K , cluster initialization, and distance metric [14]. Previous studies present different methods for determining the number of clusters [19], but our study focuses on the Elbow Method and on Silhouette Analysis. Choosing any random K seed points suffices cluster initialization, because the algorithm should converge regardless. Our study uses K random points from the dataset as seed points for simplicity. K-means clustering typically uses the Euclidean distance metric (which results to spherical clusters), so our study also applies the same metric.

Elbow method

The Elbow Method graphs the overall MSE from Eqn 2.11 along the y-axis and the number of clusters along the x-axis [20, 21]. A distinct flattening of the graph means that increasing the number of clusters decreases the overall MSE only by a small amount. The *elbow* of the graph (point where flattening begins) may represent the optimal number of clusters. The Elbow Method narrows down the user’s choice for K to a small range of values. However, the subjectivity of determining the graph *elbow* necessitates the need to apply another technique for choosing the number of clusters.

2.1.3 Silhouette analysis

Silhouette analysis determines which data points are well-classified, misclassified, or are near the cluster boundary for different number of clusters [22]. The analysis provides the user with quantitative metrics using silhouette scores (Table 2.1) in choosing the optimal number of clusters. Silhouette analysis visualizes the structure of each cluster through the silhouette score $s(i)$. The number $s(i)$ is given by Eqn 2.12 [22, 23] and visualized in Fig. 2.3.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{where} \quad (2.12)$$

$a(i)$ = mean distance of i to other points in its cluster A

$$b(i) = \min_{C \neq A} d(i, C), \quad \text{and}$$

$d(i, C)$ = mean distance of i to all points in other clusters C

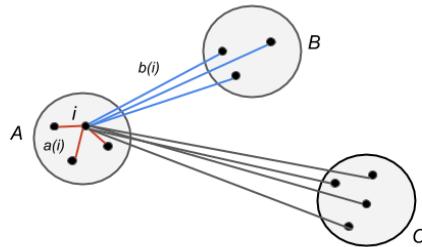


Figure 2.3: Visualization of elements involved in calculating the silhouette score $s(i)$. Silhouette analysis compares the distance of i to cluster B (blue lines) and A (red lines).

The cluster assigned to $b(i)$ represents the second best choice for i 's cluster classification. Eqn 2.12 limits the value of $s(i)$ to $-1 \leq s(i) \leq 1$, with the assumption that $K > 1$.

Table 2.1: Conclusions based on the calculated silhouette score $s(i)$ per data point i

$s(i)$	meaning
closer to 1	well-classified
closer to 0	near cluster boundary
negative	misclassified

Silhouette analysis sorts the points into their assigned clusters from lowest to highest $s(i)$, then plots $s(i)$ in a 2D plot. The y-axis corresponds to each data point, while the x-axis corresponds to $s(i)$. To create cluster silhouettes, the analysis draws a horizontal line from $s(i) = 0$ to the corresponding $s(i)$ per data point.

The silhouettes plot exposes artificial cluster fusions or divisions from under-specifying or over-specifying the number of clusters, respectively [22]. When determining how many clusters K to set, the user may choose the K that maximizes the overall mean $\bar{s}(i)$ [23], or that has the most points with $s(i) \geq \bar{s}(i)$.

2.2 Hyperspectral unmixing

Hyperspectral unmixing methods classify as either linear or nonlinear (illustrated in Fig. 2.4). Linear hyperspectral unmixing assumes that each pixel linearly combines archetype spectral signatures multiplied by their corresponding weights. For example, a pixel may contain 40% land (brown), 20% road (gray), and 40% trees (green) (Fig. 2.4a). Under the linear unmixing model, the pixel's spectral signature sums 40% the spectral signature of land, 20% of roads, and 40% of trees.

The linear unmixing model assumes that mixing occurs in the hyperspectral sensor at the pixel level [6]. The light reflected from each material, despite being distinguishable from each other, get mixed due to the instrument's coarse resolution. In Fig. 2.4a, the light reflected by the pixel clearly distinguishes the three archetypes: land, road, and trees. However, the sensor can only detect the overall summed pixel spectra given by y . Thus, linear unmixing aims to decompose the pixel spectra y into its constituent archetype spectra \mathbf{m}_1 , \mathbf{m}_2 , and \mathbf{m}_3 .

On the other hand, the nonlinear unmixing model assumes that mixing occurs as the light interacts with multiple other materials in the scene [6]. For example sunlight hits a tree, the land, another tree, then finally the hyperspectral sensor on the satellite (Fig. 2.4b). Nonlinear unmixing also considers the light scattered by air particles before reaching the hyperspectral sensor.

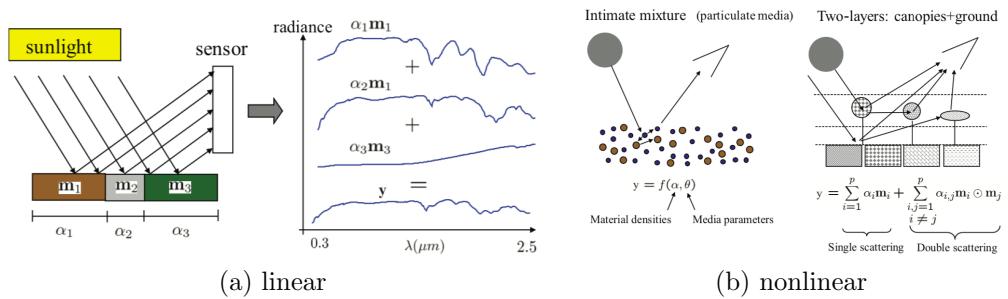


Figure 2.4: Comparison of linear and nonlinear unmixing (figure from Ref. [6])

Our study applies linear unmixing, because we assume sufficient mixing of individual votes in precinct-level election results. Just like the hyperspectral sensor in linear unmixing, the precinct-level results only include the overall votes per precinct, but not the individual votes. Through linear unmixing, we can decompose further the precinct election results into their constituent individual voting archetypes.

The linear hyperspectral unmixing model estimates \mathbf{M} and \mathbf{A} in

$$\mathbf{Y} = \mathbf{MA} + \mathbf{N}, \quad (2.13)$$

where \mathbf{Y} is the $L \times N$ observed spectral matrix, \mathbf{M} is the $L \times k$ mixing matrix containing the archetypes' spectral signatures, \mathbf{A} is the $k \times N$ abundance matrix containing the archetypes' weights per pixel, and \mathbf{N} is the noise matrix [24–27]. For the dimensions of the matrices, $L = \text{no. of spectral bands (wavelengths)}$, $N = \text{no. of observed data points}$, and $k = \text{no. of archetypes}$. The solution to Eq. 2.13 is subject to the positivity constraint $\mathbf{A} \succeq 0$ and the sum constraint $\mathbf{1}_k^T \mathbf{A} = \mathbf{1}_n$. The archetype weights should be positive and should sum to one for each data point.

Linear hyperspectral unmixing generally follows a three-step processing chain: (1) dimension reduction, (2) unmixing, and (3) inversion. The remaining sections of this chapter discuss in detail the algorithms we used for each step.

2.2.1 Dimension reduction

Dimension reduction, an essential first step in hyperspectral unmixing, reduces computational complexity, and determines the number of archetypes k . In the linear unmixing model, the number of archetypes k should be much smaller than the number of spectral bands L . Thus, the observed signal \mathbf{Y} belongs to a low-dimensional subspace $k < L$ when projected to the archetype subspace spanned by \mathbf{M} . Dimension reduction aims to determine k , and to estimate the projection of \mathbf{Y} to the k -dimensional subspace. Our study uses Hyperspectral signal subspace identification by minimum error (HySime) to perform dimension reduction on hyperspectral data [25].

Overestimating k increases noise error, while underestimating k increases projection error. Overestimating means that the projection includes unnecessary information from higher dimensions. On the other hand, underestimating means that the projection lacks information from higher dimensions. Determining the optimal k entails minimizing the sum of the noise and projection error [25].

Noise estimation

Multiple regression performs well on hyperspectral data, because of the high correlation between neighboring spectral bands. Multiple regression assumes that the linear

combination of all spectral bands except for the i th band can explain the i th band such that

$$\mathbf{z}_i = \mathbf{Z}_{\partial_i} \boldsymbol{\beta}_i + \boldsymbol{\xi}_i \quad (2.14)$$

where \mathbf{z}_i represents intensities along the i th spectral band (columns of \mathbf{Y}), \mathbf{Z}_{∂_i} contains intensities from all spectral bands except i , $\boldsymbol{\beta}_i$ is the regression vector, and $\boldsymbol{\xi}_i$ is the modeling error vector. HySime's noise estimation calculates $\hat{\boldsymbol{\xi}}_i$, given by

$$\hat{\boldsymbol{\xi}}_i = \mathbf{z}_i - \mathbf{Z}_{\partial_i} \hat{\boldsymbol{\beta}}_i \quad (2.15)$$

The least squares estimator of $\boldsymbol{\beta}_i$ under linear multiple regression is given by

$$\hat{\boldsymbol{\beta}}_i = (\mathbf{Z}_{\partial_i}^T \mathbf{Z}_{\partial_i})^{-1} \mathbf{Z}_{\partial_i}^T \mathbf{z}_i \quad (2.16)$$

Eq. 2.17 overcomes repeatedly calculating $(\mathbf{Z}_{\partial_i}^T \mathbf{Z}_{\partial_i})^{-1}$ for all $i = 1, \dots, L$ by calculating the inverse $(\mathbf{Z}^T \mathbf{Z})^{-1}$ only once.

$$(\mathbf{Z}_{\partial_i}^T \mathbf{Z}_{\partial_i})^{-1} = (\mathbf{Z}^T \mathbf{Z})_{\partial_i, \partial_i}^{-1} - \frac{(\mathbf{Z}^T \mathbf{Z})_{\partial_i, i}^{-1} (\mathbf{Z}^T \mathbf{Z})_{i, \partial_i}^{-1}}{(\mathbf{Z}^T \mathbf{Z})_{i, i}^{-1}} \quad (2.17)$$

where $\mathbf{Z} = \mathbf{Y}^T$. The following table summarizes the meaning of the subscripts in Eq. 2.17.

Table 2.2: Summary of subscripts in Eq. 2.17 for solving $(\mathbf{Z}_{\partial_i}^T \mathbf{Z}_{\partial_i})^{-1}$

subscript	meaning
∂_i, ∂_i	remove i th row and column
∂_i, i	i th column after removing i th row
i, ∂_i	i th row after removing i th column
i, i	i th row and column

HySime estimates the noise $\hat{\boldsymbol{\xi}}_i$ using the observed data \mathbf{Y} by plugging in Eqs 2.16 and 2.17 to Eq. 2.15. The noise correlation matrix is then computed as

$$\hat{\mathbf{R}}_n := 1/N \sum_i \left(\hat{\boldsymbol{\xi}}_i \hat{\boldsymbol{\xi}}_i^T \right) \quad (2.18)$$

Signal Subspace Estimation

Signal subspace estimation entails estimating the number of dimensions spanned by the archetypes and weights $\mathbf{x} = \mathbf{MA}$. Finding the minimum mean square error (MSE) between the original signal \mathbf{x} , and a noisy projection of \mathbf{x} given by $\mathbf{y} = \mathbf{x} + \mathbf{n}$ amounts to estimating the signal subspace. Similar to Eq. 2.18, the signal correlation matrix is estimated as

$$\widehat{\mathbf{R}}_x := 1/N \sum_i \left((\mathbf{y}_i - \widehat{\boldsymbol{\xi}}_i) (\mathbf{y}_i - \widehat{\boldsymbol{\xi}}_i)^T \right) \quad (2.19)$$

where \mathbf{y}_i denotes the rows of \mathbf{Y} , and the noise $\widehat{\boldsymbol{\xi}}_i$ was previously obtained. The sample correlation matrix is estimated as

$$\widehat{\mathbf{R}}_y \equiv (\mathbf{Z}^T \mathbf{Z}) / N \quad (2.20)$$

The final MSE minimization problem between the original signal and its noisy projection is given by

$$(\widehat{k}) = \arg \min_k \left(c + \sum_{j=1}^k \underbrace{\mathbf{e}_{i_j}^T \widehat{\mathbf{R}}_y \mathbf{e}_{i_j} + 2\mathbf{e}_{i_j}^T \widehat{\mathbf{R}}_n \mathbf{e}_{i_j}}_{\delta_{i_j}} \right) \quad (2.21)$$

where c is an irrelevant constant, \mathbf{e}_i are the eigenvectors of $\widehat{\mathbf{R}}_x$, and $k < L$ is the reduced signal subspace dimension. The first term in the summation represents the projection error, while the second term represents the noise error. The index j stands for the j th dimension in the k -dimensional subspace.

The right hand side of Eq. 2.21 can be minimized by considering only the negative values of δ_i for $i = 1, \dots, L$. The number of terms $\delta_i < 0$ dictates the signal subspace dimension k . The eigenvectors \mathbf{e}_i corresponding to the $\delta_i < 0$ arranged in ascending order denote the signal subspace \mathbf{E}_k . To project the observed data into the signal subspace, HySime uses $\mathbf{Y}_k = \mathbf{E}_k \mathbf{E}_k^T \mathbf{Y}$. HySime reduces the L -dimension data by projecting \mathbf{Y} to a subspace with dimension $k < L$ as \mathbf{Y}_k . We use \mathbf{Y}_k instead of \mathbf{Y} in the succeeding steps of hyperspectral unmixing.

2.2.2 Unmixing

The linear unmixing algorithms applied in our study use the geometric approach. In the geometric approach, (1) the vertices of a simplex spanning the data represent the archetypes (illustrated in Fig 2.5), and (2) affine transforming a simplex results to another simplex [6]. Affine transforms preserve the length ratios and parallel relationships of vectors. A simplex generalizes a triangle or a tetrahedron in any dimension. A 0-dimensional simplex is a point; a 1-dimensional simplex is a line; a 2-dimensional simplex is a triangle; and a 3-dimensional simplex is a tetrahedron. A simplex may extend to any number of dimensions depending on the number of archetypes k .

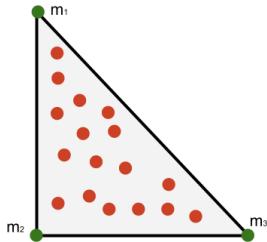


Figure 2.5: Diagram of a 2-dimensional simplex with three archetypes (vertices). Red points are hyperspectral data, while green points are the archetypes.

Unmixing estimates the spectral signature of the archetypes contained in the mixing matrix \mathbf{M} (Eq. 2.13). To estimate \mathbf{M} , geometric unmixing employs minimum volume based algorithms that minimize the volume of the simplex spanning the data [6]. Some geometric unmixing algorithms assume the presence of a pure pixel (illustrated in Fig 2.6). For example, pure pixel algorithms expect that a satellite image contains at least one pixel containing purely trees or purely water.

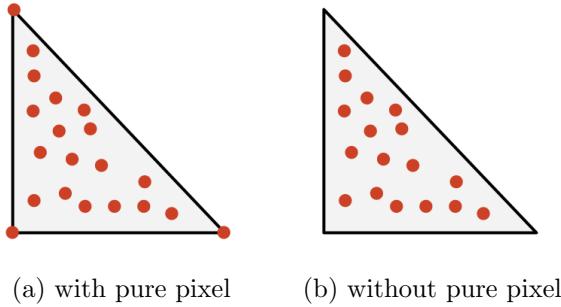


Figure 2.6: Comparison of hyperspectral data (a) with, and (b) without the pure pixel assumption. Data points can be found in the vertices of the simplex in (a), while none in (b).

Vertex Component Analysis

Vertex component analysis (VCA), a pure-pixel algorithm, takes advantage of the two properties of the geometric approach previously mentioned [24]. VCA orthogonally projects data from the subspace spanned by the current archetypes, and sets the extremes of the projection as new archetypes. VCA exhausts all archetypes by repeatedly projecting data. VCA performs better than other pure-pixel algorithms such as pixel purity index (PPI) and N-FINDR [24].

VCA adds a scaling factor γ to Eq. 2.13 that accounts for illumination variability due to surface topography.

$$\mathbf{Y}_k = \mathbf{x} + \mathbf{N} = \mathbf{M}\gamma\mathbf{A} + \mathbf{N} \quad (2.22)$$

The signal \mathbf{x} contained in the set

$$S_x = \{\mathbf{x} \in \Re^L : \mathbf{x} = \mathbf{MA}, \mathbf{1}^T \mathbf{A} = 1, \boldsymbol{\alpha} \succeq 0\} \quad (2.23)$$

belongs to a k -dimensional simplex. However, given that VCA takes into account the scaling factor γ , the observed data now belongs to the convex cone (also a simplex) defined by Eq. 2.24.

$$C_p = \{\mathbf{Y}_k \in \Re^L : \mathbf{Y}_k = \mathbf{M}\gamma\mathbf{A}, \mathbf{1}^T \mathbf{A} = 1, \mathbf{A} \succeq 0, \gamma \geq 0\}, \quad (2.24)$$

When we project C_p into a properly chosen hyperplane (subspace with dimension one less than the surrounding space), we get another simplex whose vertices correspond to the vertices of S_x .

The projection of C_p onto the chosen hyperplane $\mathbf{r}^T \mathbf{u} = 1$ is given by the simplex

$$S_p = \{\mathbf{Y}_p \in \Re^L : \mathbf{Y}_p = \mathbf{Y}_k / (\mathbf{r}_k^T \mathbf{u}), \mathbf{r} \in C_p\} \quad (2.25)$$

In Fig. 2.7, VCA first projects the simplex S_p along the random initial vector \mathbf{f}_1 . Since the point \mathbf{m}_a corresponds to the extreme of the projection (farthest from origin), VCA sets it as the first endmember. In the next iteration, VCA projects the simplex S_p along the vector \mathbf{f}_2 , which is orthogonal to the vector \mathbf{m}_a . VCA finds that the point at the extreme of the projection corresponds to \mathbf{m}_b , and sets it as the next endmember. The next direction \mathbf{f}_3 should then be orthogonal to both \mathbf{m}_a and \mathbf{m}_b .

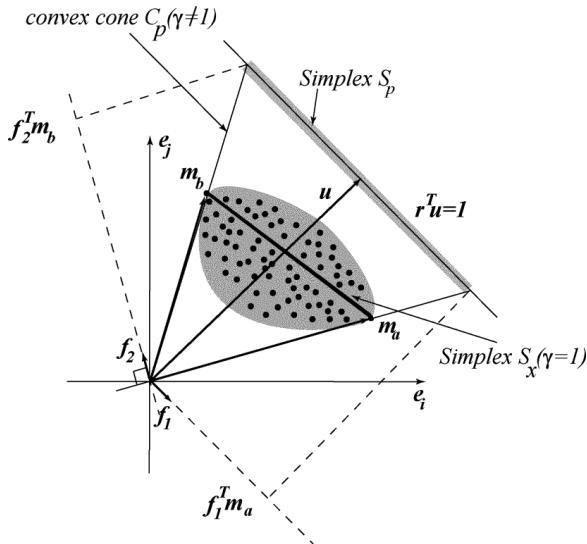


Figure 2.7: Illustration of the VCA algorithm. VCA projects S_p along \mathbf{f}_1 and \mathbf{f}_2 , then sets the extremes as the archetypes. (figure from Ref. [24])

Finding the projection matrix

VCA implements two dimension reduction algorithms to obtain the projection matrix \mathbf{U}_d depending on the signal-to-noise ratio (SNR). If $\text{SNR} > \text{SNR}_{th}$, VCA applies singular value decomposition (SVD), while if $\text{SNR} < \text{SNR}_{th}$, VCA applies principal component analysis (PCA). The threshold is given by

$$\text{SNR}_{th} = 15 + 10 \log_{10}(k) \text{dB} \quad (2.26)$$

For simplicity, SNR is set as $\text{SNR} = 1$ [28]. Given that Eq. 2.26 always exceeds 1, we focus the discussion on PCA. Since VCA projects C_p to a hyperplane, the number of dimensions is set at $d = k - 1$. PCA obtains \mathbf{U}_d from $(\mathbf{Y}_k - \bar{\mathbf{y}}_k)(\mathbf{Y} - \bar{\mathbf{y}}_k)^T/k$,

where $\bar{\mathbf{y}}_k$ denotes the sample mean per column of \mathbf{Y}_k , and $\mathbf{Y}_k \equiv [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k]$. The projection matrix \mathbf{U}_d corresponds to \mathbf{a} in Section 2.1.1.

To project C_p into the d -dimensional hyperplane, we use $\mathbf{X}_p = \mathbf{U}_d^T(\mathbf{Y}_k - \bar{\mathbf{y}}_k)$. In the original dimension k , the simplex S_p is given by

$$S_p := \begin{bmatrix} \mathbf{X}_p \\ \mathbf{c} \end{bmatrix} \quad (2.27)$$

where \mathbf{c} is a $1 \times N$ vector with all elements given by $c := \arg \max_{j=1 \dots N} \|[\mathbf{X}_p]_{:,j}\|$. The index $:,j$ indicates the j th column of the matrix \mathbf{X}_p .

Estimating archetype signatures

VCA initializes the matrix containing the archetype signatures as

$$\mathbf{M} := [\mathbf{e}_u | \mathbf{0} | \dots | \mathbf{0}] \quad (2.28)$$

where \mathbf{M} is a $k \times k$ matrix, and $\mathbf{e}_w = [0, \dots, 0, 1]^T$. Then, VCA sets a random vector \mathbf{f} orthonormal to the columns of \mathbf{M} given by

$$\mathbf{f} := ((\mathbf{I} - \mathbf{M}\mathbf{M}^\#) \mathbf{w}) / (\|(\mathbf{I} - \mathbf{M}\mathbf{M}^\#) \mathbf{w}\|) \quad (2.29)$$

where \mathbf{w} is a zero-mean random Gaussian $k \times 1$ vector, and the operation $\mathbf{M}^\#$ stands for the pseudoinverse of \mathbf{M} . To project S_p along \mathbf{f} , we use $\mathbf{v} = \mathbf{f}^T S_p$. Matrix \mathbf{M} stores the pixel spectra corresponding to the data point with the highest $|\mathbf{v}|$ as the next endmember. VCA exhausts the number of archetypes k by iterating all steps starting from Eq. 2.29 k times. The final matrix \mathbf{M} corresponds to the estimated mixing matrix containing the pure pixels in the scene.

Minimum Volume Simplex Analysis

Unlike VCA, minimum volume simplex analysis (MVSA) [26] assumes the absence of pure pixels, a more realistic scenario. MVSA employs soft constraints that allow violations to the positivity constraint [6]. Since MVSA allows negative weights, MVSA can estimate archetypes closer to the true simplex when analysing noisy data (illustrated in Fig. 2.8). To efficiently solve the volume maximization problem, MVSA implements a sequence of quadratically constrained subproblems using the interior point method [26]. This allows MVSA to unmix large and complex data.

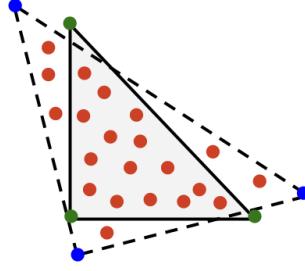


Figure 2.8: Sample of noisy data where some data points lie outside the true simplex (solid triangle). The incorrect simplex (dashed triangle) estimates archetypes (blue points) far from the true values (green points).

MVSA Preprocessing

Similar to VCA, MVSA also takes into account pixel-dependent scale factors such as spectral variability (variations in surface topography) that violate the sum-to-one constraint. MVSA removes the effect of pixel-dependent scale factors by projecting \mathbf{Y}_k onto the hyperplane that best represents \mathbf{Y}_k in the least squares sense. The orthogonal projection of the vectors of \mathbf{Y}_k is given by

$$\mathbf{y}_{p,i} \leftarrow \bar{\mathbf{y}} + \mathbf{E}_{p-1}^T (\mathbf{y}_{k,i} - \bar{\mathbf{y}}), \quad i \in \{1, 2, \dots, N\} \quad (2.30)$$

where the columns of \mathbf{E}_{p-1} contain the eigenvalues of the sample covariance matrix $(\mathbf{Y}_k - \bar{\mathbf{Y}}_k)(\mathbf{Y}_k - \bar{\mathbf{Y}}_k)^T/N$, $\mathbf{y}_{k,i}$ denote the rows of \mathbf{Y}_k , $\mathbf{y}_{p,i}$ denote the rows of \mathbf{Y}_p , and

$$\bar{\mathbf{y}} \equiv \frac{1}{N} \sum_{i=1}^N \mathbf{y}_{k,i}; \quad \bar{\mathbf{Y}}_k = \bar{\mathbf{y}} \mathbf{1}_N^T \quad (2.31)$$

From here on, we write the orthogonal projection of \mathbf{Y}_k as \mathbf{Y}_p .

MVSA inference criterion

MVSA assumes that the noise can be neglected after data projection. Thus, the minimum volume simplex problem can be written as

$$\begin{aligned} \widehat{\mathbf{M}} &= \arg \min_{\mathbf{M}} |\det(\mathbf{M})| \\ \text{s.t.: } \mathbf{Q}\mathbf{Y}_p &\geq 0, \quad \mathbf{1}_k^T \mathbf{Q}\mathbf{Y}_p = \mathbf{1}_N^T \end{aligned} \quad (2.32)$$

where $\mathbf{Q} \equiv \mathbf{M}^{-1}$. Since $|\det(\mathbf{M})|$ represents the volume of the simplex spanned by the archetypes, Eq. 2.32 estimates \mathbf{M} corresponding to the minimum simplex volume that satisfies the constraints.

Rewriting Eq. 2.32 in terms of \mathbf{Q} , we get

$$\widehat{\mathbf{Q}} = \arg \max_{\mathbf{Q}} \log |\det(\mathbf{Q})| \quad (2.33)$$

since $\det(\mathbf{Q}) = 1/\det(\mathbf{M})$. The constraints remain unchanged.

Constraint Reduction

MVSA assumes that \mathbf{Y}_p has linearly independent rows and/or columns (depending on its size), which means $\mathbf{Y}_p \mathbf{Y}_p^T$ can be inverted. Thus, we can rewrite the sum-to-one constrain in Eq. 2.32 as

$$\mathbf{1}_k^T \mathbf{Q} \mathbf{Y}_p = \mathbf{1}_N^T \Leftrightarrow \mathbf{1}_k^T \mathbf{Q} = \mathbf{q}_k \quad (2.34)$$

where $\mathbf{q}_k \equiv \mathbf{1}_N^T \mathbf{Y}_p^T (\mathbf{Y}_p \mathbf{Y}_p^T)^{-1}$. The new optimization problem reduces the number of equality constraints from kN to k .

Minorize-maximization optimization

MVSA adopts the minorize-maximization framework that builds a minorizer of the objective function, and iteratively maximizes it [29]. In our case, the objective function is given by $f(\mathbf{x}) \equiv \log |\det(\mathbf{Q})|$, and the iterative procedure is given by

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \arg \max_{\mathbf{x}} \phi(\mathbf{x}; \mathbf{x}^{(t)}) \\ \text{s.t.: } \mathbf{A}_I \mathbf{x} &\geq \mathbf{b}_I, \quad \mathbf{A}_E \mathbf{x} = \mathbf{b}_E \end{aligned} \quad (2.35)$$

where

$$\begin{aligned} \mathbf{A}_I &\equiv (\mathbf{Y}_p^T \otimes \mathbf{I}) \in \mathbb{R}^{kN \times k^2} \\ \mathbf{A}_E &\equiv (\mathbf{I} \otimes \mathbf{1}_k^T) \in \mathbb{R}^{k^2 \times k^2} \\ \mathbf{b}_I &\equiv \mathbf{0} \in \mathbb{R}^{kN} \end{aligned} \quad (2.36)$$

$$\mathbf{b}_E \equiv \mathbf{q}_p \in \mathbb{R}^p$$

In Eq. 2.35, we let $\mathbf{x} \equiv \text{vec}(\mathbf{Q})$, which stacks the columns of \mathbf{Q} into one column of \mathbf{x} . The symbol \otimes denotes the Kronecker operator, $\phi(\mathbf{x}; \mathbf{x}^{(t)})$ denotes the minorizer of $f(\mathbf{x})$, and t denotes the iteration index. MVSA uses a quadratic function as the minorizer for f .

$$\phi(\mathbf{x}; \mathbf{x}^{(t)}) \equiv f(\mathbf{x}^{(t)}) + \mathbf{c}^{(t)T} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{G}^{(t)} \mathbf{x} \quad (2.37)$$

where

$$\begin{aligned}\mathbf{c}^{(t)} &\equiv \mathbf{g}^{(t)} - \mathbf{G}^{(t)}\mathbf{x}^{(t)} \\ \mathbf{G} &\equiv \min \{\lambda_{\min}(\mathbf{H}), -v\} \mathbf{I}.\end{aligned}\tag{2.38}$$

In Eq. 2.38, $\mathbf{g}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x})$ denote the gradient and Hessian (matrix of second-order partial derivatives) of f , respectively. The notation $\lambda_{\min}(\mathbf{H})$ stands for the smallest eigenvalue of \mathbf{H} , and v is a small positive number. To maximize the minorizer function $\phi(\mathbf{x}; \mathbf{x}^{(t)})$, MVSA's core step in the algorithm solves the following convex optimization problem

$$\begin{aligned}\max \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} \\ \text{s.t.: } \mathbf{A}_I \mathbf{x} \geq \mathbf{b}_I, \quad \mathbf{A}_E \mathbf{x} = \mathbf{b}_E\end{aligned}\tag{2.39}$$

that evolved from the nonconvex optimization problem in Eq. 2.33. Eq. 2.39 combines Eqns 2.35 and 2.37.

Interior Point Method

MVSA uses the interior-point method to solve Eq. 2.39. To begin, we write the Karush-Kuhn-Tucker (KKT) conditions—conditions to be satisfied by the solution of a nonlinear optimization problem [30]—of Eq. 2.39 as follows

$$\begin{aligned}\mathbf{G} \mathbf{x} - \mathbf{A}_I^T \boldsymbol{\lambda} + \mathbf{A}_E^T \boldsymbol{\mu} + \mathbf{c} &= 0 \\ \mathbf{A}_I \mathbf{x} - \mathbf{b}_I &\geq 0 \\ \mathbf{A}_E \mathbf{x} - \mathbf{b}_E &= 0 \\ (\mathbf{A}_I \mathbf{x} - \mathbf{b}_I)_i \lambda_i &= 0, i = 1 \dots n_I \equiv Nk \\ \boldsymbol{\lambda} &\geq 0\end{aligned}\tag{2.40}$$

where $\boldsymbol{\lambda} \equiv [\lambda_1, \dots, \lambda_{n_I}]^T$, and $\boldsymbol{\mu} \in \mathbb{R}^p$ denote the Lagrange multipliers of the inequality and equality constraints, respectively, and n_I denotes the number of inequality constraints. The first equation represents the maximizing stationarity condition; the second and third represent the primal feasibility conditions; the fourth represents the complementary slackness condition; and the last represents the dual feasibility condition. By introducing a slack vector $\mathbf{s} \equiv [s_1, \dots, s_{n_I}]^T$, Eq. 2.40 can be rewritten such that the interior point method can solve it. When added/subtracted to, slack variables/vectors convert inequality constraints to equality constraints [30].

$$\begin{aligned}
\mathbf{G}\mathbf{x} - \mathbf{A}_I^T \boldsymbol{\lambda} + \mathbf{A}_E^T \boldsymbol{\mu} + \mathbf{c} &= 0 \\
\mathbf{A}_I \mathbf{x} - \mathbf{s} - \mathbf{b}_I &= 0 \\
\mathbf{A}_E \mathbf{x} - \mathbf{b}_E &= 0 \\
s_i \lambda_i &= 0, i = 1 \dots n_I \\
\boldsymbol{\lambda}, \mathbf{s} &\geq 0
\end{aligned} \tag{2.41}$$

The interior-point method applies Newton's method to calculate the roots in the systems of equations from Eq. 2.41. Newton's method for nonlinear systems of equations uses the following equation

$$J_F(\mathbf{x}_n)(\mathbf{x}_{n+1} - \mathbf{x}_n) = -F(\mathbf{x}_n) \tag{2.42}$$

where $J_F(\mathbf{x}_n)$ is the Jacobian matrix (of first-order partial derivatives) for $F(\mathbf{x}_n)$ [30]. In our case, $F(\mathbf{x}_n)$ denotes the left-hand side of Eq. 2.41, and the unknown variables \mathbf{x}_n denote \mathbf{x} , $\boldsymbol{\mu}$, \mathbf{s} , and $\boldsymbol{\lambda}$. MVSA's predictor corrector interior point algorithm applies Newton's method twice: first, to get the affine step, then second, to correct the affine step. Newton's method applied to the affine step is given by

$$\begin{bmatrix} \mathbf{G} & \mathbf{A}_E^T & \mathbf{0} & -\mathbf{A}_I^T \\ \mathbf{A}_I & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{A}_E & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{\text{aff}} \\ \Delta \boldsymbol{\mu}^{\text{aff}} \\ \Delta \mathbf{s}^{\text{aff}} \\ \Delta \boldsymbol{\lambda}^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_d \\ -\mathbf{r}_I \\ -\mathbf{r}_E \\ -\boldsymbol{\Lambda} \mathbf{Se} \end{bmatrix} \tag{2.43}$$

where

$$\begin{aligned}
\mathbf{r}_d &\equiv \mathbf{G}\mathbf{x} - \mathbf{A}_I^T \boldsymbol{\lambda} + \mathbf{A}_E^T \boldsymbol{\mu} + \mathbf{c} \\
\mathbf{r}_I &\equiv \mathbf{A}_I \mathbf{x} - \mathbf{s} - \mathbf{b}_I \\
\mathbf{r}_E &\equiv \mathbf{A}_E \mathbf{x} - \mathbf{b}_E \\
\boldsymbol{\Lambda} &\equiv \text{diag}(\lambda_1, \dots, \lambda_{n_I}) \\
\mathbf{S} &\equiv \text{diag}(s_1, \dots, s_{n_I}) \\
\mathbf{e} &\equiv [1, \dots, 1]^T
\end{aligned} \tag{2.44}$$

Newton's method applied again to the affine step for correction is given by

$$\begin{bmatrix} \mathbf{G} & \mathbf{A}_E^T & \mathbf{0} & -\mathbf{A}_I^T \\ \mathbf{A}_I & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{A}_E & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\mu} \\ \Delta \mathbf{s} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_d \\ -\mathbf{r}_I \\ -\mathbf{r}_E \\ -\boldsymbol{\Lambda} \mathbf{Se} - \Delta \boldsymbol{\Lambda}^{\text{aff}} \Delta \mathbf{S}^{\text{aff}} \mathbf{e} + \sigma \rho \mathbf{e} \end{bmatrix} \tag{2.45}$$

where

$$\begin{aligned}
\Delta\Lambda^{\text{aff}} &\equiv \text{diag}(\Delta\lambda_1^{\text{aff}}, \dots, \Delta\lambda_{n_I}^{\text{aff}}) \\
\Delta\mathbf{S}^{\text{aff}} &\equiv \text{diag}(\Delta s_1^{\text{aff}}, \dots, \Delta s_{n_I}^{\text{aff}}) \\
\rho &\equiv \frac{\mathbf{s}^T \boldsymbol{\lambda}}{n_I} \\
\sigma &= \left(\frac{\rho_{\text{aff}}}{\rho} \right)^3
\end{aligned} \tag{2.46}$$

The quantity ρ , known as the duality measure, measures the average value of the pairwise products $s_i\lambda_i$ [30]. MVSA implements a less aggressive approach to Newton's method such that $s_i\lambda_i$ converges to a small average value $s_i\lambda_i = \sigma\rho$, instead of at 0. The quantity $\sigma \in [0, 1]$, known as the centering parameter, dictates how strict the algorithm applies the condition $s_i\lambda_i = 0$. A lower centering parameter produces systems of equations closer to the affine Newton step in Eq. 2.43.

Normal Equations

To ease the computational difficulty of solving the system of equations 2.43 and 2.45, we reformulate them to more compact symmetric coefficient matrices. Given that $\boldsymbol{\Lambda}$ and \mathbf{S} are nonsingular matrices (i.e. matrices whose inverse exist), we can multiply the fourth equation in Eqs. 2.43 and 2.45 by \mathbf{S}^{-1} , and isolate $\Delta\boldsymbol{\lambda}$. We get

$$\Delta\boldsymbol{\lambda} = -\mathbf{S}^{-1}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\mathbf{r}_{\Lambda S} + \Delta\mathbf{s}) \tag{2.47}$$

where $\mathbf{r}_{\Lambda S}$ represents the fourth element in the right hand side of Eq. 2.43 and 2.45. We can get $\Delta\mathbf{s}$ from the second equation.

$$\Delta\mathbf{s} = \mathbf{A}_I\Delta\mathbf{x} + \mathbf{r}_I \tag{2.48}$$

The third equation remains unchanged.

$$\mathbf{A}_E\Delta\mathbf{x} = -\mathbf{r}_E \tag{2.49}$$

By plugging in $\Delta\boldsymbol{\lambda}$ from Eq. 2.47 to the first equation, we get

$$(\mathbf{G} + \mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I)\Delta\mathbf{x} + \mathbf{A}_E^T\Delta\boldsymbol{\mu} = -\mathbf{r}_d + \mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}(-\mathbf{r}_I - \boldsymbol{\Lambda}^{-1}\mathbf{r}_{\Lambda S}) \tag{2.50}$$

Eqs 2.47-2.50 are known as the “normal equations,” of the systems of equations 2.43 and 2.45 [30].

Solving the normal equations

By combining Eqs 2.49 and 2.50, we can formulate a $(k^2+k) \times (k^2+k)$ linear system,

$$\Delta\mathbf{m} = \begin{bmatrix} \mathbf{K} & \mathbf{A}_E^T \\ \mathbf{A}_E & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{r}_h \\ -\mathbf{r}_E \end{bmatrix} \quad (2.51)$$

where \mathbf{r}_h denotes the right hand side of Eq 2.50, and $\mathbf{K} = \mathbf{G} + \mathbf{A}_I^T \mathbf{S}^{-1} \boldsymbol{\Lambda} \mathbf{A}_I$. From Eq. 2.51, we can solve for $\Delta\mathbf{x}$ and $\Delta\boldsymbol{\mu}$ as $\Delta\mathbf{x} = \Delta\mathbf{m}[:k^2]$, and $\Delta\boldsymbol{\mu} = \Delta\mathbf{m}[k^2+1:]$. We used the array index notation used by *Python* here.

MVSA first applies the normal equations on the affine Newton step in Eq. 2.43 to solve for $\Delta\mathbf{x}^{\text{aff}}$, $\Delta\boldsymbol{\mu}^{\text{aff}}$, $\Delta\mathbf{s}^{\text{aff}}$, and $\Delta\boldsymbol{\lambda}^{\text{aff}}$. Based on Eqs 2.43 and 2.45, $\Delta\mathbf{x}^{\text{aff}} = \Delta\mathbf{x}$, and $\Delta\boldsymbol{\mu}^{\text{aff}} = \Delta\boldsymbol{\mu}$. We can then use $\Delta\mathbf{x}$ to solve for $\Delta\mathbf{s}$ in Eq. 2.48, and use $\Delta\mathbf{s}$ to solve for $\Delta\boldsymbol{\lambda}$ in Eq. 2.47. The initial values for the unknown variables are $\boldsymbol{\mu} = \mathbf{s} = \boldsymbol{\lambda} = [1, \dots, 1]^T$, while the estimated mixing matrix obtained from VCA give the initial values for \mathbf{x} .

MVSA iterates through the interior point method until MVSA reaches a stopping criterion. When $\sigma < 10^{-8}$ and $\rho < 10^{-8}$, the iterations stop, and the final solution \mathbf{x} for Eq. 2.39 is obtained [26]. Recall that $\mathbf{x} \equiv \text{vec}(\mathbf{Q})$, and $\mathbf{Q} \equiv \mathbf{M}^{-1}$. Thus, we have an estimate for the final mixing matrix

$$\mathbf{M} = [\text{vec}^{-1}(\mathbf{x})]^{-1} \quad (2.52)$$

2.2.3 Inversion

Inversion estimates the archetype weights contained in the abundance matrix \mathbf{A} . Spectral unmixing by splitting and augmented Lagrangian (SUUnSAL) unmixes \mathbf{A} through sparse regression [27]. Sparse regression fits the observed hyperspectral data (\mathbf{Y} in Eq. 2.13) with the known spectral signatures of the archetypes \mathbf{M} to estimate \mathbf{A} . SUUnSAL was formulated based on the alternating direction method of multipliers (ADMM), which breaks down complicated convex optimization problems into simpler ones.

SUUnSAL solves the following general constrained sparse regression problem

$$\begin{aligned} & \min_{\mathbf{A}} (1/2) \|\mathbf{MA} - \mathbf{Y}_k\|_2^2 + \lambda \|\mathbf{A}\|_1 \\ & \text{subject to: } \mathbf{A} \geq \mathbf{0}, \quad \mathbf{1}^T \mathbf{A} = 1 \end{aligned} \quad (2.53)$$

The operators $\|\dots\|_1$ and $\|\dots\|_2$ stand for the Manhattan and Euclidean norms, respectively, and $\lambda \geq 0$ controls the relative weight between the two norms.

Alternating direction method of multipliers

Consider an unconstrained problem with the form

$$\min_{\mathbf{A} \in \mathbb{R}^n} f_1(\mathbf{A}) + f_2(\mathbf{G}\mathbf{A}) \quad (2.54)$$

where \mathbf{G} has linearly independent rows and columns, and f_1 and f_2 are closed proper convex functions. A proper convex function $f(x)$ satisfies the following conditions: (1) $f(x) < +\infty$ for at least one x , and (2) $f(x) > -\infty$ for every x [31]. By applying ADMM, the convergence of the sequence \mathbf{A}_k solves Eq. 2.54.

$$\mathbf{A}_{k+1} = \arg \min_{\mathbf{A}} f_1(\mathbf{A}) + \frac{\mu}{2} \|\mathbf{G}\mathbf{A} - \mathbf{u}_k - \mathbf{d}_k\|_2^2 \quad (2.55)$$

$$\mathbf{u}_{k+1} = \arg \min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2} \|\mathbf{G}\mathbf{A}_{k+1} - \mathbf{u} - \mathbf{d}_k\|_2^2 \quad (2.56)$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\mathbf{A}_{k+1} - \mathbf{u}_{k+1}) \quad (2.57)$$

where $\mu > 0$ is some arbitrary constant.

ADMM for SUNSAL

By rewriting Eq. 2.53 so that we can translate it to Eq. 2.54, we have

$$\min_{\mathbf{A}} (1/2)\|\mathbf{M}\mathbf{A} - \mathbf{Y}_k\|_2^2 + \lambda\|\mathbf{A}\|_1 + \iota_{\{1\}}(\mathbf{1}^T \mathbf{A}) + \iota_{\mathbb{R}_+^n}(\mathbf{A}) \quad (2.58)$$

where $\iota_{\{s\}}(\mathbf{x})$ is the indicator function given by

$$\iota_{\{s\}}(\mathbf{A}) = \begin{cases} 0, & \mathbf{A} \in S \\ \infty, & \mathbf{A} \notin S \end{cases} \quad (2.59)$$

In terms of the ADMM problem, Eq. 2.58 can be interpreted as

$$f_1(\mathbf{M}) \equiv \frac{1}{2}\|\mathbf{M}\mathbf{A} - \mathbf{Y}_k\|_2^2 + \iota_{\{1\}}(\mathbf{1}^T \mathbf{A}) \quad (2.60)$$

$$f_2(\mathbf{A}) \equiv \lambda\|\mathbf{A}\|_1 + \iota_{\mathbb{R}_+^n}(\mathbf{A}) \quad (2.61)$$

$$\mathbf{G} \equiv \mathbf{I} \quad (2.62)$$

The solution of Eq. 2.55 is then, given by

$$\mathbf{A}_{k+1} \leftarrow \mathbf{B}^{-1}\mathbf{w} - \mathbf{C} (\mathbf{1}^T \mathbf{B}^{-1} \mathbf{w} - 1) \quad (2.63)$$

where

$$\mathbf{B} \equiv \mathbf{M}^T \mathbf{M} + \mu \mathbf{I} \quad (2.64)$$

$$\mathbf{C} \equiv \mathbf{B}^{-1} \mathbf{1} (\mathbf{1}^T \mathbf{B}^{-1} \mathbf{1})^{-1} \quad (2.65)$$

$$\mathbf{w} \equiv \mathbf{M}^T \mathbf{Y}_k + \mu (\mathbf{u}_k + \mathbf{d}_k) \quad (2.66)$$

We can rewrite Eq. 2.56 as

$$\mathbf{u}_{k+1} \leftarrow \arg \min_{\mathbf{u}} (1/2) \|\mathbf{u} - \boldsymbol{\nu}_k\|_2^2 + (\lambda/\mu) \|\mathbf{u}\|_1 + \iota_{\mathbb{R}_+^n}(\mathbf{u}) \quad (2.67)$$

where $\boldsymbol{\nu}_k \equiv \mathbf{A}_{k+1} - \mathbf{d}_k$. To solve Eq. 2.67, we use

$$\mathbf{u}_{k+1} \leftarrow \max \{ \mathbf{0}, \text{soft}(\boldsymbol{\nu}_k, \lambda/\mu) \} \quad (2.68)$$

where $\text{soft}(\boldsymbol{\nu}_k, \lambda/\mu) = \text{sign}(\boldsymbol{\nu}_k) (|\boldsymbol{\nu}_k| - \lambda/\mu)_+$, and

$$(x)_+ = \begin{cases} x, & x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.69)$$

SUnSAL iterates through Eqs 2.63, 2.68, and 2.57 until \mathbf{A} converges [27]. The convergence criterion can be set as either a maximum number of iterations, or as a threshold value for the difference between previous and current values of \mathbf{A} .

Chapter 3

Methodology

In this chapter, we discuss (1) how we handled the Philippine senate elections dataset (Section 3.1), (2) how we performed K-means clustering on the candidates' cumulative votes distribution (Section 3.2), and (3) how we performed hyperspectral unmixing on the precinct-level senate elections results (Section 3.3).

3.1 Election dataset

The results for the 2013 and 2016 Philippine elections were obtained from elections.org.ph. The results of the 2019 Philippine were scraped from the Commission on Elections (COMELEC) website. Table 3.1 summarizes the timestamps of the election results we used, and Fig. 3.1 shows sample data from the election results.

Table 3.1: Summary of timestamps of election results we used per year

	date	time	% of total results
2013	05/18	2:21 pm	76%
2016	05/12	3:45 pm	96%
2019	<i>not available</i>		98%

Clustered precincts with 0 ballot counts in 2013 and 2019 correspond to precincts whose results have not yet reached counting centers, so we disregarded them from analysis. We used the necessary information from the datasets for analysis based on Table 3.2.

	Region	Province	Municipality/City	Barangay	Precinct Code	Total Registered Voters	ALCANTARA, SAMSON (SJS)	ANGARA, EDGARDO (LDP)	AQUINO, BENIGNO BAM (LP)	BELGICA, GRECO (DPP)
0	Region I	ILOCOS NORTE		ADAMS	ADAMS (POB.)	402890	0.0	0	0	0
1	Region I	ILOCOS NORTE		ADAMS	ADAMS (POB.)	402891	0.0	0	0	0

(a) 2013

clustered_precinct	position_code	candidate	party	votes	ballot_order	total_ballots	H	I	date_time
23	10030074	399009	SOTTO, VICENTE (NPC)	99	199	46	535	2318	8 5/9/2016 17:07
27	10030074	399009	MONTANO, ALLAN (UNA)	163	23	31	535	2318	8 5/9/2016 17:07
30	10030074	399009	GADON, LARRY (KBL)	72	25	16	535	2318	8 5/9/2016 17:07

(b) 2016

bo	ser	tot	v	vbc	ballot_count
0	1	HPM180PA01053537	2353	8 54779 [1, number-of-voters-who-actually-voted, 418.0]	
1	2	HPM180PA01053537	2353	13 54779 [1, number-of-voters-who-actually-voted, 418.0]	

(c) 2019

Figure 3.1: Sample election data for every year

Table 3.2: Equivalent column labels of election data for necessary information

	2013	2016	2019
ballot count	Total Registered Voters	total_ballots	ballot_count
votes per candidate	<i>found in column 6 onwards</i>	votes	v
precinct code	Precinct Code	precinct_code	vbc

Each precinct code in the 2019 elections dataset correspond to 62 votes. The 62 votes correspond to the 62 candidates arranged by their alphabetical last names.

Converting to hyperspectral data

We created a matrix of vote fractions per year that represents \mathbf{Y} in Eq. 2.13. The columns correspond to each senate candidate arranged from the candidate with the most overall votes to the least. The rows correspond to each clustered precinct in no particular order. The vote fraction of a candidate per precinct is given by

$$\text{fraction of votes in precinct} = \frac{\text{total \# of votes}}{\text{total \# of ballots}} \quad (3.1)$$

3.2 Clustering candidates

We performed K-means clustering using features from the cumulative distribution of candidate votes three times, one for each year.

Cumulative distribution of votes

We created a histogram with 60 bins from the vote fractions for each candidate. From the histogram, we calculated the cumulative distribution of votes, where the 60 cumulative vote fractions (from the 60 bins) served as the 60 features of each candidate for clustering. Thus, the matrix of features per candidate is given by

$$\mathbf{V} = \{v_{ij}\} = \{CDF_i^{-1}(p_j)\}, \quad \text{where } p_j = \frac{j}{N} \quad (3.2)$$

where i denotes the i th candidate, $j = 1, 2, \dots, N$ denotes the j th feature, and $N = 60$ is the total number of features. Each candidate corresponds to one data point in clustering.

3.2.1 Dimension reduction using PCA

Before clustering, we reduced the feature dimensions using principal component analysis (PCA). From Eqn 2.1, the principal components (PCs) of \mathbf{V} are given by

$$\sum_{j=1}^N a_j \mathbf{v}_j = \mathbf{Va} \quad (3.3)$$

From Eqn 2.2, the variance explained by each principal component is given by

$$var(\mathbf{Va}) = \mathbf{a}^T \mathbf{S} \mathbf{a} \quad (3.4)$$

where \mathbf{S} is the covariance matrix of \mathbf{V} .

PCA was implemented using `scikit-learn's decomposition` package in Python. The `PCA` function was used to calculate the PCs based on Eqn 3.3, and the explained variance ratio given by Eq. 2.7. We applied the elbow method to the plot of the cumulative explained variance ratio vs number of PCs to choose how many PCs should remain after dimension reduction.

3.2.2 K-means clustering on candidates' cumulative vote distributions

K-means clustering was implemented using the `KMeans` function under `scikit-learn's cluster` package in Python. The `KMeans` function was applied to \mathbf{V} , and was used to calculate the overall MSE based on Eqn 2.11 for different values of K from 1-15. The elbow method was applied to the plot of the overall MSE vs values of K (number of

clusters) to choose the optimal number of clusters. However, due to the subjectivity of the elbow method, we also used silhouette analysis for added accuracy.

Silhouette analysis was implemented using `scikit-learn`'s `metrics` package in Python. We used two functions from the package: `silhouette_samples` and `silhouette_score`. The former calculates $s(i)$ for each data point, while the latter calculates $\bar{s}(i)$. We first looked at the most probable values of K based on the elbow method, then we performed silhouette analysis on a smaller range of values ($K = 2$ to $K = 5$). We used two criteria for choosing K : $\bar{s}(i)$, and the number of data points with $s(i) \geq \bar{s}(i)$.

3.3 Hyperspectral unmixing on election results

We used the MatLab code for HySime, MVSA, VCA, and SuNSAL in Ref. [26] to perform unmixing on the election results. HySime dictates how many voting archetypes optimally characterize each year. VCA and MVSA estimate the vote spectra **M** of each archetype. SuNSAL fits these vote spectra with **Y** to estimate their corresponding weights **A** per clustered precinct.

The candidates with high vote intensities per archetype spectra hold more information for analysis than the exact values themselves. Thus, we can normalize the archetypes to maintain uniformity. The maximum vote intensity per archetype was calculated and assigned as the normalization constants **n**. By dividing **M** and multiplying **A** by **n**, we normalize the archetype intensities while still following Eq. 2.13. We labeled each archetype according to their overall mean weight calculated from **nA** (i.e. archetype 1 has the highest overall mean weight, and so on).

Categorizing archetypes

To determine the voting pattern per archetype, we correlated candidates with high vote intensities—candidates with vote intensities higher than the archetype mean plus one standard deviation. Their correlations were based on personal profiles and news articles, which are all publicly available online. A candidate's personal profile covers their province, religion, stance on social issues, party alignment, profession, and name recall. Name recall refers to media popularity and/or political history. By comparing the voting archetypes in 2013, 2016, and 2019, we can determine which patterns persist and which patterns evolve.

3.3.1 Finding dominant archetypes

We grouped the clustered precincts in each year by province, except for the 2016 precincts. They were grouped by region, because that was the highest location resolution available in the dataset. The 2013 election dataset includes the province of each clustered precinct (in the column labeled *Province* in Fig. 3.1a). We only need the columns labeled *region_name* and *precincts_code* for analysis in the 2016 elections results per region (Fig. 3.2).

item	region_name	precincts_code	numvotes	overvotes	undervotes	rv	datestamp	precinct_reported	accumulated_undervotes	accumulated_overvotes	rv_overtime	
0	1	ARMM	66010030	391	1	3583	395	05/09/2016 17:01	0.001061	3583	1	395
1	2	REGION IX	72190014	205	2	1308	679	05/09/2016 17:01	0.002121	4891	3	1074

Figure 3.2: 2016 election results per region. Our analysis uses the first two columns to determine each precinct's region.

For each province/region, we obtained the mean and median of the normalized weights \mathbf{nA} per archetype. The archetype with the highest mean or median was deemed the dominant archetype. Since precincts have different total number of ballots, one precinct may hold more weight than another. To account for this, we multiplied the normalized weights by the total number of ballots per clustered precinct.

We visualized the dominant archetypes through provincial/regional map shapefiles of the Philippines from <http://philgis.org> (sample show in Fig. 3.3). By adding new columns containing the dominant archetypes per province/region, we can plot the map by color, where each color represents each archetype.

ID_0	ISO	NAME_0	ID_1	NAME_1	NL_NAME_1	VARNAME_1	TYPE_1	ENGTYPE_1	PROVINCE	REGION	geometry
0	177	PHL	Philippines	1	Abra	None	None	Lalawigan Probinsya	Province	Abra	Cordillera Administrative Region (CAR)
1	177	PHL	Philippines	2	Agusan del Norte	None	None	Lalawigan Probinsya	Province	Agusan del Norte	Caraga (Region XIII)

(a) province

	REGION	geometry	REGION NUMBER
0	Autonomous Region of Muslim Mindanao (ARMM)	(POLYGON ((119.4669418334962 4.586939811706523...))	ARMM
1	Bicol Region (Region V)	(POLYGON ((122.9841690063476 11.71055984497071...))	Region V

(b) region

Figure 3.3: Sample shapefiles of Philippine maps. We used (a) for 2013 and 2019, and (b) for 2016.

PhilGIS uses different province labels from the election data we used, so we modified the election labels according to Table 3.3.

Table 3.3: Modified provincial labels of the 2013 and 2019 election results to match the PhilGIS map labels.

election results	PhilGIS provincial map
NATIONAL CAPITAL REGION - FOURTH DISTRICT	METROPOLITAN MANILA
NATIONAL CAPITAL REGION - SECOND DISTRICT	
NATIONAL CAPITAL REGION - THIRD DISTRICT	
NATIONAL CAPITAL REGION - MANILA	
DAVAO (DAVAO DEL NORTE)	DAVAO DEL NORTE
SAMAR (WESTERN SAMAR)	SAMAR
COTABATO (NORTH COT.)	NORTH COTABATO
SHARIFF KABUNSUAN	MAGUINDANAO

The region names also differ, so we renamed the regions in Fig. 3.3b by the region numbers written in the parentheses (ex. Bicol Region (Region V) → Region V).

Chapter 4

Results and Discussion

In this chapter, we discuss (1) the candidate clusters after K-means clustering (Section 4.1), (2) the voting archetypes after hyperspectral unmixing (Section 4.2), and (3) how our results compare to studies that use election surveys (Section 4.3).

4.1 K-means clustering of candidates

We first look at the possible voting patterns extracted by K-means clustering.

Reduced dimensions after PCA

The elbow method suggests three principal components (PC) as the optimal dimensions for all years. However, for simpler visualization of plotted clusters, we only considered two PCs. Two PCs can already account for at least 96% of the overall variance, and adding a 3rd PC contributes significantly less variance (Table 4.1).

Table 4.1: Cumulative explained variance ratio for 1PC, 2PCs, and 3PCs

	1 PC	2 PCs	3 PCs
2013	87.55%	96.65%	98.92%
2016	90.03%	96.66%	98.65%
2019	91.68%	98.28%	99.62%

The top 16 candidates possess high positive values in the first PC in all years (Fig. 4.1). Along the second PC, the top 15-20 candidates from 2013 and 2016 possess high positive values, while the top 15-25 candidates from 2019 possess high negative values.

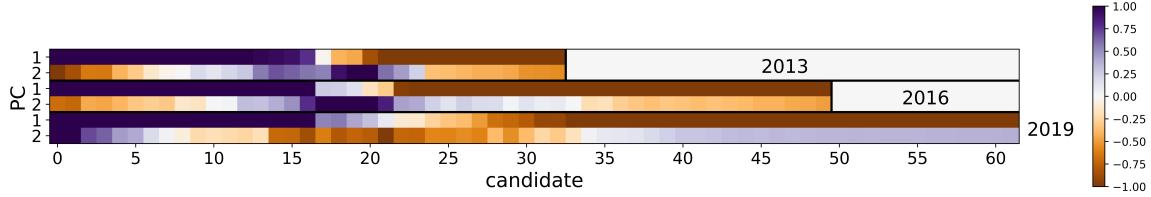


Figure 4.1: Reduced features of candidates' cumulative vote distributions along the first two PCs. PCA reduced the number of features from 60 to 2.

Number of clusters from silhouette analysis

From the plot of distortion (overall MSE) vs number of clusters (Fig. 4.2), the elbow looks to be around $K = [2, 5]$ for all years. Thus, we applied silhouette analysis on $K = [2, 5]$ (Fig. 4.3). After performing silhouette analysis, we deemed $K = 2$ as the optimal number of clusters for all election years (Table 4.2).

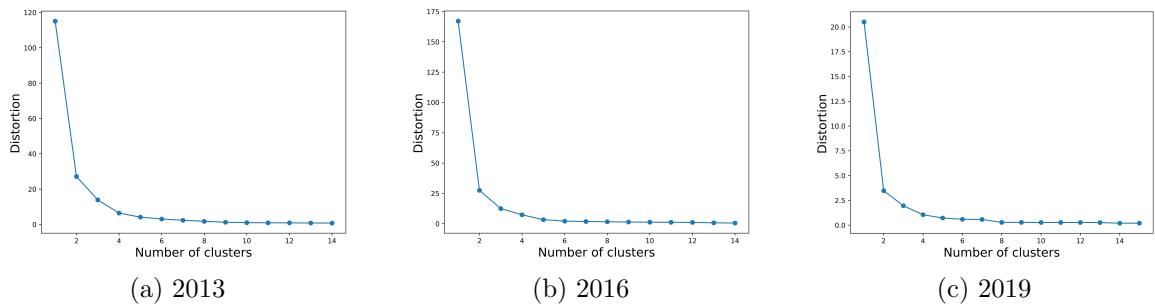


Figure 4.2: Distortion (overall mean square error) vs number of clusters for K-means clustering per year. The elbow looks to be around $K = 2 - 5$.

In 2013, $K = 2$ has the highest $\bar{s}(i)$, but $K = 3$ has the most number of points with $s(i) \geq \bar{s}(i)$. $K = 3$ only differs by one from $K = 2$, which validates using either. We decided to use $K = 2$ for simplicity.

In 2016 and 2019, $K = 2$ has the highest $\bar{s}(i)$ and the most number of points with $s(i) \geq \bar{s}(i)$. However, in 2019, one point has a negative $s(i)$ at $K = 2$. To remedy this, we reclassified that point to a different cluster. The $\bar{s}(i)$ decreased by about 0.002, but remained as the highest for 2019.

Table 4.2: Summary of results for silhouette analysis

K	$\bar{s}(i)$			points w/ $s(i) \geq \bar{s}(i)$			points w/ $s(i) < 0$		
	2013	2016	2019	2013	2016	2019	2013	2016	2019
2	0.64	0.74	0.80	22	37	40	0	0	1
3	0.62	0.69	0.76	23	37	40	0	0	0
4	0.57	0.64	0.73	19	33	37	1	0	0
5	0.59	0.63	0.72	19	31	35	0	0	0

Clustered candidates after K-means clustering

The top 18, 17, and 16 senate candidates clustered together in 2013, 2016, and 2019, respectively, for $K = 2$ (Fig. 4.4). By performing K-means clustering for $K > 2$, we found that the candidates only split across ranks (example shown in Table 4.3). We also observe the cluster splitting in the silhouette plots for $K > 2$ (Fig. 4.3). The big cluster silhouettes in $K = 2$ either remain up to $K = 4$ (2016), or split into smaller clusters (2013).

Table 4.3: K-means clustering of candidates for $K = 5$. K-means clustering only splits the candidates along their overall ranking.

cluster	1	2	3	4	5
2013	0-6	7-16	17-20	21-23	23-32
2016	0-11	12-16	17-21	22-33	34-49
2019	0-5	6-16	17-26	27-33	34-61

K-means clustering merely separates the winning candidates from the losing candidates, and increasing K provides no further distinction other than the candidates' overall rank. Thus, we now turn to hyperspectral unmixing to see how the unmixed voting archetypes compare.

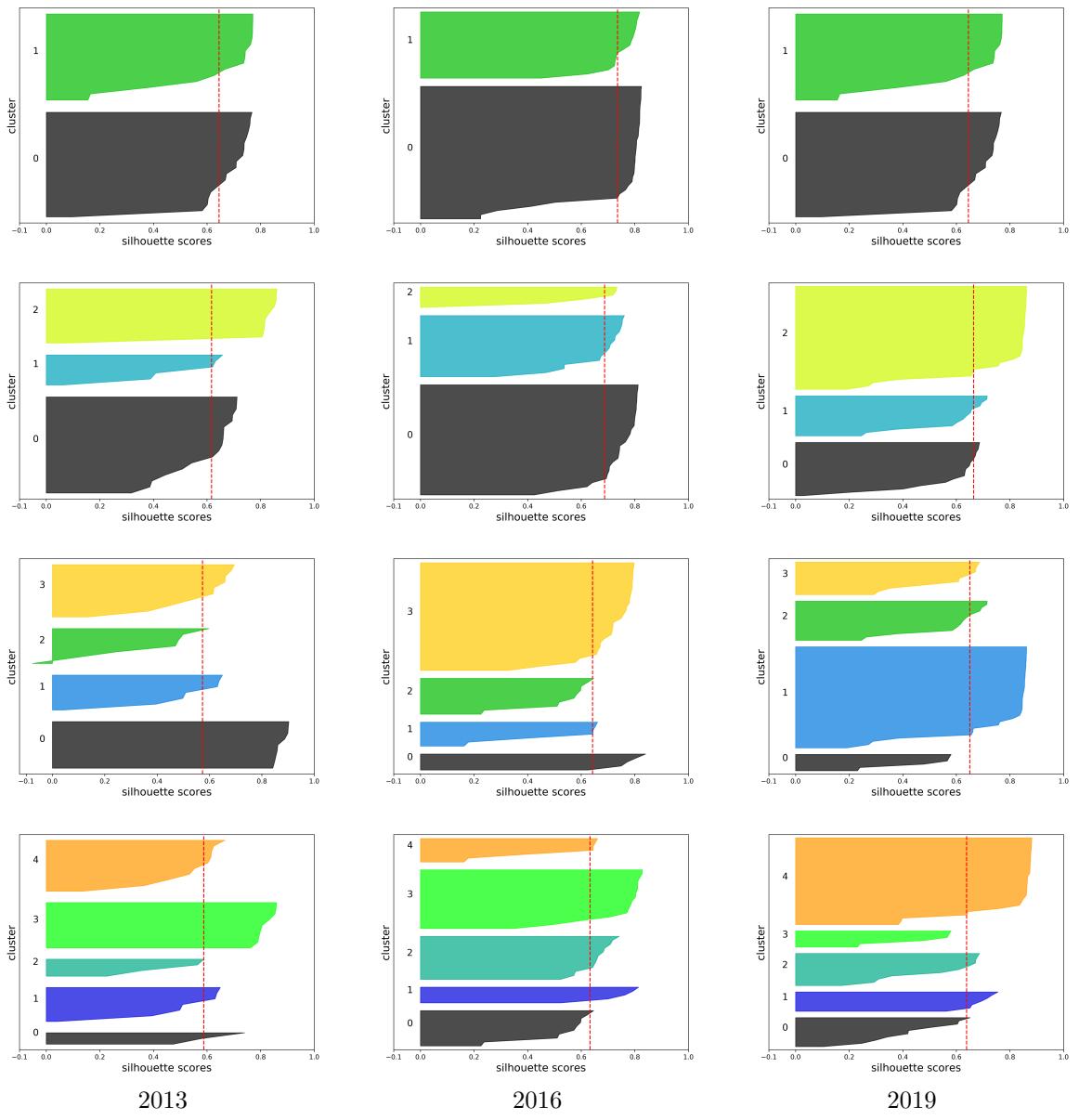


Figure 4.3: Silhouette plots of $K = 2, 3, 4, 5$ in K-means clustering per year. The red line denotes the overall mean silhouette score. Cluster silhouette size is proportional to the number of data points in cluster.

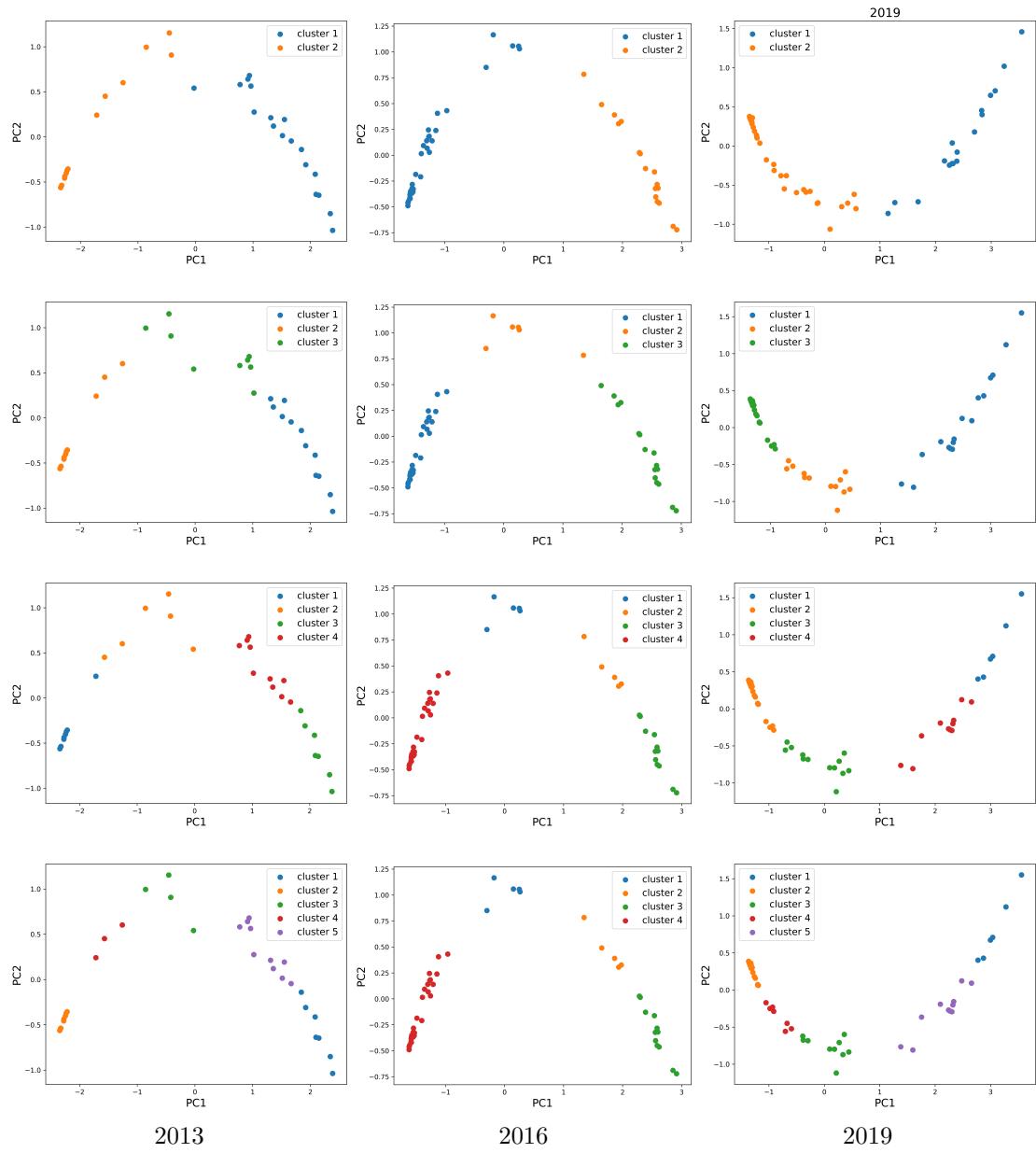


Figure 4.4: Clusters of candidates formed after using K-means clustering for $K = 2, 3, 4, 5$ on candidates' cumulative votes distribution. The features were reduced from 60 to two using PCA.

4.2 Voting archetypes after hyperspectral unmixing

Table 4.4: Voting archetypes per year arranged from highest to lowest mean weight. Archetypes are based on candidates with high vote intensities in Fig. 4.5.

	2013	2016	2019
1	opposition	popular coalitions	previous senators
2	pro-reproductive health (RH) bill	provincial politicians	conservatives
3	anti-RH bill	winners	multimedia popularity
4	joint campaign	celebrities	island group association
5	conservatives	TV news popularity	celebrities
6		written news popularity	opposition

HySime determined five, six, and six optimal number of archetypes (or reduced dimensions) for the 2013, 2016, and 2019 election years, respectively. We based the archetype labels (Table 4.4) on the normalized vote spectra \mathbf{M}/\mathbf{n} estimated by MVSA (Fig. 4.5). We also mapped the dominant archetypes per Philippine province based on the median weights and mean weights estimated by SuNSAL (Figs. 4.6 and 4.7). By mapping the dominant archetypes, we can determine associations between candidates and specific locations. The complete list of candidate names per year can be found in the Appendix. We now detail how we labeled each archetype.

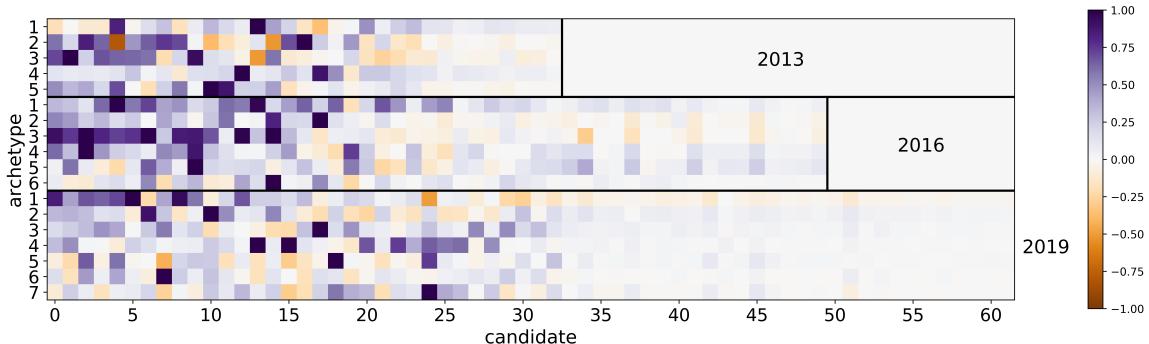


Figure 4.5: Normalized vote intensities of archetypes per year. Archetypes were arranged based on overall mean weight, and candidates were arranged based on total votes.

2013 elections

The stance of candidates on the Philippine Reproductive Health (RH) Bill, a prominent social issue within the country last 2013, influenced voters' decisions. During that year, the Catholic church endorsed candidates opposing the Bill, while condoning those that support it [32]. Issues surrounding the RH Bill divided the Filipinos, a predominantly Catholic population, thereby resulting to two opposing archetypes: archetypes 2 and 3.

Candidates with high vote intensities in the first and fifth archetypes rallied behind the then vice president and president, respectively. The conservatives called themselves *Team PNoy*, aptly named after the president Benigno Simeon "Noynoy" Aquino III. The vice president formed his own opposition coalition for the 2013 senate elections: United Nationalist Alliance or UNA (translates to *first*) [3]. The fourth archetype dominates in the home province of candidate 12 (A in Fig. 4.6a and 4.7a), the candidate with the highest intensity in the fourth archetype. Majority of Filipinos also know him as the chairman of the Philippine Red Cross, a humanitarian organization that provides life-saving services throughout the country. Candidates 17 and 18—2nd and 3rd highest intensity in the fourth archetype—endorsed each other and campaigned together as good friends [33]. Unfortunately, we cannot find a clear correlation among candidates 12, 17, and 18, even if they rank high in the same archetype.

2016 elections

President-vice president candidate tandems typically choose 12 senate candidates to endorse while campaigning. Senate candidates may belong to more than one ticket, or may join a ticket hosted by a party they do not belong to [2]. The high intensity candidates in the most dominant archetype came from two different senatorial tickets—the senatorial tickets of the 2nd and 3rd most popular presidential candidates.

The party of the most popular presidential candidate in 2016 excluded themselves from forming a senatorial ticket, saying they wanted to focus campaigning for their presidential and vice presidential candidates. The 2nd most popular presidential candidate formed a senatorial ticket called *Koalisyon ng Daang Matuwid* (translated as Coalition for the Straight Path), while the 3rd most popular candidate formed a senatorial ticket called Partido Galing at Puso (translated to Party with Ability and

Heart). Candidates 5, 11, and 16 belong to *Koalisyon ng Daang Matuwid* [34], while candidates 4, 6, 18, and 21 belong to Partido Galing at Puso [35].

The second archetype dominates Region VIII (B in Fig. 4.6b and 4.7b), where the province of the candidates with the highest intensities, candidates 17 and 14, belongs to. Candidate 17 served as the governor from 2004 to 2012, while candidate 14 served as 1st district representative for three consecutive terms from 2007 to 2016. Candidates 17 and 15 expectedly dominate in their home region due to their history as local politicians.

Candidates 2, 7, 9, 15, and 19—high intensities in the fourth archetype—represent Filipino celebrities [36]. Candidate 7 married a Filipino celebrity, while candidate 15 campaigned using his more popular screen name as a previous celebrity. On the other hand, candidates 2 and 19 capitalized on the popularity of their last names in the Filipino action films industry. Celebrity candidates possess powerful name recall due to their frequent media exposure, thereby endowing them with higher chances of being voted together.

Candidate 9 (highest intensity in the fifth archetype), a famous Filipino international boxer and celebrity, received the most reports from prime time newscasts. His high news coverage stemmed from his vocal criticisms on same-sex marriage and not necessarily his political career [37].

On the other hand, candidate 14—highest intensity in the sixth archetype—received the most coverage from newspapers. Campaign reports recorded his stand on government employees' pay hike, contingency for overseas Filipino workers, and the bill on increasing Social Security System pensions [37]. The Manila Standard, a newspaper company his family owns, published half of his articles. Archetype five and six demonstrate how voters' decisions may depend on how they partake of news: whether through written or TV news.

2019 elections

During the 2019 elections, the current administration underwent strict scrutiny from the public eye and from other politicians (especially the opposition) [38]. The polarizing behavior of the administration motivated a clear divide between opposition supporters and conservative supporters. The opposition candidates called themselves *Otso Diretso* (translated as “Straight Eight”), while the conservative candidates called

themselves *Hugpong ng Pagbabago* (translated as “Faction for Change”).

Candidates 2, 4, 7, and 8—high intensities in first archetype—belong to *Hugpong ng Pagbabago*. The other members of *Hugpong ng Pagbabago* classify to other archetypes. The conservatives dominate in the northern regions of the Philippines (D in Fig. 4.6c and 4.7c) where candidate 7’s home province resides.

Candidates 20, 22, 24, 25, 26, and 28—high intensities in sixth archetype—comprise *Otso Diretso*. The other two members of *Otso Diretso* held previous senate positions, so they classify into the first archetype. The opposition dominates in candidate 24’s home province (C in Fig. 4.6c and 4.7c).

Given their significantly high intensities, we can consider candidates 7 and 24 as the strongest representatives of the conservatives and opposition, respectively. Candidate 7’s family name carries heavy political history then and now—thus, her strong name recall. Her relatives served as current and/or previous officials both in local and national positions [39]. Candidate 24 paints herself as an advocate of her minority group to the senate, thereby garnering their strong support [40].

Candidates 2, 4, 18, and 24—highest intensities in the fourth archetype—all hail from the same island group: the southern regions of the Philippines. Candidates 18 and 24 represent members of a minority group predominantly found in southern Philippines. On other hand, the current president, who also resides in southern Philippines, vouches for candidates 2 and 4.

Candidates 17 and 19—highest intensities in the fourth archetype—represent multimedia popularity. Candidate 17 shares simplified medical content through two social media platforms: YouTube and Facebook. He accumulated over a million subscribers in YouTube, and about 10 million likes in Facebook before campaigning [41]. Candidate 19, on the other hand, carries a familiar name as a news journalist in one of the Philippines’ biggest news networks.

Candidates 1, 6, 10 and 14—highest intensities in the fifth archetype—represent Filipino celebrities. Candidates 6, 10, and 14 worked as previous actors in the action film industry before delving into politics. On the other hand, candidate 1 carries the last name of a very famous action star from classic Filipino films. Thus, the fifth archetype can more specifically represent action stars.

Archetype 1’s domination of the Philippines shows that Filipinos tend to vote for current or previous senators. Candidates 0, 1, 13, 15, and 16—highest intensities

in first archetype—all became senators in previous elections, thereby endowing them with high name recall.

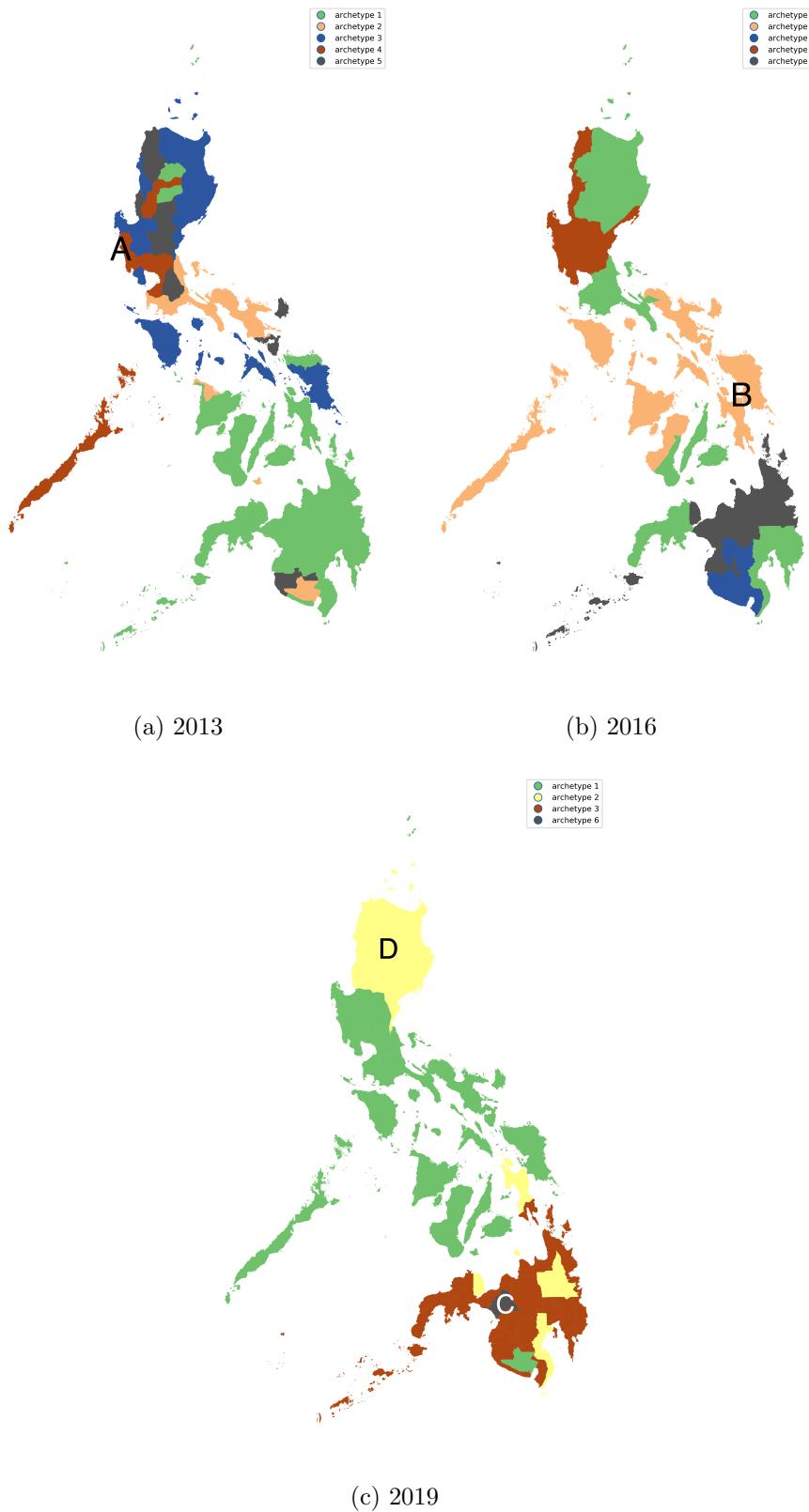
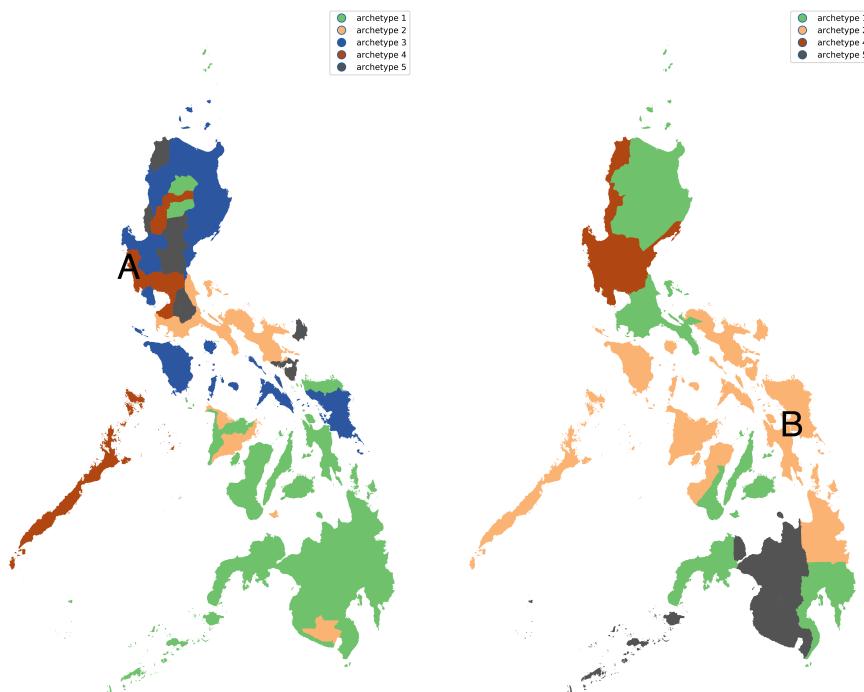
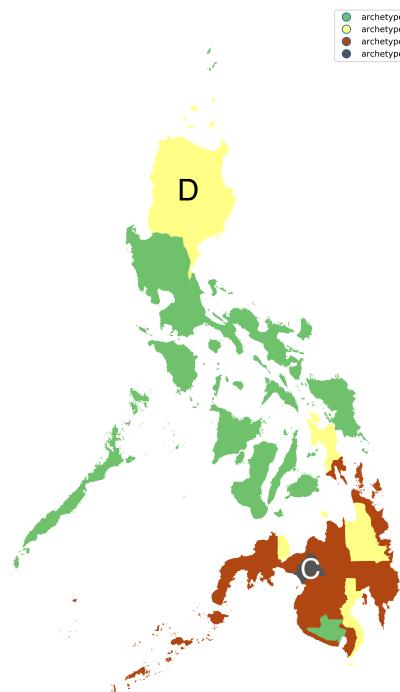


Figure 4.6: Dominant archetypes per year based on median weight per province/region. The archetypes in the legend are arranged from highest to lowest overall mean weight.



(a) 2013

(b) 2016



(c) 2019

Figure 4.7: Dominant archetypes per year based on mean weight per province/region. The archetypes in the legend are arranged from highest to lowest overall mean weight.

4.3 Comparison with studies using exit polls and pre-election surveys

We found that we obtained similar findings with other studies on Philippine elections using surveys. Some of the archetypes we obtained can also be found in earlier election years, and in papers with different methods.

Ethno-linguistic affiliation

Previous studies on Philippine voting patterns found that Filipinos tend to vote for a candidate with the same hometown, or more specifically, the same language [3, 4]. The following voting archetypes support the notion of ethno-linguistic affiliation: archetype 4 in 2013, archetype 2 in 2016, and archetypes 2, 4, and 6 in 2019.

Multimedia popularity

Other studies on Philippine voting patterns delved into the impact of celebrity candidates to voters' decisions [1, 2]. According to previous studies, celebrity candidates represent a familiar name in the ballot—thus, their high chance of being chosen. Celebrity candidates possess high name recall, especially since they receive substantial news and media coverage. Since majority of news migrated to social media starting around 2016, candidates began to exploit social media sites (such as Facebook) for campaigning [42]. In our study, we obtained archetypes that support the notion of celebrity candidates popularity: archetype 4 in 2016 and archetype 5 in 2019. We also obtained archetypes that show the reliance of voters on multimedia: archetypes 5 and 6 in 2016 and archetype 3 in 2019.

Party affiliation

A paper that investigated Philippine elections during 1946-1965 found that party affiliation provides weak explanatory power for voting behaviors in senate elections [4]. However, a more recent study on the 2010 senate elections found that voters hold party affiliation in high regard when it comes to supporting their candidates [2]. Campaigning alongside the current president (during midterm elections), or the popular presidential candidate (during presidential elections) tends to grant candidates higher

chances of winning. The following archetypes showcase party affiliation: archetypes 1 and 5 in 2013, archetype 1 in 2016, and archetypes 2 and 6 in 2019.

Incumbent politicians

Incumbent or previous senate candidates tend to dominate the elections, especially given their high name recall [1]. Archetype 1, the archetype with the highest mean weight in 2019, demonstrates the power of name recall. Political history could also apply to local politicians as seen in archetype 2 from 2016.

Chapter 5

Conclusions and Recommendations

Our work provides empirical evidence that voting patterns exist in the Philippine senate elections. K-means clustering can only separate top and bottom ranking candidates, while hyperspectral unmixing decomposed precinct-level results into distinct voting patterns. Thus, hyperspectral unmixing fares better than K-means clustering in pattern extraction from mixed election data.

The ability of hyperspectral unmixing to decompose mixed data provides a new approach to pattern extraction of high-dimension data beyond just satellite images. Dimension reduction allows unmixing to work with significantly lower data dimension, instead of computationally intensive high dimension data. Unmixing condenses the impractical number of senate voting combinations to fewer comprehensive archetypes.

Since hyperspectral unmixing has been thoroughly studied over the years, several algorithms on how to solve the unmixing problem have surfaced. Other unmixing algorithms could also be applied to the election results, and compared with the algorithms we used.

Hyperspectral unmixing as a pattern extraction technique can be extended to other combinations of electoral positions (not just senators). Some possible combinations of positions include the provincial board and barangay councils. We chose to work on senators, because the senate elections produce the highest possible number of voting combinations in the Philippines. The more voting combinations to unmix, the more powerful hyperspectral unmixing becomes.

Hyperspectral unmixing can also be extended to election studies outside the Philippines; one example would be the elections to the European Parliament. In this grain, surveys and polls involving different possible combinations of choices would work well with hyperspectral unmixing.

Appendix

2013 Candidates

no.	candidate
0	POE, GRACE
1	LEGARDA, LOREN (NPC)
2	ESCUDERO, CHIZ
3	CAYETANO, ALAN PETER (NP)
4	BINAY, NANCY (UNA)
5	ANGARA, EDGARDO (LDP)
6	AQUINO, BENIGNO BAM (LP)
7	PIMENTEL, KOKO (PDP)
8	TRILLANES, ANTONIO IV (NP)
9	VILLAR, CYNTHIA HANEPUHAY (NP)
10	EJERCITO ESTRADA, JV (UNA)
11	HONASAN, GRINGO (UNA)
12	GORDON, DICK (UNA)
13	ZUBIRI, MIGZ (UNA)
14	ENRILE, JUAN PONCE JR.(NPC)
15	MAGSAYSAY, RAMON JR. (LP)
16	HONTIVEROS, RISA (AKBAYAN)
17	HAGEDORN, ED
18	VILLANUEVA, BRO.EDDIE (BP)
19	MADRIGAL, JAMBY (LP)
20	MAGSAYSAY, MITOS (UNA)

no.	candidate
20	MAGSAYSAY, MITOS (UNA)
21	CASINO, TEDDY (MKB)
22	MACEADA, MANONG ERNIE (UNA)
23	COJUANGCO, TINGTING (UNA)
24	DELOS REYES, JC (KPTRAN)
25	ALCANTARA, SAMSON (SJS)
26	BELGICA, GRECO (DPP)
27	PENSON, RICARDO
28	DAVID, LITO (KPTRAN)
29	MONTANO, MON
30	LLASOS, MARWIL (KPTRAN)
31	SENERES, CHRISTIAN (DPP)
32	FALCONE, BAL (DPP)

2016 candidates

no.	candidate
0	VILLANUEVA, JOEL TESDAMAN (LP)
1	DRILON, FRANK (LP)
2	SOTTO, VICENTE (NPC)
3	LACSON, PANFILO PING (IND)
4	GORDON, DICK (IND)
5	HONTIVEROS, RISA (AKBYN)
6	ZUBIRI, MIGZ (IND)
7	PANGILINAN, KIKO (LP)
8	GATCHALIAN, WIN (NPC)
9	PACQUIAO, MANNY (UNA)
10	RECTO, RALPH (LP)

no.	candidate
11	DE LIMA, LEILA (LP)
12	GORDON, DICK (UNA)
13	OSME?A, SERGIO III (IND)
14	ROMUALDEZ, MARTIN (LAKAS)
15	DOMAGOSO, ISKO MORENO (PMP)
16	GUINGONA, TG (LP)
17	PETILLA, CARLOS JERICHO (LP)
18	COLMENARES, NERI (MKBYN)
19	LAPID, MARK (AKSYON)
20	MANZANO, EDU (IND)
21	ROMULO, ROMAN (IND)
22	OPLE, SUSAN (NP)
23	LACSAMANA, ALMA MORENO (UNA)
24	BELGICA, GRECO (IND)
25	ALUNAN, RAFFY (IND)
26	GADON, LARRY (KBL)
27	LANGIT, REY (UNA)
28	KAPUNAN, ATTY. LORNA (AKSYON)
29	PAGDILAO, SAMUEL (IND)
30	SANTIAGO, DIONISIO (IND)
31	NAPE?AS, GETULIO (UNA)
32	CHAVEZ, MEL (WPPPMM)
33	MONTANO, ALLAN (UNA)
34	AMBOLODTO, INA (LP)
35	BELLO, WALDEN (IND)
36	PALPARAN, JOVITO JR. (IND)
37	KIRAM, PRINCESS JACEL (UNA)
38	LIBAN, DANTE (IND)
39	CAM, SANDRA (PMP)
40	PAEZ, MR. COOP (IND)

no.	candidate
41	ALBANI, SHARIFF (IND)
42	MAGANTO, ROMEO (LAKAS)
43	MONTA?O, MON (IND)
44	ARQUIZA, GODOFREDO (IND)
45	ALI, ALDIN (WPPPMM)
46	BALIGOD, LEVITO (IND)
47	VALEROZO, DIOSDADO (IND)
48	DORONA, RAY (IND)
49	KABALU, EID (IND)

2019 candidates

no.	candidate
0	VILLAR, CYNTHIA
1	POE, GRACE
2	GO, BONG
3	CAYETANO, PIA
4	DELA ROSA, RONALDO
5	ANGARA, SONNY
6	LAPID, LITO
7	MARCOS, IMEE
8	TOLENTINO, FRANCIS
9	PIMENTEL, KOKO
10	REVILLA, BONG

no.	candidate
11	BINAY, NANCY
12	EJERCITO, JV
13	AQUINO, BAM
14	ESTRADA, JINGGOY
15	ROXAS, MAR
16	OSME?A, SERGE
17	ONG, WILLIE
18	MANGUDADATU, DONG
19	MANICAD, JIGGY
20	DIOKNO, CHEL
21	ENRILE, JUAN PONCE
22	ALEJANO, GARY
23	COLMENARES, NERI
24	GUTOC, SAMIRA
25	MACALINTAL, ROMULO
26	TANADA, ERIN
27	GADON, LARRY
28	HILBAY, FLORIN
29	AGUILAR, FREDDIE
30	CHONG, GLENN
31	ALUNAN, RAFFY
32	MANGONDATO, FAISAL
33	ESCUDERO, AGNES
34	PADILLA, DADO
35	ARELLANO, ERNESTO

no.	candidate
36	MANTANO, ALLAN
37	DE GUZMAN, LEODY
38	CHAVEZ, MELCHOR
39	ABEJO, VANJIE
40	CASINO, TOTI
41	AFUANG, ABNER
42	ALBANI, SHARIFF
43	ROLEDA, DAN
44	GENEROZO, DING
45	SAHIDULLA, LADY ANN
46	JANGAO, ABRAHAM
47	ARIAS, MARCELINO
48	ALFAJORJA, RICHARD
49	MATULA, SONNY
50	FRANCISCO, ELMER
51	NALLIW, JOAN SHEELAH
52	ARCEGA, GERALD
53	VALDES, BUTCH
54	CACERES, JESUS
55	AUSTRIA, BERNARD
56	BALDEVARONA, JONATHAN
57	MALLILLIN, EMILY
58	GADDI, CHARLIE
59	JAVELLANA, RJ
60	GUIGAYUMA, JUNBERT
61	MENIANO, LUTHER

Source codes

The source codes we used can be found at <https://github.com/cmmdecastro/bstthesis>. The repository does not include the datasets used, because the file sizes are too big for *github*. The datasets for 2013 and 2016 can be downloaded from <http://elections.org.ph/>.

Bibliography

- [1] C. C. David and J. M. L. Atun. Celebrity politics: Correlates of voting for celebrities in Philippine presidential elections. *Social Science Diliman*, 11(2), 2015.
- [2] C. C. David and E. F. T. Legara. How voters combine candidates on the ballot: The case of the Philippine senatorial elections. *International Journal of Public Opinion Research*, 29(1):70–94, 2017.
- [3] R. D. Holmes. The dark side of electoralism: Opinion polls and voting in the 2016 Philippine presidential election. *Journal of Current Southeast Asian Affairs*, 35(3), 2016.
- [4] H. Ando. A study of voting patterns in the Philippine presidential and senatorial elections, 1946-1965. *Midwest Journal of Political Science*, 13(4):567–586, 1969.
- [5] Republic Act No. 8436, 22 December 1997.
- [6] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches, 2012.
- [7] B. W. Rockwell, R. N. Clark, C. G. Cunningham, S. J. Sutley, C. A. Gent, R. R. McDougal, K. E. Livo, and Ra. F. Kokaly. Mineral mapping in the Marysvale volcanic field, Utah using AVIRIS data. 2000.
- [8] Y. B. Cheng, S. L. Ustin, D. Riaño, and V. C. Vanderbilt. Water content estimation from hyperspectral images and MODIS indexes in Southeastern Arizona. *Remote Sensing of Environment*, 112:363–374, 2008.
- [9] E. R. Hunt and A. E. P. Williams. Detection of flowering leafy spurge with satellite multispectral imagery. *Rangeland Ecology and Management*, 59, 2006.

- [10] M. Kanning, I. Kühling, D. Trautz, and T. Jarmer. High-resolution UAV-based hyperspectral imagery for LAI and chlorophyll estimations from wheat for yield prediction. *Remote Sensing*, 10:2000, 2018.
- [11] M. Mladenov, S. Penchev, and M. Dejanov. Complex assessment of the quality of foodstuffs through the analysis of visual images, spectrophotometric and hyperspectral characteristics. *IFAC-PapersOnLine*, 4:23–32, 2015.
- [12] R. X. Xu, D. W. Allen, J. Huang, S. Gnyawali, J. Melvin, H. Elgarably, G. Gordillo, K. Huang, V. Bergdall, M. Litorja, J. P. Rice, J. Hwang, and C. K. Sen. Developing digital tissue phantoms for hyperspectral imaging of ischemic wounds. *Biomedical Optics Express*, 3(6):1433–1445, 2012.
- [13] Y. Hart, H. Sheftel, J. Hausser, P. Szekely, N. B. Ben-Moshe, Y. Korem, A. Tendler, A. E. Mayo, and U. Alon. Inferring biological tasks using Pareto analysis of high-dimensional data. *Nature Methods*, 12:233–235, 2015.
- [14] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010.
- [15] J. Lever, M. Krzywinski, and N. Altman. Principal component analysis. *Nature Methods*, 14(7):641–642, 2017.
- [16] M. Ringnér. What is principal component analysis?, 2008.
- [17] I. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.
- [18] M. McCool, A. D. Robison, and J. Reinders. Chapter 11 - K-Means Clustering. pages 279–289. Morgan Kaufmann, Boston, 2012.
- [19] T. M. Kodinariya and P. R. Makwana. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 2013.
- [20] D. J. Ketchen Jr. and C. L. Shook. The application of cluster analysis in strategic management research: An analysis and critique. *Strategic Management Journal*, 17(6), 1996.

- [21] M. A. Syakur, B. K. Khotimah, E. M.S. Rochman, and B. D. Satoto. Integration K-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster. In *IOP Conference Series: Materials Science and Engineering*, 2018.
- [22] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.
- [23] K. Pollard and M. van der Laan. A method to identify significant clusters in gene expression data. *Proceedings of SCI*, 2002.
- [24] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, April 2005.
- [25] J. M. Bioucas-Dias and J. M. P. Nascimento. Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445, Aug 2008.
- [26] J. Li, A. Agathos, D. Zaharie, J. M. Bioucas-Dias, A. Plaza, and X. Li. Minimum volume simplex analysis: A fast algorithm for linear hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 53(9):5067–5082, Sep. 2015.
- [27] J. M. Bioucas-Dias and M. A. T. Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, June 2010.
- [28] M. Ringnér. What is principal component analysis? *nat. biotechnol.* 26, 303–304. *Nature biotechnology*, 26:303–4, 04 2008.
- [29] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.
- [30] J. Nocedal and S. J. Wright. *Numerical optimization 2nd edition*. Springer, New York, NY, USA, 2000.

- [31] R. T. Rockafellar. Convex Analysis (Princeton Mathematical Series). *Princeton University Press*, 1970.
- [32] A. Calonzo. Winners and losers from ‘Team Buhay’ and ‘Team Patay’ in 2013 polls, 2013. Last accessed 3 November 2019.
- [33] Rappler.com. ‘Two Eds are better than one’, 2013. Last accessed 3 November 2019.
- [34] M. Ager. Poe-Escudero’s ‘Partido Galing at Puso’ senate slate launched, 2015. Last accessed 05 June 2020.
- [35] B. Cupin. Ruling coalition completes 12-person senate slate, 2015. Last accessed 05 June 2020.
- [36] Rappler.com. Celebrities running, supporting candidates in the 2016 national elections, 2015. Last accessed 23 April 2019.
- [37] Center for Media Freedom and Responsibility. Candidates for senate and party-list: Hardly noticed, 2016. Last accessed 23 April 2019.
- [38] D. G. Timberman. Philippine politics under duterte: A midterm assessment, 2019. Last accessed 4 November 2019.
- [39] C. Fonbuena. Marcoses take seats in senate, congress, province and city, 2019. Last accessed 4 November 2019.
- [40] C. Fonbuena. Samira Gutoc tops senatorial polls in Marawi City, 2019. Last accessed 12 June 2019.
- [41] L. K. Subido. “We should uplift the conversation. Should I change myself in order to please them to get the votes?”-Dr. Willie Ong, senatorial candidate, 2019. Last accessed 4 November 2019.
- [42] C. C. David, R. S. San Pascual, and E. S. Torres. Reliance on Facebook for news and its influence on political engagement. *PLoS ONE*, 14(3), 2019.