Preliminary Master Use Case List (2/23/26)

CSE416, S01 – Spring 2026

This draft of the use case list is meant to help you think about requirements. You should think about additional use cases that serve the project goals and also think about more detail for each use case.

Project goals

- Understand the implications of the potential Supreme Court dismissal of Section 2 of the Voting Rights Act (VRA)
- Assess the impact on a pre-clearance state and a non-preclearance state

Approach

- Develop an estimate of the probability distribution of minority representation under the existing VRA
- Compare with the estimated probability distribution of minority representation under a restructured VRA (i.e., race-blind districting)
- Apply data analysis techniques to the available data
- Generate data tables and visualizations that summarize the results of the analysis

Terminology

- Population – the population of a region (e.g., state, precinct, etc.) refers to the total population as defined by the US Census Bureau. Some calculations might refer to the voting age population (VAP) or citizen voting age population (CVAP). You can use any measure of population, but it should be done consistently throughout the application.
- Ideal district population is the total population of the state divided by the number of districts in the state.
- Ensemble – a collection of district plans generated on the SeaWulf. A race-blind ensemble contains random district plan, each of which is random and a subset of all of the possible graph partitions that are constrained by the limits on population equality. A VRA-constrained ensemble is further constrained by VRA compliance. A race-blind ensemble is generated with the basic GerryChain software while a VRA-constrained ensemble is created with the VRA constrained GerryChain software.
- Feasible demographic group – a racial/language group for which the state population is at least 400,000.
- Demographic group definitions – use the same ones as in the Becker paper, namely every person who identified as Hispanic/Latino on the census or ACS is classified as Latino. The term Black is used for non-Hispanic respondents who selected Black as their single racial category, and White similarly. All other respondents (those non-Hispanic persons selecting two or more races, Asian American, Native American, and so on) are grouped together and designated as Other.
- The Becker paper refers to a "candidate of choice". This should be replaced by a "party of choice." For each racial group, the party with the highest average estimated vote total for

the 2024 Presidential election is identified as the group's ‘‘party of choice,’’ with the highest EI frequency as the confidence score.

Notation

Use cases listed below include a categorization following the use case title. Three categories of use cases are provided as “required,” “preferred,” and “optional.” Use cases with an “SD,” “AD,” or “part of GUI SD” indicate that the use case might be requested in the design review with “SD” referring to a sequence diagram and “AD” referring to an activity diagram. For some use cases, the GUI and the server part can be combined into one use case diagram. Many of the use cases that are related and that require activity diagrams can and should be combined into one activity diagram.

Ensemble sizes

For testing purposes, you should create smaller ensembles. For your final project presentation, your ensembles should include 5,000 plans.

General GUI (9 required)

GUI-1. Select state to display (required) (SD)
The user can pick a state through a dropdown menu or possibly through clicking on the state in a map of the US. The state selection will cause a table to be displayed that contains a summary of the ensembles available for the state. Ensemble data in the table will include the number of district plans in the ensemble and the population equality threshold used in the MCMC computation. In addition, state selection will also cause the map of the state to be displayed as described in GUI-2.

GUI-2. Display the current district plan when state is selected (required) (SD)
After selecting a state either from the map or the dropdown, by default, the user should be shown the current Congressional district plan displayed on the centered state map at a zoom level appropriate to the size and location of the state.

GUI-3. State data summary (required) (SD)
The data associated with the state will be summarized in response to the user selecting the state and shown concurrently with the map of the state (GUI-2). At a minimum, the summary data will include state population (either total or voting age population), state voter distribution (estimate of Republican and Democratic voting percentage based on 2024 Presidential voting), population of each significant racial/ethnic group in the state,), party control of the redistricting process (if any), and summary of Congressional representatives by party. You may substitute population percentage for population.

GUI-4. Display demographic heat map by precinct (required) (SD)
When the user selects a minority group from a drop-down menu, a heat map for the demographic group in the state will be displayed. The monochromatic heat map will show the percentage range of the selected group in each precinct. Choose a number of bins that effectively shows the population distribution with bin ranges that are equal. Use bounds that are integer values of population percentage. The map will include a legend that displays the bin ranges and associated colors. To improve readability, you can eliminate any bins that contain no values thereby improving the color separation.

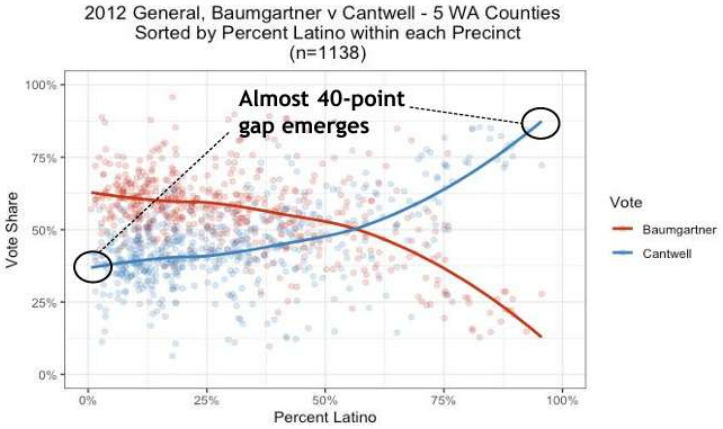
GUI-5. Display demographic heat map by census block (preferred) (SD)
When the user selects a minority group from a drop-down menu, a heat map for the demographic group in the state will be displayed. The monochromatic heat map will show the percentage range of the selected group in each census block. Choose a number of bins that effectively shows the population distribution with bin ranges that are equal. Use bounds that are integer values of population percentage. The map will include a legend that displays the bin ranges and associated colors. To improve readability, you can eliminate any bins that contain no values thereby improving the color separation.

GUI-6. Display Congressional representation table (required) (SD)
When the user clicks on screen component selecting district detail (or some other appropriate trigger), a table will be displayed. Each row in the table will contain data for one Congressional district. At a minimum, the data will contain the district number, the representative (for the enacted plan only), the representative's party, the representative's racial/ethnic group, and the vote margin as a percentage in the selected recent election.

GUI-7. Highlight district (preferred)
If a user clicks on some identifier of a district in the Congressional detail table, the district will be highlighted on the map. Highlighting can be performed in a variety of ways. For example, the border of the highlighted district might change color or thickness.

GUI-8. Compare two district plans on the map (preferred) (SD)
Compare two district plans by showing both plans on the map. This could be limited to comparing a selected random plan (i.e., interesting) with the enacted plan. The trigger will be some GUI component (e.g., "Compare with enacted" button).

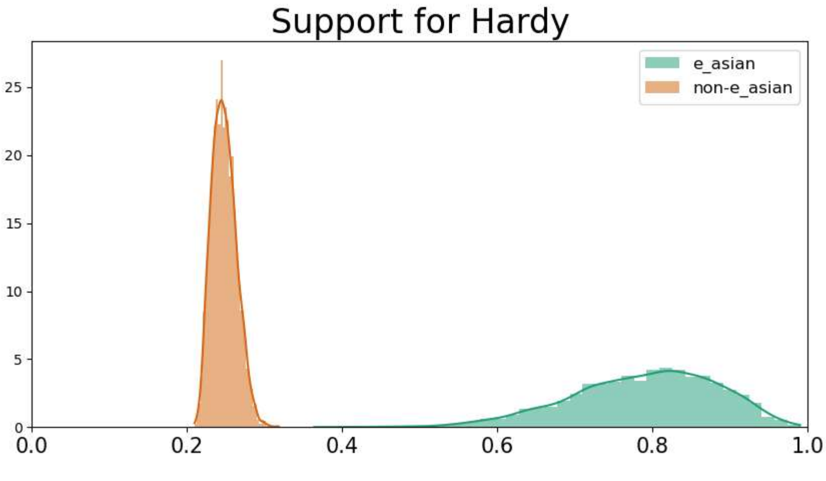
GUI-9. Display Gingles analysis results (required) (SD)
In response to a user request, display a scatter plot (example below) for each of your states that shows the 2024 precinct-level Presidential election results for each party organized on an x, y axis by percentage of racial/ethnic group in the precinct (x-axis) and party vote share (y-axis). Any of the feasible racial/ethnic groups in the state should be selectable for display. For each precinct, there will be a blue dot for Democratic votes and a red dot for Republican votes.



GUI-10. Display the Gingles 2/3 analysis data in a tabular display (preferred)
For all of the Gingles 2/3 analysis data, a table display of the precinct-by-precinct results will be displayed. Each row will display the data for a precinct including, total population, minority non-white population, average household income, Republican votes, and Democratic votes.

GUI-11. Highlight a Gingles 2/3 table row (preferred)
In response to a user selecting a dot in each of the Gingles scatter plots, the precinct identified by the dot will be highlighted in the corresponding table.

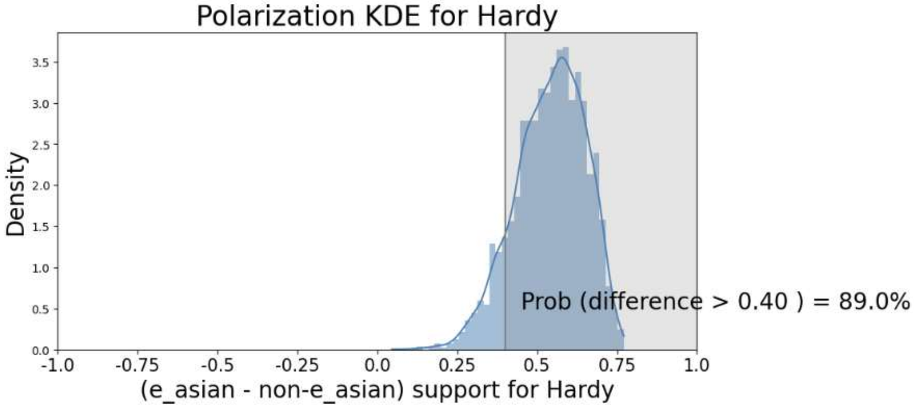
GUI-12. Display candidate results of Ecological Inference (EI) analysis (required) (SD)
Display the results of the EI analysis in response to a user GUI request. The user shall have the ability to select the racial/language groups to compare. The results will be shown in a display for each candidate (example below) in which the x-axis represents the percentage of a racial/economic/region group in the state that voted for a candidate and the y-axis represents the associated probability value for each x-axis value. The racial groups will be contained in a separate chart.



GUI-13. Display EI precinct results in a bar chart (preferred)
Display the EI results in a bar chart for the categories mentioned in a previous use case. The height of each bar will correspond to the peak value in the chart for each category. Each bar will also display a confidence interval showing the range of values determined in the EI analysis.

GUI-14. Display EI precinct results in a choropleth maps (preferred)
Display the EI results in a choropleth maps (one map per candidate) in which each precinct is displayed in a color that is consistent with the most likely level of support for the candidate in the precinct.

GUI-15. Display EI KDE results (preferred)
Display the EI KDE results that compare support for a candidate between two racial/ethnic groups (e.g., white and African American).



GUI-16. Display ensemble splits in a bar chart (required) (SD)
Display the race-blind and VRA constrained ensemble results to allow the user to make a comparison. Each display will take the form of a bar chart of Republican/Democratic splits where each bar will show the frequency of a distinct simulated election as #Republican wins / #Democratic wins. The range of splits shown should be the range of the union of the two sets of splits (i.e., the range should be the same for both displays, but the tails omitted if they are zero in both sets.

GUI-17. Display box & whisker data (required) (SD)
The user will be able to request the display of box & whisker data for each of your ensembles of district plans. The displays will be available for each of the feasible racial/ethnic groups in the state. Dots for each district in the current enacted district plan will be shown in the display (in order of increasing percentage of the minority group or region type associated display. If there is a proposed district plan awaiting approval (e.g., California), that will also be shown. The display should be sufficient in size to show your largest state and should include a legend and color selection to make the chart easily readable.

GUI-18. Display vote share vs seat share curve (preferred)
If your Gingles-2/3 test shows racially polarized voting, display the vote share vs seat share curve for the state. If the Gingles-2/3 test does not indicate racially polarized voting in one of your states, the GUI component that allows the user to select the display should be disabled.

GUI-19. Display an "interesting" district plan (preferred)
Display on the map one of the interesting plans identified in a SeaWulf use case.

GUI-20. Reset page (preferred)
When the user clicks a reset button, the GUI will reset to the condition before the user selected a state.

Preprocessing (10 required)

Prepro-1. Integrate multiple data sources (required) (AD)

Integrate and merge US Census data (income, region type (rural, urban, suburban) population, both for total and for any opportunity groups), precinct data (boundary, name, demographics, etc.), and existing district data (boundary, name, district#, etc.). Geographic boundary data should be converted (if necessary) to a consistent format (e.g., GeoJSON).

Prepro-2. Identify precinct neighbors (required) (AD)

Identify two precincts as neighbors if they share a common boundary of at least 200 feet and the edges of each precinct are within 200 feet of its neighbors' edges. If possible, try to locate a data source for which this computation is already done.

Prepro-3. Integrate enacted plan with dataset (required)

Integrate the enacted plan for the state within the server database.

Prepro-4. Store preprocessed data (required)

The preprocessed data should be stored in the NoSQL or relational database. If a relational database is used, the data should be stored in third normal form. Data might also be stored in a file system accessible to the server.

Prepro-5. Store SeaWulf data (required)

Retrieve generated data from SeaWulf for each of your states, convert to an appropriate format, and store either in your database or in a file system. Data stored in a file system should be accessible through a path obtained from your database.

Prepro-6. Generate data files required for SeaWulf processing (required) (AD)

Generate all the data files required for SeaWulf processing. This will include the graph representation of the precincts in a state as well as geographic, election, and incumbent data for each precinct.

Prepro-7. Gingles 2/3 precinct analysis (required) (AD)

Perform a precinct-by-precinct analysis of voting results and minority population percentage for some 2024 statewide race (Presidential, most likely). For each precinct, the analysis will identify the winning party, the Republican vote share, the Democratic vote share, and the population percentage of each significant racial/ethnic group. The analysis is repeated for each feasible racial/ethnic group in the state.

Prepro-8. Gingles 2/3 non-linear regression analysis (required) (AD)

For the statewide race used in the use case above, calculate the non-linear regression curve for the Republican and Democratic precinct values for each Gingles 2/3 graph. Multiple equation forms will be used to determine the best form for non-linear regression.

Prepro-9. Use the PyEI MGGG software to calculate Ecological Inference data (required)

Use the PyEI MGGG software to calculate results for the statewide race (e.g., 2024 presidential)

Prepro-10. Calculate the vote share vs seat share curve data (preferred)

Using the Shen software as a starting point, calculate the data for the vote share vs. seat share curve in any of your states that display racially polarized voting. Use the current district plan as the basis for calculation. Also use relatively fine grain increments of vote share and possibly randomization to reduce a stair-stepping effect.

Prepro-11. Calculate Box & Whisker Data for Enacted Plan (required)

Calculate the box & whisker data for the enacted district plan for any of the box & whisker displays attempted.

SeaWulf (10 required)

SeaWulf-1. Server dispatcher (required)

Establish a protected directory on SeaWulf to store your team's data. Pre-stage any data that might be used repeatedly for SeaWulf runs. Prior to submitting a batch districting run request to SeaWulf, the data required for the run should be marshalled (from memory and/or DB) and passed to the SeaWulf as a file (or multiple files) to be stored in the team's SeaWulf file system.

SeaWulf-2. Run MGGG ReCom algorithm on the SeaWulf (required) (AD)

Set the constants in the MGGG code to define the properties (e.g., constraints) of the run. Any run-control information should be packaged in a SeaWulf acceptable format (e.g., script commands) and executed on SeaWulf. Your activity diagram should demonstrate that you understand how the MGGG algorithm operates. You should generate a test ensemble and a large ensemble. The test ensemble will contain approximately 250 random district plans and the large ensemble will contain approximately 5,000 plans.

SeaWulf-3. Run MGGG VRA Constrained ReCom algorithm on the SeaWulf (required) (AD)

Set the constants in the MGGG code to define the properties (e.g., constraints) of the run. Any run-control information should be packaged in a SeaWulf acceptable format (e.g., script commands) and executed on SeaWulf. Your activity diagram should demonstrate that you understand how the MGGG algorithm operates. You should generate a test ensemble and a large ensemble. The test ensemble will contain approximately 250 random district plans and the large ensemble will contain approximately 5,000 plans.

SeaWulf-4. Coordinate/aggregate SeaWulf core generated data (required) (AD)

You will run your code on a single SeaWulf node, one that has multiple cores. Each of the cores will generate one or more random graph partitions (i.e., district plans) and store a concise version of those results in a shared file directory for your team. Following the completion of each random graph partition, the core begins the generation of the next random district plan. You will coordinate the work of the multiple cores so that when the target number of district plans is completed, each of the cores ends its processing.

SeaWulf-5. Calculate election winners (required) (AD)

Using 2024 statewide presidential results, estimate the election results in each district of each ensemble district plan. You will calculate this by summing up the estimated votes in each node (i.e., precinct) of a partition sub-graph. You can use a suitable precinct by precinct vote in the 2024 presidential election.

SeaWulf-6. Calculate minority effectiveness score per random district (required)

For each random district in a random district plan, calculate the minority effectiveness for each minority that is feasible in the state. Also, determine if that effectiveness score exceeds the effectiveness threshold.

SeaWulf-7. Calculate minority population percentage per random district (required)

For each random district in a random district plan, calculate the minority population percentage for each minority that is feasible in the state. Also, determine if that population percentage exceeds the minority population percentage threshold.

SeaWulf-8. Calculate the Republican/Democratic split for each random district plan (required) (AD)

For each generated plan in an ensemble, estimate the Republican/Democratic votes in each district. Since each district is a collection of precincts, use the historic precinct vote totals (e.g., 2024 Presidential) to estimate the winner of an election in each district.

SeaWulf-9. Identify and store additional random district plans of note (preferred) (AD)

You will not be able to store all your random district plans in the server database, but you will store some subset of those plans. Summary information and detailed information of such plans should be stored in your server database for eventual display by the user. Teams should decide what is “interesting”, but at a minimum, these should include maximum and minimum effectiveness plans. About 5-10 plans would be sufficient. The “interesting” plans would be available for display in the GUI.

SeaWulf-10. Calculate ensemble measures (required) (AD)

Calculate the summary measures for each ensemble. At a minimum, measures will include the number of district plans, and for each plan, Republican/Democratic splits, number of minority effective districts, and the number of opportunity districts (i.e., majority-minority districts).

SeaWulf-11. Calculate box & whisker data (required) (AD)

Calculate the box & whisker summary data for all the random district plans generated by the SeaWulf. These calculations will be made for each feasible racial/ethnic group in the state.

SeaWulf-12. Run on multiple SeaWulf nodes (preferred)

Run the MGGG algorithm and supplemental code on multiple nodes on SeaWulf. Node coordination should be done using a suitable coordination mechanism (e.g., MPI). Be able to estimate speed-up during your final project presentation. Note that this is optional since your algorithms should run effectively on a large-core single node processor.

SeaWulf-13. Python profiler (preferred)

Profile your system performance on SeaWulf using a Python profiler tool. Identify the procedures that consume the most CPU time. Results can be displayed using some Python-appropriate tool and displayed as an image in your final presentation.

Server Processing (0 required)

All server processing addressed in GUI use cases.

Additional Use Cases