README

General guidance to analysis: Estimating political polarization / partisanship series

We describe below the multi-step estimation procedure[1] used to calculate the polarization series (with different specifications). Most of the results in the paper and Online Appendix are produced by changing the parameters of these scripts. Note that running these scripts requires a substantial amount of computing resources, and thus, a computing cluster is usually needed to run most of the scripts. Using a computing cluster, you can run these various Python and R scripts using the .sh files (e.g.., tool-exec-1.sh, tool-exec-2.sh, etc.) included in each of the subfolders.

The important scripts are the `tool-exec` bash scripts located in subfolders inside the **`/analysis/code/`** directory.

Scripts run on `partyvar` level. For example, if you want to run the 'vanilla' left-right comparison with all controls, you set `partyvar='left'` and `covariates = 'c3'`. For each main text Figure and Table (where applicable), we describe what these parameters are in the section "Producing main text figures and tables".

Scripts create a working directory called [partyvar] within the `temp` folder, into which the intermediate files are being created. Results are similarly output under `analysis/output/[partyvar]` folder.

[1] We use the Gentzkow et al. (2019) estimator to estimate partisanship/polarization using Finnish parliamentary speech data. We use some code and notation (such as the way of naming some variables and groups) from the Gentzkow et al. (2019) minimal example. In our codes, "rep" and "dem" labels obviously do not have the same meaning as in the Gentzkow et al. paper. Instead, in our codes they are used simply as labels for any two groups that are compared to each other (see "partyvar" parameter in the table).

See the below table for parameters used in the .sh scripts. Below the table, we describe what kind of scripts different .sh scripts run. Some Online Appendix robustness checks use other values for “partyvar” parameter and those are described later in this file.

Parameter	Description	Value	Description
`partyvar`	The party split used for the analysis. `partyvar='left'` runs the main left-right speech comparison.	`left`	left-right comparison
		`govparty`	government-opposition comparison
		`leftnonsmp`	left-right comparison without SMP
		`leftnonvennamo`	left-right comparison without Vennamo
		`demkesk`	SDP-Centre Party comparison
		`demkok`	SDP-Coalition Party comparison
		`kokkesk`	Coalition Party-Centre Party comparison
		`vasdem`	Vas-SDP comparison
		`vaskesk`	Vas-Centre Party comparison
		`vaskok`	Vas-Coalition Party comparison
`nr_datasets`	into how many datasets data is split	10 by default	

Parameter	Description	Value	Description
`Cpar`	determines which counts dataset is used	`Cadj` (default)	`speaker_phrase_counts_bipartisan_adj.rds` (more procedural phrases removed)
		empty	`speaker_phrase_counts_bipartisan.rds`
`covariates`		`c0`	No covariates
		`c1`	controls for party's government-opposition status
		`c2`	controls for gov-opp status + dialect
		`c3`	controls for gov-opp status + dialect + gender
		`c4`	controls for dialect + gender
		`c99`	no controls, try a low value for the penalty coefficient psi ψ
		`c100`	no controls, try different values for the penalty coefficient psi ψ
`penalty`	whether or not LASSO is used		
`fake_indicator`	placebo or not (fake indicator = 1 means placebo series)		

`tool-exec-1-prepare-data.sh`

Description:

Runs three different scripts that prepare covariates and split the datasets for the purposes of the main analysis.

`prepare-data-1.py`:

Description:

Create party indicators and dialect region variable; only keep the MPs that served in the parliament during the parliamentary year. Create `id` variable, which concatenates year and `speaker_id`. Note that `speaker_id` is not used for anything as such

Input: `build/output/mps-ministers.csv`

Output: `analysis/input/speaker_metadata_bipartisan.csv`

`prepare-data-2.R`:

Description:

Do some small changes in data to better fit R way of doing things: make categorical variables factor variables. Drop MPs from Åland. Drop speech by parliament chair. Create `randlabel` variable, which is a randomly created party variable for the placebo tests.

Input:

- `analysis/input/bow-1907-2018-tf-100-df-10-ytf-10.csv`
- `analysis/input/speaker_metadata_bipartisan.csv`

Output:

- `analysis/temp/[partyvar]/speaker_metadata_bipartisan.rds`: speaker metadata and the specified party indicator

- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan.rds`: matrix with phrase counts only; speaker_id-year id as a row name

`prepare-data-2b.R`:

Description:

Filter out some more procedural language.

Input:

- `analysis/input/list-adjust-c2.csv`
- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan.rds`

Output:

- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan_adj.rds`

`prepare-data-3.R`:

Description:

Split data (both metadata and phrase count file) into [nr_datasets] different datasets (default = 10). There were some memory-related issues why I ended up using a split.

Add μ (nr of phrases spoken by a given MP during the year) into data

Input:

- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan_adj.rds`
- `analysis/temp/[partyvar]/speaker_metadata_bipartisan.rds`

Output:

- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan_adj_1.rds`

- `analysis/temp/[partyvar]/speaker_phrase_counts_bipartisan_adj_2.rds`, etc.
- `analysis/temp/[partyvar]/speaker_metadata_data_1.rds`, etc.

`tool-exec-2-estimate.sh`

Description:

Chooses the main analysis program based on the covariates specified and runs it.
If `randlabel=1`, computes the placebo series.

Input

Output:

- `analysis/temp/[partyvar]/utility_dem_[Cpar]_[dataset_nr].rds`
- `analysis/temp/[partyvar]/utility_rep_[Cpar]_[dataset_nr].rds`
- `analysis/temp/[partyvar]/utility_[Cpar]_[dataset_nr].rds`
- `analysis/temp/[partyvar]/phi_[Cpar]_[dataset_nr].rds`

`tool-exec-3-combine.sh`

Description:

Combine data from the `nr_datasets` datasets.
Memory-heavy program. Last ran with 250G memory, but I remember needing to increase this further to 320G or similar for some specifications.

Includes a lot of "clutter" (many print statements) and is overall not the most elegant piece of code. Reason was that I had some memory-related issues running this code in the past and I wanted to learn where the code failed.

`tool-exec-4a-subsample.sh`

Description:

Estimates the partisanship series 100 times for a subsample of the data. A different program is executed depending on which covariates are chosen. Subsample is drawn randomly without replacement. Currently in most of the specifications subsample size is set to 20%. This is hardcoded in the subprogram but could well be improved by writing subsampling percentage into a variable.

`tool-exec-4b-subsample-empirical.sh`

Description:

Set parameter `empirical = 1` (in scripts where this is applicable) when estimating polarization

`tool-exec-5-cis.sh`

Description:

Construct confidence intervals based on the distribution computed in 4a

`tool-exec-6-phrases.sh`

Description:

Runs scripts that compute and tabulate phrase partisanship

Producing main text figures and tables

Figure 1: Left-right partisanship, controls for government status, speaker gender and region

Producing this figure is a multi-step procedure including some data preparation, estimation, inference and plotting. To produce this figure, you will need to run all of the following scripts in this order (using the .sh scripts included in each subfolder):

```
/code/tool-exec-1/prepare-data-1.py

/code/tool-exec-1/prepare-data-2b.R

/code/tool-exec-1/prepare-data-2-left.R

/code/tool-exec-1/prepare-data-3.R

/code/tool-exec-2/compute-multidata-estimate-c3.R

/code/tool-exec-3/compute-multidata-combine.R

/code/tool-exec-4a/subsampling-c3.R

/code/tool-exec-5/compute-cis.py

/code/fig-partisanship-cis.py
```

Running these scripts require a substantial amount of computing resources, and thus, a computing cluster is usually needed to run these scripts. Using a computing cluster, you can run these scripts using the .sh files (e.g.., tool-exec-1.sh) included in each of the subfolders. The .sh files usually specify, e.g., the controls used ("c3" is the specification used in Figure 1, and partyvar is "left", see the table at the start of this file for more information)

Figure 2: Magnitude of polarization when more than 1 phrase is heard

This figure is produced by running the .sh file "\code\other_analysis\magnitudes\exec_left.sh", which runs the following R scripts:

- `"\code\other_analysis\magnitudes\code_left.R"`
- `"\code\other_analysis\magnitudes\Magnitudes_left.R"`

Figure 3: Left-right partisanship after Soviet Union related speeches have been dropped, controls for government status, speaker gender and region

Producing this figure is a multi-step procedure including some data preparation, estimation, inference and plotting. To produce this figure, you will need to run the following scripts in this order:

```
/code/other_analysis/drop_NL_phrases/tool-exec-1/prepare-data-1.py

/code/other_analysis/drop_NL_phrases//tool-exec-1/prepare-data-2b.R

/code/other_analysis/drop_NL_phrases//tool-exec-1/prepare-data-2-left.R

/code/other_analysis/drop_NL_phrases//tool-exec-1/prepare-data-3.R

/code/other_analysis/drop_NL_phrases//tool-exec-2/compute-multidata-estimate-c3.R

/code/other_analysis/drop_NL_phrases//tool-exec-3/compute-multidata-combine.R

/code/other_analysis/drop_NL_phrases//tool-exec-4a/subsampling-c3.R

/code/other_analysis/drop_NL_phrases//tool-exec-5/compute-cis.py

/code/other_analysis/drop_NL_phrases//fig-partisanship-cis-nldropped.py
```

Running these scripts require a substantial amount of computing resources, and thus, a computing cluster is usually needed to run these scripts. Using a computing cluster, you can run these scripts using the .sh files (e.g.., tool-exec-1.sh) included in each of the subfolders. The .sh files usually specify, e.g., the controls used ("c3" is the specification used in Figure 1, and partyvar is "left", see README file for more information)

Figure 4: Left-right partisanship without the extreme left party (SKDL), controls for government status, speaker gender and region

Producing this figure is a multi-step procedure including some data preparation, estimation, inference and plotting. To produce this figure, you will need to run the following scripts in this order:

```
/code/tool-exec-1/prepare-data-1.py

/code/tool-exec-1/prepare-data-2b.R
```

/code/tool-exec-1/prepare-data-2-left.R

/code/tool-exec-1/prepare-data-3.R

/code/tool-exec-2/compute-multidata-estimate-c3.R

/code/tool-exec-3/compute-multidata-combine.R

/code/tool-exec-4a/subsampling-c3.R

/code/tool-exec-5/compute-cis.py

/code/fig-partisanship-cis.py

Running these scripts require a substantial amount of computing resources, and thus, a computing cluster is usually needed to run these scripts. Using a computing cluster, you can run these scripts using the .sh files (e.g.., tool-exec-1.sh) included in each of the subfolders. The .sh files usually specify, e.g., the controls used ("c3" is the specification used in Figure 1, and partyvar is "left", see README file for more information)

Figure 5: Polarization of parliamentary speech and the efficiency of policymaking

These are plotted in Stata dofile "C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\figures.do"

Figure 6: Net-of-random polarization of parliamentary speech and voter polarization

These are plotted in Stata dofile "C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\figures.do"

Table 1: Pearson's product moment correlations (bills, laws, etc.)

No interpolation of NAs: "C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\time_series_regressions\do_not_impute.R"

Interpolation of NAs:

"C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\time_series_regressions\timeseries.R"

Table 2: Pearson's product moment correlations (gini, gdp, etc.)

No interpolation of NAs:

"C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\time_series_regressions\do_not_impute.R"

Interpolation of NAs:

"C:\Users\jernie\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\time_series_regressions\timeseries.R"

Results in Supplementary Material: Producing Online Appendix Figures and Tables

Figure A1

Panel (a) No covariates: choose partyvar = "left" and covariates = "c0" and run the files described in previous section

Panel (b) Covariates: indicator for gov party: choose partyvar = "left" and covariates = "c1" and run the files described in previous section

Panel (c) Covariates: indicator for gov party and region: choose partyvar = "left" and covariates = "c2" and run the files described in previous section

Panel (d) Comparison of covariate sets: figure shows estimations with no controls "c0" and with different covariate sets ("c1", "c2", "c3"). To produce these results, first run all of the tool-exec-scripts with partyvar = "left" and for each of these different covariate specifications. Then, after you have saved the results from the estimation procedure, you can plot these results in a same figure using a file that can be found in /analysis/code/other_analysis/additional_scripts/fig-comparison-covariates.py.

Figure A2: Left-right partisanship without SMP

(a) Real series without SMP and random falsification series: choose partyvar = "leftnonsmp" and covariates = "c0". After running the scripts mentioned in the previous section with these parameters, run "fig-partisanship.py" with these same parameters and filename1 = 'partisanship-leftnonsmp_Cadj_c0-cis.csv', filename2 = 'partisanship-leftnonsmp_randlabels_Cadj_c0-cis.csv', filename3 = 'partisanship-left_Cadj_c0-cis.csv', include_random = 1, compare = 0

(b) Left partisanship with and without SMP: choose partyvar = "" and covariates = "": After running the scripts mentioned in the previous section with these parameters, run "fig-partisanship.py" with these same parameters and filename1 = 'partisanship-leftnonsmp_Cadj_c0-cis.csv', filename2 =

'partisanship-leftnonsmp_randlabels_Cadj_c0-cis.csv', filename3 = 'partisanship-left_Cadj_c0-cis.csv', include_random = 0, compare = 1

Figure A3: Left-right partisanship calculated from empirical phrase choice probabilities

Set empirical = 1 when estimating partisanship

Figure A4: Phrase usage and partisanship

(a) Number of phrases in vocabulary with nonzero counts cjt:

- ds-nr-phrases-used.R
- fig-nr-phrases-used.py

(b) Number of phrases with nonzero coefficients ϕjt for party-year indicators:

- ds-nonzero-coefs
- fig-nonzero-coefs.py

Figure A5: Left-right partisanship: the impact of ψ

Change the value of "phi" when estimating partisanship. In Panel (a) set ψ = 10−4. In Panel (b), set ψ = 10−6

Figure A6: Partisanship, pairwise comparisons

(a) Social Democrats and National Coalition Party: choose partyvar = "demkok" and covariates = "c0"

(b) Extreme left and the National Coalition Party: choose partyvar = "vaskok" and covariates = "c0"

(c) National Coalition Party and Centre Party: choose partyvar = "kokkesk" and covariates = "c0"

(d) Extreme left and the Centre Party: choose partyvar = "vaskesk" and covariates = "c0"

(e) Social Democrats and Centre Party: choose partyvar = "demkesk" and covariates = "c0"

(f) Extreme left and the Social Democrats: choose partyvar = "vasdem" and covariates = "c0"

Figure A7: Government-opposition partisanship, controls for gender and dialect region

Set partyvar = "govparty" and covariates = "c0"

Figure A8: Left-right polarization with all controls, senior (not first-term) MPs only

Set partyvar = "left_senior" and covariates = "c3"

Figure A9: Left-right polarization with all controls, changing phrase inclusion criteria from 100-10-10 to 80-10-10

In prepare-data-2, change "suffix" to suffix <- '-1907-2018-tf-80-df-10-ytf-10.csv' and run rest of the scripts as before.

Figure A10: Left-right polarization with all controls, changing phrase inclusion criteria from 100-10-10 to 100-8-10

In prepare-data-2, change "suffix" to suffix <- '-1907-2018-tf-100-df-8-ytf-10.csv' and run rest of the scripts as before.

Tables D1-D13: Most partisan phrases

"Most partisan phrases" are calculated using the scripts in /analysis/code/tool-exec-6/.

Figure D1: Left-right partisanship of selected phrases

Run "zeta-select-phrases.R" and "fig-selected-zetas.py"

Figure E1: Phrases related to the Soviet Union and Left-Right Political Polarization

See dofile in "/analysis/code/other_analysis/NL/NL_relativecounts.do

Figure E2: The usage of Soviet Union phrases by the SKDL and its successor (Left Alliance)

The code "analysis\code\other_analysis\SKDL\SKDL_code_remote.R", which can be run in a computing cluster using the "tool-skdl.sh" script located in the same folder. You need to have run the main analysis (polarization series, etc.) before running this.

Figure E3: Seat share of left parties in the Parliament

Run "/analysis/code/other_analysis/additional_scripts/fig-timeline-left"

Figure E4: The number is based on a manual audit of 10 randomly selected parliamentary transcripts from one randomly selected year in each decade. The manual audit was conducted in October, 2019.

Figure E5: Amount of speech over time (a) Total number of speeches given in a year (b) Average number of phrases spoken in a year

Panel (a): see the following files in "/analysis/code/other_analysis/additional_scripts" folder: "ds-amount of-speech.py" and "fig-amount-speech.py"

Panel (b): see " \analysis\code\other_analysis\additional_scripts\fig-descriptives.py"

Figure E6: Share of government and opposition MPs speaking, 1907–2018

See the following files: "\analysis\code\other_analysis\additional_scripts\ds-speeches-by-govparty.py" and "\analysis\code\other_analysis\additional_scripts\fig-speeches-by-govparty"

Tables E1: Share of MPs and speeches by party

File "\analysis\code\other_analysis\additional_scripts\ds-speeches-per-party.py" constructs a data that has these results per party

Table E2: Summary statistics: Left parties, gender and government

File "\analysis\code\other_analysis\ds-speeches-per-covariate" constructs a datframe that contains these results per covariate

Table E3: Summary statistics: Regions

File "\analysis\code\other_analysis\ds-speeches-per-covariate" constructs also a dataframe that contains the results by district

Table E4: Regression results for the time series

See " analysis\code\other_analysis\time_series\time_series_regressions\do_not_impute.R"

Figure E5: The Finnish economy during 1917-2018

"C:\Users\jeremias.nieminen\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\time_series\figures.do"

Figure E6: Within and Between Topic Polarization

First, you need to run the regular polarization estimations. Then, run the scripts in all of the following folders:

/analysis/code/topics/topic_classification_left/partisanship_between_topic

/analysis/code/topics/topic_classification_left/partisanship_by_topic

/analysis/code/topics/topic_classification_left/partisanship_topic_plots

Figure E7: Topics by Parties

See "C:\Users\jeremias.nieminen\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\topics\topicregs_left\code_remote.R"

Figure E8: Topics by Party Groups

See "C:\Users\jeremias.nieminen\Dropbox\REPLICATION_PACKAGE_JHPE\analysis\code\other_analysis\topics\topicregs_left\code_remote.R"

Table E5: Named topics. Topics named by the authors, i.e., table shows how we have chosen to name the topics.

Figure E9: SKDL by Category

See files "\analysis\code\other_analysis\topics\stm_leftright\structural_topic_model_all_years.R"

and "analysis\code\other_analysis\topics\stm_leftright\STM_postestimation.R"

Figure E10: Left-Right by Category

See files "\analysis\code\other_analysis\topics\stm_leftright\structural_topic_model_all_years.R"

and "analysis\code\other_analysis\topics\stm_leftright\STM_postestimation.R"

Figure E11: Urbanicity, speech divergence

Run the polarization estimation procedure (described at the beginning of this readme file) and choose partyvar = “urban” and covariates = “c0_partyf”

Figure E12: Uusimaa region vs. other regions, speech divergence

Run the polarization estimation procedure (described at the beginning of this readme file) and choose partyvar = “uusimaa” and covariates = “c0_partyf”

Figure E13: White-Collar vs. Blue-Collar MPs, speech divergence

Run the polarization estimation procedure (described at the beginning of this readme file) and choose partyvar = “whitecollar” and covariates = “c0_partyf”