Abstract

Abstract

Contents

1 Introduction

In Republic of Lithuania, public elects their representatives to a parliament in which new legislation is considered. By this election each citizen delegates specifics of legislation process to their representatives so they don't have to actively participate in the process.

However, a problem arises once citizen wants to validate what his representative has been doing. Single term of office NEEDS-CITATION involves thousands of complicated laws and votes. To analyze everything becomes almost an impossible task for a single citizen who is out of the loop.

To make it easier there are journalists, politologists and other personas who review new legislation, current issues. Delegates themselves also do press conferences, debates where they state their intentions, comment on their actions. However, this requires citizens to trust that journalists and delegates only state truth, don't omit important information and don't have other hidden agendas. Study done about intrinsic honesty showed that the more society is corrupt - the more people lie in a simple dice game. This applies to politicians too and citizens trust in parliament is relatively low. Therefore, anything that can be done to better observe representatives is useful.

This thesis goal is to X and Y

2 Analysis

2.1 Literature analysis

There is a decent amount of previous work analyzing voting on roll call data. A huge part of this research is on specific elections that happened in the past.

In *Spatial Models of Parliamentary Voting* [1] author discusses how voter's positions on specific issues can be captured by his position on one or two dimensions such as liberalism or conservatism. This constraint means there are two spaces - one with few dimensions - basic or ideological. The other - high dimensional space which represents remaining issues. This breakthrough might suggest Multidimensional scaling as a good performance method for visualization and analysis as majority of data is encoded in few dimensions.

There is also research done specifically on Parliament of the Republic of Lithuania (LRS) data. One such is *On Structural Analysis of Parliamentarian Voting Data* [2]. In this paper authors discuss about data reduction to dissimilarity matrix, vote encoding, Multidimensional scaling (MDS) and its performance on specific dimensions. Authors focus on specific elections and term of office which is different from our goal. However, methods discussed and research results are relevant for this project.

In *The new Voteview.com: preserving and continuing Keith Poole's infrastructure for scholars, students and observers of Congress* [3] paper, authors discuss famous *Voteview.com* website. While website's primary goal is to provide open data access is different from ours - it contains useful information about how specific methods are used, how visualizations work. It also contains visualization which shows how data changed over time - how ideology and party composition changes.

2.2 Materials and methods

2.2.1 Lithuania's parliament open data analysis

In this section open data from XML is reviewed. Only data and properties important to the project are included and commented on. All data is imported into MySQL database, therefore to demonstrate structure tables are used. Field types are inferred empirically by looking at data, therefore might not be correct in some cases. Moreover, data itself is not consistent through all years of Parliament of the Republic of Lithuania activity.

Table 1 contains term of office table structure.

In XML	In database	Type	Comments
kadencijos_id	term_of_office_id	int(11)	Unique term of office id
pavadinimas	name	varchar(255) not null	Name
data_nuo	date_from	date not null	Date when term of office begins
data_iki	date_to	default null	Date when term of office ends

Table 1: Term of office table structure

Table 2 contains parliament sessions table structure.

In XML	In database	Type	Comments
sesijos_id	session_id	int(11) not null	Unique parliament session id
kadencijos_id	term_of_office_id	int(11)	Unique term of office id
numeris	number	varchar(255) not null	Number
pavadinimas	name	varchar(255) not null	Session name
data_nuo	date_from	date not null	Date when session begins
data_iki	date_to	date default null	Date when session ends. Current session doesn't have this value set.

Table 2: Parliament session table structure

Table 3 contains parliament plenaries table structure.

In XML	In database	Type	Comments
posėdžio_id	plenary_id	int(11) not null	Unique plenary id
sesijos_id	session_id	int(11) not null	Unique parliament session id
numeris	number	varchar(255) not null	Number
tipas	plenary_type	varchar(255) not null	Plenary type
pradžia	time_start	datetime default null	Plenary starting time
pabaiga	time_finish	datetime default null	Plenary ending time. Some plenaries are not yet finished and some entries don't include times at all

Table 3: Plenary table structure

Table 4 contains parliament member table structure.

In XML	In database	Type	Comments
asmens_id	person_id	int(11) not null	Person id
-	unique_id	varchar(100) not null	Unique id for person, generated by script
vardas	person_name	varchar(255) not null	Member name
pavardė	person_surname	varchar(255) not null	Member surname
lytis	gender	varchar(1) not null	Gender
data_nuo	date_from	date not null	Date from inauguration
data_iki	date_to	date default null	Date when term ends. Current session doesn't have this value set.
iškėlusi_partija	faction_name	text	Faction name
išrinkimo_būdas	elected_how	varchar(255) not null	How member was elected
biografi-jos_nuoroda	biography_link	varchar(255) default null	Hyperlink to biography
-	term_of_office_id	int(11)	Unique term of office id
-	term_of_office_specific_id	int(11)	Specific term id used for computations
-	faction_id	int(11) not null	Faction id

Table 4: Parliament member table structure

Table 5 contains agenda question table structure.

In XML	In database	Type	Comments
darbotvarkės_klausimo_id	agenda_question_id	int(11) not null	Agenda question id
-	agenda_question_group_id	varchar(255) not null	Group id
pavadinimas	title	text	Title
laikas_nuo	time_from	time default null	Question start time
laikas_iki	time_to	time default null	Question end time
-	datetime_from	datetime default null	Custom generated date time
-	datetime_to	datetime default null	Custom generated date time
data	date	date not null	Date
pavadinimas	raw_status	varchar(255) not null	Status as taken from XML
-	status	int(5) default null	Status encoded
dokumento_nuoroda	document_link	varchar(255) default null	Hyperlink to document
-	speakers	text	Speakers list converted from XML to single text field with separators
numeris	number	varchar(255) not null	Number
-	plenary_id	int(11) not null	Plenary id

Table 5: Agenda question table structure

Table 6 contains plenary question table structure.

In XML	In database	Type	Comments
darbotvarkės_klausimo_id	agenda_question_id	int(11) not null	Agenda question id
-	plenary_question_group_id	varchar(255) not null	Group id
pavadinimas	title	text	Title
laikas_nuo	time_from	time default null	Question start time
-	datetime_from	datetime default null	Custom generated date time
pavadinimas	raw_status	varchar(255) not null	Status as taken from XML
-	status	int(5) default null	Status encoded
numeris	number	varchar(255) not null	Number
-	plenary_id	int(11) not null	Plenary id

Table 6: Plenary question table structure

Table 7 contains discussion event table structure.

In XML	In database	Type	Comments
-	agenda_question_id	int(11) not null	Agenda question id
-	unique_id	varchar(255) not null	Custom generated unique id
laikas_nuo	discusstion_time_from	time default null	Event start time
asmens_id	person_id	nt(11) not null	Person id
balsavimo_id	vote_id	int(11) not null	Vote id
balsavimo_tipas	vote_type	int(5) default null	Vote type
-	plenary_id	int(11) not null	Plenary id

Table 7: Discussion event table structure

Table 8 contains vote table structure.

In XML	In database	Type	Comments
-	vote_person_id	int(11) not null	Custom generated unique id for vote
balsavimo_laikas	time	time DEFAULT not null	Voting time
balsavo	vote_total	int(11) not null	How many voted
viso	vote_total_max	int(11) not null	How many can vote
už	vote_for	int(11) not null	Voted for
prieš	vote_against	int(11) not null	Voted against
susilaikė	vote_abstain	int(11) not null	Voted abstain
komentaras	comment	text	Vote comment
asmens_id	person_id	int(11) not null	Person id
asmeyns_vardas	person_name	varchar(255) not null	Person name
asmens_pavarde	person_surname	varchar(255) not null	Person surname
kaip_balsavo	vote	int(11) not null	Vote itself
-	vote_person_id	int(11) not null	Custom generated unique id for vote
frakcija	faction_acronym	varchar(255) default null	Acronym for faction
balsavimo_id	vote_id	int(11) not null	Vote id
balsavimo_tipas	vote_type	int(5) default null	Vote type
-	vote_id	int(11) not null	Vote id
-	plenary_id	int(11) not null	Plenary id

Table 8: Votes table structure

2.3 Problem analysis

Problem can be divided into two pieces:

- Research part, including MDS and k means classification
- Software development

If we consider what is the output of parliaments during their term of office in parliament - it would be their voting outcomes. Let's say we have a set of all votes made by parliament members $V = \{v_1, v_2, ...v_n\}$ where each vote v_i has a tuple of parameters $P = \{timestamp, term\ of\ office, parliament\ member\ id, voting\ outcome, ...\}$. Our goal is visualize set V in a way that similar voting patterns of different members are visible.

With MDS XX

With k means XX

2.4 Method analysis

2.4.1 MDS

2.4.2 Unsupervised learning: *k*-means clusterization

Due missing labels of this dataset we are forced to look into unsupervised machine learning methods. *k-means* clusterization [**?**] is one such method and it fits well with our features. This is due TFIDF which transforms text into vectors be used with *k-means*. For distance between comments we can use cosine similarity [**?**] due its nature of performing better when comparing texts. To compare we can try with Euclidian and Manhattan distances too.

2.5 Visualizations

3 Software design

3.1 Components of system

3.1.1 Data flow diagram

Data flow diagram

3.2 Tools

3.2.1 Database: *M*ySQL

3.3 Downloader

3.4 Coordinator

3.5 API server

3.6 Repository

3.7 User interface (Frontend)

4 Description of experimental research

4.1 Data

4.1.1 Data semantics

4.1.2 Data statistics

4.2 Experiments

4.2.1 MDS

4.2.2 Unsupervised learning: *k*-means clusterization

4.3 Results

References

[1] K. T. Poole, *Spatial Models of Parliamentary Voting*. Analytical Methods for Social Research, Cambridge University Press, 2005.

[2] T. Krilavicius and A. Zilinskas, "On structural analysis of parliamentarian voting data," *Informatica, Lith. Acad. Sci.*, vol. 19, no. 3, pp. 377–390, 2008.

[3] A. Boche, J. B. Lewis, A. Rudkin, and L. Sonnet, "The new voteview.com: preserving and continuing keith poole's infrastructure for scholars, students and observers of congress," *Public Choice*, vol. 176, 05 2018.

A Appendix