Exploring U.S. Congressional Voting Behavior Using Bayesian Models

Ana Mateos Mata

January 2025

1 Introduction

In recent years, machine learning has revolutionized multiple disciplines, including political science. Its application spans across areas such as political communication, public policy, international political economy, and even security and conflict studies. By leveraging advanced techniques, researchers have been able to analyze large datasets, uncovering complex patterns and insights that would otherwise remain hidden [3] [5] [2].

The voting behavior of the U.S. Congress, a cornerstone of the nation's political system, is one such area where machine learning can provide valuable insights. The two major political parties—Democrats and Republicans—hold divergent political ideologies that often guide how their members vote on key issues. This ideological divide offers a unique opportunity to predict voting behavior based on party affiliation, even when individual votes remain anonymous.

This study delves into the 1984 U.S. Congressional Voting Records dataset, which compiles the votes of 435 anonymous congress members on 16 critical issues. By analyzing this dataset, we aim to reveal underlying patterns in party affiliation and voting tendencies. Specifically, we apply Bayesian models to uncover how party alignment influences voting decisions and to predict the potential outcomes of future legislative decisions. Our goal is to demonstrate how machine learning, particularly Bayesian approaches, can provide a deeper understanding of political behavior and offer predictive power in the realm of U.S. politics.

2 Database

2.1 The 1984 Congressional Voting Records

The 1984 Congressional Voting Records dataset offers a detailed account of how U.S. Representatives voted during the 98th Congress (second session). The data was sourced from the Congressional Quarterly Almanac (CQA) of 1985, which identified 16 critical votes that shaped the political landscape of the time. These votes were categorized into three main groups, simplifying the original nine types of voting actions identified by the CQA. The dataset also includes party affiliation as a binary variable, making it possible to analyze voting behavior in relation to political party lines.

2.2 Voting Behavior Categorization and Simplifications

Voting behavior in Congress is a nuanced process, with members choosing from several voting options when deciding on legislation. The CQA originally recognized nine distinct voting actions, but for simplicity and analytical efficiency, these have been reduced to three main categories in the dataset:

- **Yea**: This includes votes for the bill, paired votes for, and announced votes for.
- **Nay**: This includes votes against the bill, paired votes against, and announced votes against.
- **Unknown**: This covers abstentions, votes made to avoid conflicts of interest, and instances where no vote or position is recorded.

While these simplifications make the dataset more manageable, they also omit finer details that could offer deeper insights into voting behaviors. The original nine categories included additional classifications such as "voted present" or "did not vote," which have now been grouped under the "Unknown" category in the dataset.

3 Software Used

For this analysis, we used `pgmpy`, a powerful Python library tailored for probabilistic graphical models. `pgmpy` [1] offers a comprehensive set of tools for constructing, learning, and performing inference on Bayesian Networks. This library was chosen for its user-friendly interface, extensive documentation, and robust capabilities in both structure and parameter learning. Its flexibility and ease of integration with other Python libraries make it an ideal choice for our Bayesian modeling tasks.

4 Exploratory Data Analysis (EDA)

4.1 Class Imbalance

The dataset consists of 435 instances, representing votes from members of the U.S. Congress, categorized into two classes: Democrats and Republicans. Of these instances, 267 belong to the Democrat class, while 168 belong to the Republican class. This distribution reveals a class imbalance, with the Democrat class being overrepresented. Such imbalances can impact model performance, as predictive algorithms may become biased toward the majority class, leading to skewed results. Therefore, it is important to consider techniques for handling class imbalance, such as resampling or adjusting class weights during model training.

4.2 Correlation Analysis

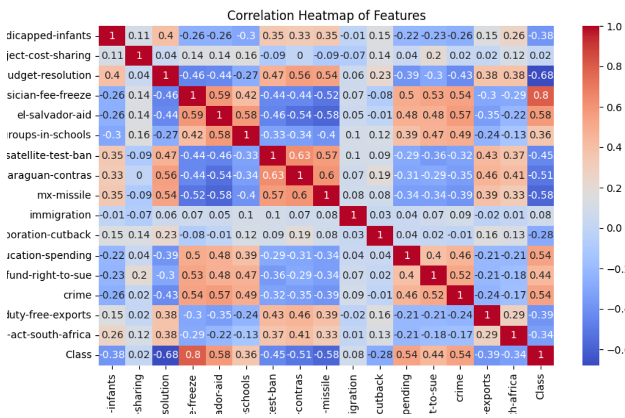

Figure 1: Correlation Matrix of Features

The correlation matrix shown in Figure 1 illustrates the relationships between each feature and the target variable (Class). The analysis reveals several key insights that can inform model development. Strong positive correlations with the target variable are observed for features like physician-fee-freeze (0.91), el-salvador-aid (0.71), *education-spending* (0.69), crime (0.61), and superfund-right-to-sue (0.54). These variables exhibit strong associations with specific voting outcomes, suggesting their significant predictive power.

Similarly, negative correlations are equally informative. Features such as adoption-of-the-budget-resolution (-0.74), aid-to-nicaraguan-contras (-0.65), mx-missile (-0.62),anti-satellite-test-ban (-0.51), and duty-free-exports (-0.52) have strong negative correlations with the target variable. These suggest that opposition or absence of these issues may align with specific class outcomes, making them critical for understanding voting behavior.

On the other hand, certain features such as water-project-cost-sharing (0.00), immigration (0.08), and export-administration-act-south-africa (-0.22) display weak or negligible correlations with the target variable. These variables may have little impact on voting behavior, and their inclusion in the model could potentially decrease its effectiveness. Therefore, it is advisable to carefully evaluate these features before deciding to incorporate them into predictive models.

5 Data Preprocessing

5.1 Handling Missing Values

The dataset contains relatively few missing values, with most attributes having fewer than 30 missing entries out of the 435 total instances, which accounts for less than 7% missing data per feature. Notably, columns 1 and 2 have no missing values. Additionally, several other columns—such as attributes 3, 7, and 8—have only minor gaps, with fewer than 17 missing values.

5.1.1 Strategies for Handling Missing Values in Voting Data

Missing values, particularly abstentions in voting records, present a unique challenge in political data analysis [4]. Abstentions are not simply absent data points; they reflect significant political decisions that require careful consideration. In the context of Congressional voting, abstentions may be strategically motivated—such as to avoid conflicts with party leadership or to appeal to specific constituency preferences.

In the case of abstentions, treating them as mere missing values could misrepresent the voting context. Abstentions are unlikely to reflect a lack of opinion or availability but instead could indicate tactical political decisions. Simplifying abstentions as missing data could lead to distorted interpretations, particularly for votes that carry significant political weight. Thus, it is crucial to consider the strategic nature of abstentions when deciding how to handle them in data preprocessing. Therefore, absents where treated as another category.

6 Structure Planning

The process of learning the network structure from data is crucial for accurately capturing the relationships between variables. In this study, two different structure learning algorithms were compared: the PC algorithm and Hill Climb Search. These methods were evaluated using several scoring techniques, including BIC, K2, BDeu, and BDs. Below, we explain the evaluation methods and how the best model was selected based on these scores.

6.1 Evaluation of the Methods

In this study, the best model was selected based on several evaluation techniques, including the log-likelihood score, correlation scores, Fisher's C, and various structure scores such as K2, BDeu, BDs, and BIC. These metrics were computed for each model, including those generated by the PC algorithm and the Hill Climb search method for different scoring methods (BIC, K2, BDeu, and BDs). Each evaluation method provides unique insights into the performance of the models, and together, they form a comprehensive picture of how well the models fit the data.

- **Log-Likelihood Score**: This score measures how well a model's parameters fit the data. Higher values indicate a better fit. In the context of Bayesian networks, the log-likelihood score calculates the probability of observing the data given the model. A model with a higher log-likelihood suggests that it is more likely to have generated the observed data.
- **Correlation Scores**: The correlation score measures the alignment between the model's predicted conditional independence relationships and the true relationships in the data. A higher correlation score indicates that the model more accurately represents the dependencies among variables. It reflects the overall quality of the model in terms of its ability to capture the underlying structure of the data.
- **Fisher's C**: Fisher's C test evaluates how well a given model fits the data in terms of statistical independence between variables. A lower value of Fisher's C indicates a better fit because it suggests fewer violations of independence assumptions. This is particularly useful when comparing models that aim to capture the conditional independencies between the variables.
- **K2, BDeu, BDs, and BIC Scores**: These are scoring methods used in the context of structure learning. Each method evaluates the structure of the Bayesian network in different ways.
  - **K2**: A score based on the likelihood of the data, which penalizes complex models with many parameters. It is useful when the goal is to strike a balance between fit and model complexity.
  - **BDeu**: A Bayesian Dirichlet equivalent uniform prior score that incorporates prior knowledge about the structure of the model, aiming to capture the most probable structure given the data.
  - **BDs**: The Bayesian Dirichlet score with a slight variation, designed to provide a robust evaluation by incorporating prior distributions.
  - **BIC (Bayesian Information Criterion)**: A penalty-based score that balances model fit and complexity, where lower BIC values indicate a better model. It is commonly used to prevent overfitting by penalizing the inclusion of too many parameters.

Score Method Model	Number of Edges	Log-Likelihood Scores	Correlation Scores	Fisher's C	k2	bdeu	bds	bic
BIC	17	-4348.815021	0.000000	0.000000e+00	-4.574580e+03	-4582.117516	-4.688169e+03	-4.649545e+03
K2	117	-2651.431347	0.000000	5.831104e-04	2.982926e+07	-7322.512028	-1.478398e+08	-2.615133e+08
BDeu	30	-4089.303682	0.000000	2.341967e-08	-4.498432e+03	-4479.501010	-4.614306e+03	-4.833534e+03
BDs	29	-4099.487793	0.000000	1.087064e-11	-4.492308e+03	-4492.758625	-4.610651e+03	-4.807266e+03
PC	9		0.092308	0.000000e+00	-5.310152e+03	-5325.158266	-5.425665e+03	-5.362571e+03

Figure 2: Model Comparison: Number of Edges, Log-Likelihood, Correlation Scores, Fisher's C, and Structure Scores

The K2 method, with the highest log-likelihood score (-2651.43), demonstrates the best fit to the data but results in a more complex model with 117 edges. This might indicate overfitting due to its high complexity. In contrast, the PC method is much simpler with only 9 edges, suggesting a more parsimonious model. However, it has a lower log-likelihood score and does not have a calculated value for the log-likelihood, indicating that the fit is not as strong as K2's.

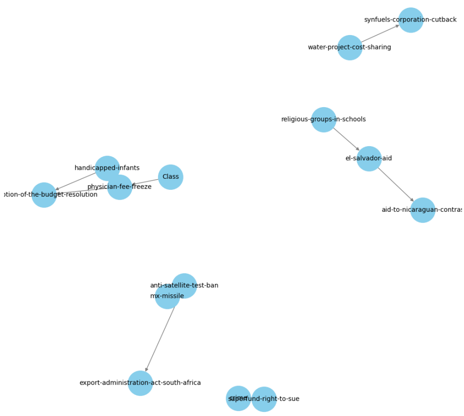

Figure 3: PC Algorithm network Structure

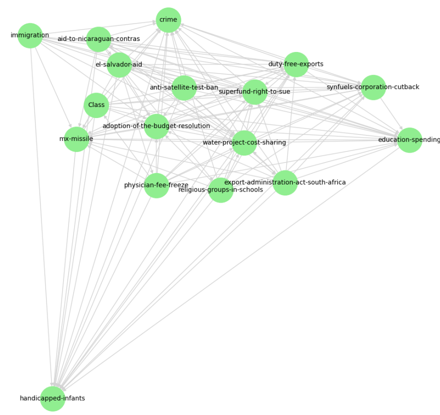

Figure 4: K2 Algorithm network Structure

The Fisher's C statistic is close to zero for all models except PC, which shows a small non-zero value (0.0923), suggesting a slight improvement in explaining the relationships in the data.

Regarding the structure scores (k2, bdeu, bds, bic), K2 scores significantly worse, indicating an overly complex structure. BIC, in contrast, shows moderate values and penalizes complexity, making it a balanced choice between fit and simplicity. PC also provides a more straightforward model with moderate structure scores, making it suitable if simplicity is prioritized.

In conclusion, K2 is the best option if model fit is prioritized, but it comes at the cost of complexity. On the other hand, if a simpler and more interpretable model is required, PC or BIC could be better choices, with PC offering simplicity at the expense of fit, and BIC providing a balanced trade-off between fit and complexity.

From the plots it can be interpreted that the PC algorithm provides a sparser, more interpretable, and potentially causal network, which is preferable for understanding the underlying data-generating processes. In contrast, the K2 model captures a denser structure that might be useful for exploratory analysis but could include redundant or spurious relationships. But PC may loose a lot of information, so the inference will be performed with K2 based model.

In the K2 Hill Climbing graph, Class is densely connected to multiple variables, including adoption-of-the-budget-resolution, handicapped-infants, physician-fee-freeze, mx-missile, immigration, and aid-to-nicaraguan-contras, suggesting a broad range of potential influences but likely including spurious relationships. In contrast, the PC algorithm identifies only three variables—adoption-of-the-budget-resolution, handicapped-infants, and physician-fee-freeze—as directly affecting Class, focusing on statistically robust and likely causal relationships. This highlights the PC algorithm's emphasis on interpretability and causal insights, whereas K2 captures a more comprehensive but noisier set of dependencies.

K2 Algorithm:

The K2 score is a scoring metric used in the K2 algorithm for structure learning in Bayesian Networks. It evaluates a given network structure by calculating the likelihood of the data given the structure and incorporates a penalty term for the number of parameters in the network. The goal is to maximize the K2 score, balancing model complexity and fit to the data. The K2 score is computed by iteratively adding directed edges to the network and calculating the score for each possible structure, selecting the one with the highest score.

7 Parameter Training

The process of parameter training involves estimating the parameters of the model, particularly the Conditional Probability Distributions (CPDs), given the structure of the Bayesian Network. In this case, the `MaximumLikelihoodEstimator` is employed to estimate the parameters based on the data.

The following steps describe the process of parameter training:

1. A *Bayesian Network* model is first constructed using the previously determined edges (the relationships between variables) from a structure-learning algorithm.

2. The model is then fitted to the data using the `fit()` function. This process estimates the CPDs of the network.

3. `MaximumLikelihoodEstimator` is used as the estimator. This method computes the maximum likelihood estimate of the parameters for each CPD, given the data.

8 Inference

Bayesian Networks enable efficient probabilistic inference. We used the Variable Elimination algorithm to query the prior probability distribution of the target variable, 'Class', without any evidence. This provides insights into the model's initial belief, based on learned data structure and parameters. The model assigns a 61.38% probability to 'Class' being 'democrat' and 38.62% to 'republican'.

CPD for water-project-cost-sharing:

Class	...	Class(republican)
adoption-of-the-budget-resolution	...	adoption-of-the-budget-resolution(y)
aid-to-nicaraguan-contras	...	aid-to-nicaraguan-contras(y)
anti-satellite-test-ban	...	anti-satellite-test-ban(y)
el-salvador-aid	...	el-salvador-aid(y)
export-administration-act-south-africa	...	export-administration-act-south-africa(y)
immigration	...	immigration(y)
physician-fee-freeze	...	physician-fee-freeze(y)
religious-groups-in-schools	...	religious-groups-in-schools(y)
water-project-cost-sharing(n)	...	0.6666666666666666
water-project-cost-sharing(p)	...	0.0
water-project-cost-sharing(y)	...	0.3333333333333333

Figure 5: CDP parameters example for water-project-cost-sharing

8.1 Advanced Queries

8.1.1 Impact of Economic Sanctions on Military and Foreign Relations

Economic sanctions, such as the `export-administration-act-south-africa` and `duty-free-exports`, influence political decisions. This query investigates how the probability of `mx-missile` changes when these sanctions are voted for (set to y).

The query was formulated as:

```
prob_mx_missile_given_sanctions = infer.query(variables=['mx-missile'],
evidence={'export-administration-act-south-africa': y, 'duty-free-exports': y})
```

Results show that when sanctions are voted for, the probability of a stronger missile defense stance (`mx-missile` = y) increases to 81.6%, suggesting that economic sanctions can drive more aggressive defense policies.

8.1.2 Relationship Between Religious Influence and Education Spending

We explored how education spending (voted for, set to y) correlates with religious groups in schools. This query investigates the probability of `religious-groups-in-schools` when education spending is high.

The query was formulated as:

```
prob_religion_given_education = infer.query(variables=['religious-groups-in-schools'],
evidence={'education-spending': y})
```

Results show that with high education spending (`education-spending` = y), the probability of having religious groups in schools (`religious-groups-in-schools`

= y) is 93.34%, suggesting that increased funding for education may promote religious influences in schools.

8.1.3 Conditional Impact of Immigration on Other Policies

Immigration policies might affect social welfare programs. Here, we query the impact of immigration on policies like `handicapped-infants` and `adoption-of-the-budget-resolution` when immigration is voted for (set to y).

The query was formulated as:

```
prob_joint_immigration = infer.query(variables=['handicapped-infants',
'adoption-of-the-budget-resolution'], evidence={'immigration': y})
```

Results show that immigration policy influences welfare programs and budget resolutions, with significant effects on the likelihood of adopting policies for handicapped infants and the budget resolution.

8.1.4 Probabilistic Interdependence Between Military and Social Policies

Exploring how military policies, such as `mx-missile` and `anti-satellite-test-ban`, interact with social policies like `handicapped-infants` and `crime`, can uncover interesting dependencies. These interdependencies could reveal whether defense spending and military strategies are linked to social welfare issues or public safety concerns. In this query, we aim to understand the joint probability distribution of these military and social policies.

The query was formulated as follows:

```
prob_military_social_interdependence = infer.query(variables=['mx-missile',
'anti-satellite-test-ban', 'handicapped-infants', 'crime'])
```

The results from the joint probability distribution suggest that military policies, such as mx-missile and anti-satellite-test-ban, are probabilistically interdependent with social policies like handicapped-infants and crime. For example, when the policy on mx-missile is set to y (stronger stance on missile defense), the probability of handicapped-infants also being y (adoption of policies for handicapped infants) increases to 0.1985. Similarly, the probability of crime being y (crime-related policy) is higher (0.2575) when both mx-missile and anti-satellite-test-ban are set to n.

This suggests that military policies might influence the adoption of social policies, reflecting a pattern where certain policy combinations, like mx-missile and crime, are more likely to occur together. The findings indicate that changes in one policy (military or social) can influence the probability distribution of other policies, illustrating the interdependence between military and social policies.

8.1.5 Multi-Policy Influence on "Class"

It is useful to examine how different policies affect the probability distribution of `Class`. This query explores how a combination of environmental and social policies may intersect at a social-political level and influence the political or social class of an individual. Specifically, investigate how the combination of policies like `water-project-cost-sharing`, `education-spending`, and `superfund-right-to-sue` impacts the likelihood of an individual belonging to a certain class.

The query was formulated as follows:

```
prob_class_given_multiple_policies = infer.query(variables=['Class'],
evidence={'water-project-cost-sharing': 'y', 'education-spending': 'y', 'superfund-right-to-
```

The results show that when all three policies (`water-project-cost-sharing`, `education-spending`, and `superfund-right-to-sue`) are set to `y`, the probability of the individual being a `democrat` is 17.75%, while the probability of the individual being a `republican` is 82.25%. This analysis demonstrates how multi-policy influences can shape political alignment, providing insights into how certain social and environmental policies might influence the political or social class of an individual.

9 Conclusion

This study demonstrates the effectiveness of Bayesian models in understanding and predicting voting behavior in the U.S. Congress. By leveraging the 1984 Congressional Voting Records dataset, we were able to explore the relationship between party affiliation and legislative decisions, uncovering critical insights into voting tendencies. The results highlight several key takeaways:

Methodological Insights: The comparison of structure learning methods, such as the PC algorithm and the K2 Hill Climbing approach, revealed the trade-offs between simplicity and predictive power. While the PC algorithm produced a more interpretable, sparse network, the K2 model offered a denser structure that captured a broader range of dependencies. Despite its complexity, the K2 model was selected for inference due to its superior fit to the data.

1. Feature Relevance: Strong correlations between certain features (e.g., physician-fee-freeze, el-salvador-aid, and education-spending) and party affiliation emphasize the importance of specific legislative issues in determining voting behavior. Conversely, weaker correlations for features like water-project-cost-sharing indicate that not all variables significantly influence outcomes, highlighting the need for feature selection in future analyses.

2. Advanced Queries and Interpretations: The probabilistic inference enabled by Bayesian networks uncovered nuanced relationships between policies. For example, the study revealed how economic sanctions influence defense policies, how education spending correlates with religious influences

in schools, and how immigration impacts social welfare and budgetary decisions. These findings underscore the utility of Bayesian networks for exploring complex, conditional relationships in political data.

3. Implications for Political Science: The ability to probabilistically model the interdependence of military and social policies suggests that voting behavior is shaped by a combination of ideological and practical considerations. Such insights can aid policymakers, political analysts, and researchers in anticipating legislative outcomes and understanding the dynamics of congressional decision-making.

By integrating machine learning techniques with political science, this study highlights the potential of Bayesian models as a powerful tool for analyzing voting behavior. Future research could expand on this work by incorporating temporal data, exploring other legislative datasets, or applying alternative probabilistic frameworks. Ultimately, this approach contributes to a deeper understanding of the factors driving political behavior and offers a foundation for predictive analytics in the field of political science.

References

[1] Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the Python in Science Conference*, SciPy. SciPy, 2015.

[2] Zeliha Khashman and Adnan Khashman. Anticipation of political party voting using artificial intelligence. In *Proceedings of the 12th International Conference on Application of Fuzzy Systems and Soft Computing (ICAFS 2016)*, Vienna, Austria, August 29-30 2016. ICAFS. Conference held in Vienna, Austria.

[3] Keith T. Poole and Howard Rosenthal. Patterns of congressional voting. *American Journal of Political Science*, 35(1):228–278, 1991. Accessed: 03-01-2025.

[4] Christoph Waldhauser and Ronald Hochreiter. Shaking the trees: Abilities and capabilities of regression and decision trees for political science. *ITM Web of Conferences*, 14(9):1–16, 2017.

[5] Łukasz Wordliczek. Neural networks and political science: Testing the methodological frontiers. *Empiria. Revista de metodología de ciencias sociales*, 57:37–62, 2023.