Sub-Matrix Factorization for Real-Time Vote Prediction

Alexander Immer*
Ecole polytechnique fédérale de Lausanne

Victor Kristof*
Ecole polytechnique fédérale de Lausanne

Matthias Grossglauser
Ecole polytechnique fédérale de Lausanne

Patrick Thiran
Ecole polytechnique fédérale de Lausanne

ABSTRACT

We address the problem of predicting aggregate vote outcomes (*e.g.*, national) from partial outcomes (*e.g.*, regional) that are revealed sequentially. We combine matrix factorization techniques and generalized linear models (GLMs) to obtain a flexible, efficient, and accurate algorithm. This algorithm works in two stages: First, it learns representations of the regions from high-dimensional historical data. Second, it uses these representations to fit a GLM to the partially observed results and to predict unobserved results. We show experimentally that our algorithm is able to accurately predict the outcomes of Swiss referenda, U.S. presidential elections, and German legislative elections. We also explore the regional representations in terms of ideological and cultural patterns. Finally, we deploy an online Web platform (www.predikon.ch) to provide real-time vote predictions in Switzerland and a data visualization tool to explore voting behavior. A by-product is a dataset of sequential vote results for 330 referenda and 2196 Swiss municipalities.

ACM Reference Format:
Alexander Immer, Victor Kristof, Matthias Grossglauser, and Patrick Thiran. 2020. Sub-Matrix Factorization for Real-Time Vote Prediction. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403277

1 INTRODUCTION

The past decade has seen the emergence of several open-government initiatives for the increase of administration transparency through the publication of governmental data. These data are of great interest to parties, companies, sub- and supra-government entities, researchers, and citizens. In particular, the results of referenda and election ballots in municipalities, districts, states, and countries are valuable for understanding the structure and the dynamics of politics.

In this paper, we address the problem of vote prediction when only partial results are available. The ability to predict the outcome of votes both before and during ballot counting is relevant to political parties, interest groups, polling agencies, news outlets, government authorities, and interested citizens. These predictions help uncover voting patterns, *e.g.*, to identify swing regions, to understand voting behaviours, and to detect fraud. Political parties and interest groups can enhance their campaigning efforts. Polling agencies and news outlets can optimize their surveying efforts. Authorities can monitor the smooth functioning of the voting process.

We focus on national vote predictions during the ballot counting, *i.e.*, after all eligible voters have cast their ballots, as government officials start count the valid votes in each region. We predict national results by using sequential regional results, and we seek to obtain accurate predictions as early as possible, *i.e.*, with a minimum number of regional results. Typically, less populated regions release their official counts earlier than more populated ones. Regions where remote voting is allowed release their results earlier than regions where this is not allowed. In some countries, for example in the U.S., some regions vote earlier than others by design. We will show that our model is able to exploit the correlations between regions and between votes to obtain accurate early predictions.

Switzerland offers a fascinating laboratory for vote prediction due to its direct-democracy system. Swiss citizens are called to vote four times a year on referenda and popular initiatives [29, 30]. As a result, the amount and frequency of voting data produced in Switzerland remains unmatched by any other country. We take Switzerland as an example to develop our methodology but, as shown in Section 3, our algorithm can be applied to other countries and in other settings.

In Section 2, we propose an algorithm to predict national vote outcomes from a sequence of regional vote results. Our model has two components: First, it learns the correlations between regions and between votes from historical data by using singular value decomposition (SVD). Second, after observing at least one regional result for a new vote, it uses the SVD as input features to a generalized linear model (GLM) to predict the unobserved regional results. The national outcome is then easily obtained by weighted aggregation of the predicted and the observed regional results. The SVD, computed only once on the historical data, is inexpensive in terms of complexity and enables interpretation. By using different likelihoods in the GLM, we gain flexibility in predicting binary outcomes (for votes) or categorical outcomes (for elections).

For Swiss votes, where people must answer "Yes" or "No" on each ballot, we show that a Gaussian and a Bernoulli likelihood provide the best performance. We also explore what the SVD offers in terms of interpretation of voting patterns. Furthermore, we show that we can predict the outcome of the popular vote of a U.S. presidential election by casting this problem as a binary choice between two candidates. We predict the outcome of parliamentary elections in Germany, where people must choose between five political parties, using a categorical likelihood. We describe our experiments on state-level and district-level results in Section 3.

*Both authors contributed equally to this research.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403277

Table 1: List of GLMs. The softmax function is denoted by $\mathcal{S}$.

Distrib.	Link g	θ	μ	$\mathcal{D}$
$\mathcal{N}(\mu, \sigma^2)$	Identity	$\theta = \mu$	$\mu = \boldsymbol{x}^\top \boldsymbol{w}$	$\mathbf{R}$
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Identity	$\boldsymbol{\theta} = \boldsymbol{\mu}$	$\boldsymbol{\mu} = X\boldsymbol{w}$	$\mathbf{R}^K$
$\mathrm{Ber}(\mu)$	Logit	$\theta = \mathrm{logit}(\mu)$	$\mu = \sigma(\boldsymbol{x}^\top \boldsymbol{w})$	$[0, 1]$
$\mathrm{Cat}(\boldsymbol{\mu}, K)$	Inv. softmax	$\boldsymbol{\theta} = \mathcal{S}^{-1}(\boldsymbol{\mu})$	$\boldsymbol{\mu} = \mathcal{S}(X\boldsymbol{w})$	$[0, 1]^K$

We also deploy a Web platform available to the general public to provide vote prediction for Switzerland. Using an API developed by the Swiss government, we are able to make real-time predictions during the official counting with partial regional results. We also provide a data-visualization tool to explore voting patterns and to understand how our model makes predictions. We describe our platform in Section 4.

In summary, our contributions are as follows: We propose an efficient, flexible, and accurate algorithm for predicting the national outcome of a referendum or an election from early regional results. We curate a new dataset of sequential vote results in Switzerland, covering 330 votes and 2 196 regions between 1981 and 2020. We deploy an interactive Web platform to display real-time vote predictions in Switzerland, together with tools to explore and visualize our dataset. The data and the code are available on github.com/indy-lab/submatrix-factorization and the Web platform is available on www.predikon.ch.

2 METHODOLOGY

2.1 Generalized Linear Models

Generalized linear models (GLMs) are probabilistic models whose likelihood belongs to the exponential family. Let $\boldsymbol{x} \in \mathbf{R}^D$ be some D-dimensional features, $\boldsymbol{w} \in \mathbf{R}^D$ be some D-dimensional parameters, and $y \in \mathcal{D}$ be an observation in a given domain $\mathcal{D}$. Let $h(y) \in \mathbf{R}$ be a scaling factor, $\theta := \boldsymbol{x}^\top \boldsymbol{w} \in \mathbf{R}$ be the natural parameter, and $A(\theta) \in \mathbf{R}$ be the log-partition function. Then, the likelihood of a GLM is

$$p(y|\boldsymbol{w}, \boldsymbol{x}) = h(y) \exp\{y\theta - A(\theta)\}. \quad (1)$$

Point-wise predictions are obtained from the mean parameter

$$\mu = \mathbb{E}[y] = A'(\theta) = g^{-1}(\theta),$$

where the invertible function $g : \mathcal{D} \longrightarrow \mathbf{R}$ is called the link function. This function links the natural parameter and the mean parameter. The choice of link function depends on the choice of distribution in the GLM. Equation (1) can be easily generalized to K outputs $\boldsymbol{y} \in \mathcal{D}$ (*e.g.*, for multi-party elections) by setting the domain $\mathcal{D}$ to be K-dimensional. One advantage of GLMs is that they can be efficiently fit to data by using convex optimization methods [7]. In Table 1, we summarize four popular GLMs and their corresponding link functions, natural parameters, mean parameters, and support of g. We will use these models in our algorithm to predict referenda and elections, as described in the next sections. We refer the curious reader to Murphy [24, Chapter 9] for a detailed introduction to GLMs.

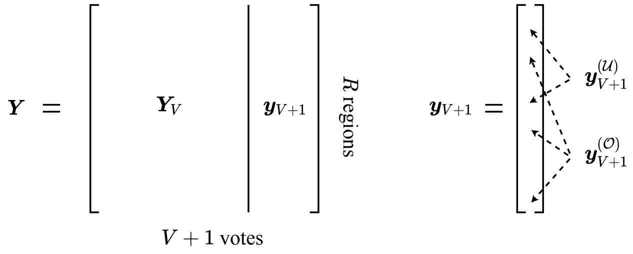

Figure 1: Decomposition of the vote matrix Y into the fully observed *sub-matrix* Y_V and the new vote $\boldsymbol{y}_{V+1}$, whose results arrive sequentially. The $(V + 1)$ votes are chronologically ordered and the R regions are arbitrarily ordered.

2.2 Problem Setup

Let $\boldsymbol{Y} \in \mathbf{R}^{R\times(V+1)}$ be the matrix of $(V+1)$ regional vote results in R regions, where a result is typically a fraction of votes. We assume the columns to be in chronological order. For a new, unobserved vote $V + 1$, we sequentially observe entries of the last column[1] in $\boldsymbol{Y}$, which we denote by $\boldsymbol{y}_{V+1}$. Let $Y_V \in \mathbf{R}^{R\times V}$ be the *sub-matrix* of all observed, historical results up to vote V. Denoting the set of consecutive integers by $[R] := \{1, 2, \ldots, R\}$, we define the set of *observed* indices for the new vote as

$$\mathcal{O} = \{r : r \in [R] \text{ and } y_{r,V+1} \in \mathbf{R}\},$$

and the set of *unobserved* indices (corresponding to values to be predicted) as

$$\mathcal{U} = \{r : r \in [R] \text{ and } y_{r,V+1} \equiv \emptyset\}.$$

Let $\boldsymbol{y}^{(\mathcal{O})}_{V+1}$ and $\boldsymbol{y}^{(\mathcal{U})}_{V+1}$ denote the observed and unobserved entries of $\boldsymbol{y}_{V+1}$, respectively. Our task is to predict the missing entries $\boldsymbol{y}^{(\mathcal{U})}_{V+1}$ from Y_V and $\boldsymbol{y}^{(\mathcal{O})}_{V+1}$ only. Figure 1 depicts the structure of the matrix Y.

To predict the missing entries of $\boldsymbol{y}^{(\mathcal{U})}_{V+1}$, Etter et al. [15] use standard matrix factorization with alternating least-squares (ALS) to minimize the non-convex loss based on the Frobenius norm

$$\min_{\boldsymbol{A},\boldsymbol{B}} \left\| Y^{(\mathcal{O})} - \left(\boldsymbol{A}\boldsymbol{B}^T\right)^{(\mathcal{O})} \right\|_F, \quad (2)$$

where $\boldsymbol{A} \in \mathbf{R}^{R\times D}$ and $\boldsymbol{B} \in \mathbf{R}^{V\times D}$ are two matrices of latent dimension $D \in \mathbf{N}_{>0}$, and where superscript $(\mathcal{O})$ denotes that, in this case, only the observed entries are kept. With ALS, each iteration is expensive, and there are neither convergence guarantees nor explicit convergence rates [4, 19]. According to the Eckart-Young-Mirsky Theorem [13], the optimal solution to Equation (2) is the SVD, which is only computable if $\boldsymbol{Y}^{(\mathcal{O})} = \boldsymbol{Y}$. We devise a more effective algorithm motivated by the special structure of this collaborative filtering problem[15].

2.3 Algorithm

Our algorithm works in four steps: First, the fully-observed sub-matrix $\boldsymbol{Y}_V$ is decomposed using SVD as

$$Y_V \approx \boldsymbol{U}\Sigma\boldsymbol{V}^\top, \quad (3)$$

[1] This problem can be trivially generalized to multiple unobserved columns $\{\boldsymbol{y}_{V+1}, \boldsymbol{y}_{V+2}, \cdots\}$.

where the diagonal matrix $\Sigma \in \mathbf{R}^{D \times D}$ stores the singular values, and where the matrices $\boldsymbol{U} \in \mathbf{R}^{R \times D}$ and $\boldsymbol{V} \in \mathbf{R}^{V \times D}$ store the D left and right singular vectors with the highest singular values, respectively.

Second, we compute the projection of the regions into the vote space as

$$\boldsymbol{X} = \boldsymbol{Y}_V \boldsymbol{V} = \boldsymbol{U}\Sigma, \tag{4}$$

where the matrix $\boldsymbol{X} \in \mathbf{R}^{R \times D}$ stores D-dimensional representations of the regions. We explore these representations in more detail in Section 3. These two steps are performed offline, *i.e.*, they are performed once.

Third, we use the observed results of a new vote $\boldsymbol{y}_{V+1}$ and the representations of observed regions in $\boldsymbol{X}$ to fit a GLM p. We find the maximum likelihood estimate $\boldsymbol{w}_* \in \mathbf{R}^D$ by minimizing the regularized negative log-likelihood of model p in Equation (1), with regularization parameter $\lambda \in \mathbf{R}$,

$$\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1}) = -\sum_{r \in O} \log p(y_{r,V+1} | \boldsymbol{w}, \boldsymbol{x}_r) + \lambda \|\boldsymbol{w}\|_2^2, \tag{5}$$

where $y_{r,V+1} \in \mathbf{R}$ is the result of the r-th (observed) region, and $\boldsymbol{x}_r \in \mathbf{R}^D$ is the r-th row of the representation matrix $\boldsymbol{X}$ corresponding to the representation of the r-th region.

Finally, we predict the unobserved regions of the new vote $\boldsymbol{y}_{V+1}^{(\mathcal{U})} \in \mathbf{R}^{|\mathcal{U}|}$ as the mean of the GLM p using the link function g. From the optimal parameters $\boldsymbol{w}_*$, we compute

$$\boldsymbol{y}_{V+1}^{(\mathcal{U})} := g^{-1}\left(\boldsymbol{X}^{(\mathcal{U})} \boldsymbol{w}_*\right), \tag{6}$$

where $\boldsymbol{X}^{(\mathcal{U})} \in \mathbf{R}^{|\mathcal{U}| \times D}$ are the representations of the unobserved regions. The prediction for the national outcome is then the average of $\boldsymbol{y}_{V+1}^{(O)}$ and $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$, weighted by the population of each region r. We summarize these steps in Algorithm 1.

Algorithm 1 SubSVD-GLM

```
Input: Sub-matrix $\boldsymbol{Y}_V$, partial results $\boldsymbol{y}_{V+1}$, and GLM $p$.
Output: Prediction of unobserved results $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$.
1: Decompose $\boldsymbol{Y}_V \approx \boldsymbol{U}\Sigma\boldsymbol{V}^\intercal$.                    ▷ Equation (3)
2: Project $\boldsymbol{X} = \boldsymbol{U}\Sigma$.                                   ▷ Equation (4)
3: Optimize $\boldsymbol{w}_* = \arg\min_{\boldsymbol{w}} -\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1})$.   ▷ Equation (5)
4: Predict $\boldsymbol{y}_{V+1}^{(\mathcal{U})} = g^{-1}\left(\boldsymbol{X}^{(\mathcal{U})}\boldsymbol{w}_*\right)$.            ▷ Equation (6)
```

To predict the outcomes of referenda and elections, we use the GLMs described in Table 1. For referenda, we use univariate Gaussian and Bernoulli likelihoods. For elections, we use multivariate Gaussian and categorical likelihood. When a univariate Gaussian likelihood is used, the optimal parameters $\boldsymbol{w}_*$ can be learned (step 3 of Algorithm 1) in closed form with least-squares

$$\boldsymbol{w}_* = \left(\boldsymbol{X}^{(O)\intercal}\boldsymbol{X}^{(O)} + \lambda \boldsymbol{I}_D\right)^{-1} \boldsymbol{X}^{(O)\intercal} \boldsymbol{y}_{V+1}^{(O)}, \tag{7}$$

where $\boldsymbol{X}^{(O)} \in \mathbf{R}^{|O| \times D}$ are the representations of the observed regions, $\boldsymbol{y}_{V+1}^{(O)} \in \mathbf{R}^{|O|}$ are the observed entries of the new vote, and $\boldsymbol{I}_D$ is a D-dimensional identity matrix. In general, we make the algorithm more efficient by reusing the optimal parameters $\boldsymbol{w}_*$ learned with $|O|$ observations when new observations arrive.

Although this algorithm is intuitive, considering the particular structure shown in Figure 1, its general performance is not obvious. In standard matrix factorization, defined in Equation (2), both $\boldsymbol{A}$ and $\boldsymbol{B}$ are learned together. Our algorithm fixes $\boldsymbol{A}$ to be equal to $\boldsymbol{X} = \boldsymbol{U}\Sigma$, at the expense of adding some constraints, but with the benefit of computational complexity and identifiability gains. In terms of identifiability, our regularized negative log-likelihood is strictly convex, which now guarantees a unique global optimum. To limit computational cost, we factorize the matrix $\boldsymbol{Y}_V$ only once and reuse its decomposition for each new observation(s) in $\boldsymbol{y}_{V+1}$. Computing one SVD has complexity $O(RD^2)$, as typically $D \leq R$. The optimization procedure (step 3) can be performed efficiently, *e.g.*, in $O(n(|O|D + D^3))$ for n iterations of Newton's method. With a univariate Gaussian likelihood, computing the least-squares solution has asymptotic complexity $O(|O|D^2 + D^3)$, which is dominated by the $|O|D^2$ term, as typically $D < |O|$. Finally, predicting unobserved values is only a (function of a) matrix-vector multiplication of complexity $O(|\mathcal{U}|D)$.

Elections are more complex than referenda because they have categorical outcomes. Let K be the number of possible outcomes (for example K political parties). The vote result matrix becomes a tensor $\boldsymbol{Y}_V \in \mathbf{R}^{R \times V \times K}$. To apply our algorithm, we concatenate the results of each party to collapse the last dimension. This yields a matrix $\boldsymbol{Y}_V \in \mathbf{R}^{R \times VK}$ that can be decomposed using SVD to obtain representations of regions (steps 1 and 2). For an election, the regional results are stored in a matrix $\boldsymbol{y}_{V+1} \in \mathbf{R}^{R \times K}$, and we use multivariate Gaussian or categorical likelihoods in the GLM to model the multiple outcomes (steps 3 and 4).

2.4 Probabilistic Interpretation

Voting data have the special property that the sum of all possible outcomes in a given region is equal to 1. The outcome $p \in [0,1]$ of a referendum is the probability p that it is accepted (and the probability $1-p$ that it is rejected). The suffrage $\boldsymbol{p} \in [0,1]^K$ obtained by K political parties in an election describes the probability mass function $p(k)$ that the k-th party is elected. As a result, we provide a probabilistic interpretation of outcomes of referenda and elections.

Let $P_{rv}^{(i)} \sim \text{Bernoulli}(p_{rv})$ be a random variable representing the vote cast by voter i in region r on referendum v. As voting is anonymous, we do not observe individual votes, rather the average vote in each region

$$\frac{1}{N_r} \sum_{i=1}^{N_r} P_{rv}^{(i)},$$

where N_r is the number of voters in region r, and whose expectation is p_{rv}. By decomposing the result matrix $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{B}^\intercal$ as in Equation (2), we posit that the parameter of the random variables describing individual voters is a product of latent features of regions and votes $p_{rv} = \boldsymbol{a}_r^\intercal \boldsymbol{b}_v$, with $\boldsymbol{a}_r, \boldsymbol{b}_v \in \mathbf{R}^D$. In Equation (3) and Equation (4), our algorithm learns the latent features of the regions $\boldsymbol{a}_r = (\boldsymbol{U}\Sigma)_r = \boldsymbol{x}_r$ from historical data. In Equation (5), it learns the latent features of the votes $\boldsymbol{b}_v = \arg\min_{\boldsymbol{b}} -\ell_p(\boldsymbol{b}; \boldsymbol{X}, \boldsymbol{y}_v)$ as the parameters of a GLM p.

So far, we have considered that each region has the same number of voters. If we have access to data about the number of voters in

Table 2: Description of datasets used in our experiments.

Country	Type	Region	R	V	K	Period
Switzerland	Binary	Munic.	2 196	330	–	1981–2020
U.S.	Binary	State	50	11	–	1976–2016
Germany	Categ.	State	16	6	5	1990–2009
Germany	Categ.	District	538	5	5	1990–2005

each region (*e.g.*, the number of valid votes, the number of eligible voters, or the population), we can include this information by replacing the regularized log-likelihood in (5) by

$$-\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1}) = \sum_{r \in O} N_r \log p(y_{r,V+1} | \boldsymbol{w}, \boldsymbol{x}_r) + \lambda \|\boldsymbol{w}\|_2^2, \tag{8}$$

where $N_r \in \mathbf{R}$ is a count related to the number of voters in region r. We refer to the variation of the algorithm that uses this log-likelihood as *weighted*. We refer to the variation of the algorithm that uses the log-likelihood in (5) as *unweighted*. A similar argument can be trivially made for elections by letting $P_{rv}^{(i)} \sim \text{Categorical}(p_{rv}, K)$ be a random variable describing the vote cast by voter i in region r on vote v for K political parties.

2.5 Limitations

By design, our approach suffers from the cold-start problem of collaborative filtering [19]. We can make predictions only when at least one past observation is available, *i.e.*, when $|O| = 1$. To bypass this problem, Etter et al. [15] include features of the regions, such as the geographical location, the population size, and the elevation, and features of the votes, such as the voting recommendation by political parties. These features are, however, not systematically and programmatically available, making it difficult to use them in a real-world system such as the one we describe in Section 4.

Our approach also makes the hypothesis that regional and vote representations are static over time. In particular, the algorithm learns the regional representations over the whole training set. The latest results might, however, provide more information than older results. To bypass this problem, we could weigh the SVD by using a sliding window or by exploiting a temporal SVD algorithm [1] to capture the dynamics of the voting process.

3 EXPERIMENTS

We evaluate our algorithm on the four datasets[2] described in Table 2. The outcomes for the Swiss referenda and for U.S. presidential elections are binary. For Switzerland, this corresponds to the referendum being accepted or rejected. For the U.S. this corresponds to one presidential candidate being elected over the other. The outcomes for the German legislative elections are one of five categories, corresponding to five political parties.

For the binary datasets, *i.e.*, for Switzerland and for the U.S., we use a GLM p with univariate Gaussian and Bernoulli likelihoods. As data about the number of valid votes and about population counts are available for these two datasets, we use a likelihood with weighting, as defined in (8). For the categorical datasets, *i.e.*, for Germany, we use a GLM with multivariate Gaussian and categorical likelihoods. As data about population counts were not available in this case, we use a likelihood without weighting, as defined in in (5).

3.1 Evaluation

For each dataset, we find the best hyperparameters using the training set only, as explained in details in Appendix A. To evaluate the performance of our algorithm, we compute the mean absolute error (MAE) and the accuracy on the national results.

We first describe the error metrics used for the binary outcome case then extend them for multiple outcomes, *e.g.*, when different parties can be voted. Let $\boldsymbol{y}^* \in \mathbf{R}^R$ be the true regional results and let $\boldsymbol{y} := \boldsymbol{y}_{V+1} \in \mathbf{R}^R$ be a prediction. The true national outcome $y^* \in \mathbf{R}$ is defined as

$$y^* := \frac{1}{N} \sum_{r \in [R]} N_r y_r^*, \tag{9}$$

where $N = \sum_{r \in [R]} N_r$ is the total number of voters. The predicted national outcome $y \in \mathbf{R}$ is defined as

$$y := \frac{1}{|\mathcal{U}|} \sum_{r \in \mathcal{U}} N_r y_r + \frac{1}{|O|} \sum_{r \in O} N_r y_r^*, \tag{10}$$

where the prediction y_r in some observed region $r \in O$ equals the true outcome y_r^*. Then, the MAE and the accuracy of the national prediction are computed as

$$\text{MAE}(y, y^*) = |y - y^*|, \tag{11}$$

$$\text{Acc}(y, y^*) = \mathbf{1}\{y \geq 0.5 \text{ and } y^* \geq 0.5\} + \mathbf{1}\{y < 0.5 \text{ and } y^* < 0.5\}, \tag{12}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function.

The MAE enables us to evaluate how far a predictor is from the exact percentage value, whereas the accuracy enables us to evaluate if the outcome is predicted correctly. For K outcomes, the true and the predicted outcomes are vectors $\boldsymbol{y}^* \in [0,1]^K$ and $\boldsymbol{y} \in [0,1]^K$, respectively, and the MAE in (11) is simply the ℓ_1-norm of the difference between the two vectors. As the accuracy is not defined for multiple outcomes, we compute the average displacement (or Spearman's footrule) [11]. Let $p : [K] \to [K]$ be a permutation map from a party to its rank for the predicted order, and let $p^* : [K] \to [K]$ be a permutation map for the true order. The average displacement is then computed as

$$D(p, p^*) = \frac{1}{K} \sum_{k=1}^{K} |p(k) - p^*(k)|. \tag{13}$$

This measures the average position shift between the true rank and the predicted rank of each party.

We train our algorithm on data up to vote V and make predictions on vote $V+1$ to evaluate our algorithm. To simulate a real setting where results arrive sequentially, we incrementally add regions to the set of observed regions O and average the MAEs on several reveal orders to obtain error bars. Current political forecasting methods for real-time estimation of the outcomes (*e.g.*, by media outlets) rely mostly on weighted averages of the regional results on the day of the vote. More sophisticated methods (developed, *e.g.*, by polling agencies) can also be used, but their technical details are not available. Hence, we compare our algorithm against

[2] The data and the code are available on github.com/indy-lab/submatrix-factorization.

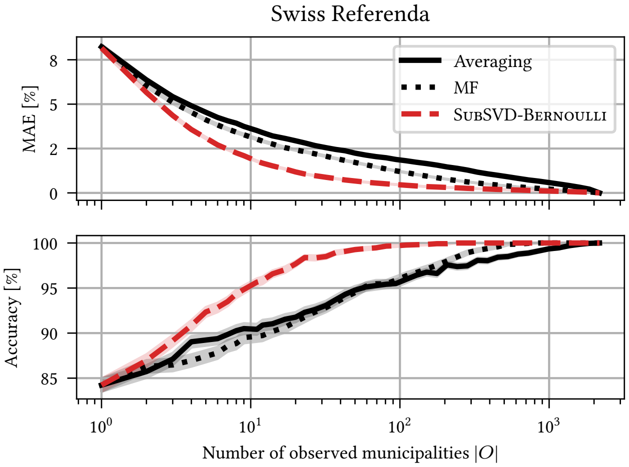

Figure 2: MAE (top) and accuracy (bottom) averaged over 26 Swiss referenda and 100 reveal orders each.

weighted averaging as a baseline. For the binary classification task, we also compare against standard matrix factorization (MF) trained with alternating least squares, as proposed by Etter et al. [15] and as formulated in Equation (2). For the multiple outcome task, we restrict our comparison to weighted averaging.

3.2 Swiss Referenda

We collect a dataset of $V = 330$ referenda in $R = 2196$ municipalities (the regions are here the municipalities) between 1981 and 2020. We start with a training set of $V = 300$ votes and report the average performance on the next 26 votes with 100 reveal orders each. As several votes can occur on the same day, we make sure that only past votes are used in the training set. In Section 4, we analyze in depth the last four votes (two votes on two dates) for which we have real, sequential data. The ranges of hyperparameters are given in Appendix A.1, and the best combination for the Bernoulli likelihood is $\lambda = 0.01$ and $D = 25$.

In Figure 2, we show the MAE and the accuracy of our algorithm to predict national results from partial municipal results. The two likelihoods used for the GLM provide equal performance, and we report only the performance of the Bernoulli likelihood for clarity. In terms of MAE (top), MF outperforms the weighted average baseline and our algorithm outperforms MF for every number of observed regions from 1 to 1000. The difference becomes marginal when more than 1000 results are observed, which suggests that a good approximation of the national result can be obtained by simply averaging the observed results when more than 50% of the results have arrived. Nevertheless, in this synthetic setting (the reveal order is randomized) our approach gains only one percentage point at best over the baseline. In Section 4, we will show that the gain becomes substantial with real data, *i.e.*, with the actual reveal order. In terms of accuracy (bottom), our algorithm predicts the final outcome with 95% accuracy with 10 observed regions only, outperforming the two baselines by 5 percentage points. The accuracy of our algorithm reaches 100% after observing 200 municipal results, *i.e.*, after observing 10% of all municipalities.

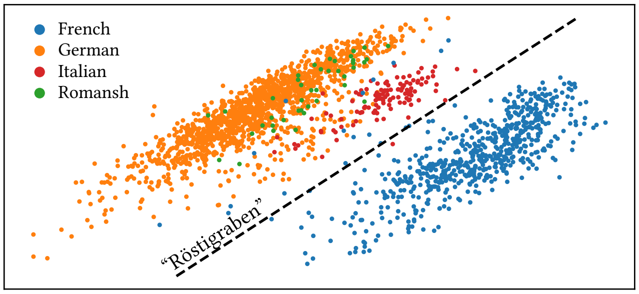

Figure 3: Projection of Swiss municipalities on the first two singular vectors of referendum matrix Y. Municipalities are colored according to their language.

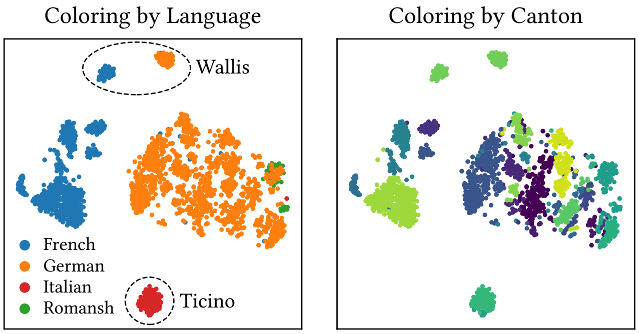

Figure 4: Projection of Swiss municipalities from referendum matrix Y with t-SNE. (Left) Municipalities are colored according to their language. (Right) Municipalities are colored according to their canton (26 cantons). The bilingual canton of Wallis is split into two clusters. The only Italian-speaking canton of Ticino is isolated from the other clusters.

We explore the patterns in the feature matrix $\boldsymbol{X} = \boldsymbol{U\Sigma}$ obtained from (4). In Figure 3, we plot the first two columns of $\boldsymbol{X}$, *i.e.*, a projection of the municipalities on the first two singular vectors of the vote representation. This plot, popularized by Etter et al. [14], shows two clear clusters of municipalities corresponding to their language. It also exhibits the infamous *Röstigraben*, a cultural separation between French-speaking municipalities and German-speaking municipalities. In addition, we show in Figure 4 a projection of the result matrix $\boldsymbol{Y}$ by using t-SNE [22]. The language separation is also clearly visible, with French-speaking municipalities on the left of the plot and German-speaking municipalities on the right. The group of municipalities are further subdivided into smaller clusters corresponding to the canton (states of the Swiss confederation) that they belong to. Most cantons are uni-lingual in Switzerland, but a few are bilingual. The most notable among them is Wallis, and interestingly enough, we observe that it is separated into two distinct clusters. The French-speaking municipalities in Wallis are closer to other French-speaking municipalities, and vice versa for the German-speaking municipalities. The municipalities of the only Italian-speaking canton, Ticino, form their own cluster.

3.3 U.S. Presidential Election

The U.S. presidential election takes place every four years. We obtain a dataset about the state-level ballots between 1976 and 2016 [23]. In the spirit of Nate Silver's FiveThirtyEight [28], we evaluate the performance of our algorithm at predicting the result of the U.S. presidential election in 2016. The U.S. presidential election relies on the electoral-college system, which adds one level of complexity to the prediction because (1) the state-level results are quantized to an integer number of delegates and (2) the candidate who wins the majority of votes in a state wins all the delegates of that state. This (non-linear) winner-take-all rule requires further modeling assumptions and is out of the scope of this paper. Instead, we focus on predicting the results of the popular vote.

We transform the outcome of the election into a binary outcome of Democratic candidate and Republican candidate. In all these elections, the results of other parties, *e.g.*, the Green party and independent candidates, are insignificant compared to the two major U.S. parties. This dataset contains the results of $V = 11$ votes in $R = 51$ regions (50 states and the District of Columbia) between 1976 and 2016. As the number of votes is small, we train our algorithm on all votes up to 2012 ($V = 10$) to set the sub-matrix $\boldsymbol{Y}_V$, and we predict the state-level results and the national results of the 2016 election. We report the averaged performance on 10000 random reveal orders. The ranges of hyperparameters are given in Appendix A.2, and the best combination for the Bernoulli likelihood is $\lambda = 0.01$ and $D = 7$.

In Figure 5, we show the MAE and the accuracy of our algorithm in predicting this election. The two likelihoods used for the GLM provide equal performance, and we report only the performance of the Bernoulli likelihood for clarity. In terms of MAE (top), our algorithm and MF outperform the weighted average baseline after observing the results in two regions. In terms of accuracy (bottom), our algorithm outperforms both MF and the weighted average for any number of observation. All models have an accuracy of 41% after observing the result of one region. This is because the Democratic candidate won in 21 of 51 regions (41%) and won the popular vote.

3.4 German Legislative Election

German legislative elections take place every four years. We obtain two datasets [25] of regional results with $R = 16$ states (1990–2009) and $R = 538$ districts (1990–2005). After 2005 (for the districts) and 2009 (for the states), the data are regrettably not publicly available any longer. We keep $K = 5$ political parties, corresponding to the five major parties in Germany[3] for which we have data over the whole period. The datasets cover $V = 6$ votes for state-level results and $V = 5$ for district-level results. As there are multiple outcomes, we use a categorical likelihood to predict the results of the five parties.

For the state-level results, we train our algorithm on all votes up to 2005 ($V = 5$) to set the sub-matrix $\boldsymbol{Y}_V$, and we predict the national results of the 2009 election. For the district-level results, we train our algorithm on all votes up to 2001 ($V = 4$), and we predict the national results of the 2005 election. In Figure 6, we show the performance of our algorithm in predicting these two elections. For both datasets, our algorithm outperforms the baseline already after a small number of observations. The performance for the prediction of the national results when using the fine-grained district-level results is better than when using coarser-grained state-level results. Remarkably, after observing the results in 10 districts (Figure 6, top right), *i.e.*, approximately the average number of districts per state, the MAE reaches 1%, which is four times better than the MAE obtained after predicting the national outcome from one state (Figure 6, top left). A similar observation can be made for the average displacement. This suggests that the finer the level of granularity of regions is, the better the predictive performance is, even if the observed results are obtained from the same number of voters.

Like with Switzerland in Section 3.2, we explore the representations of the regions contained in the feature matrix $\boldsymbol{X}$ for Germany.

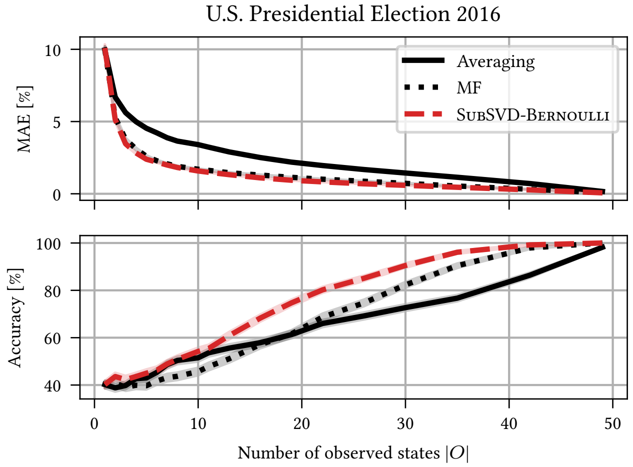

Figure 5: MAE (top) and accuracy (bottom) of the popular vote of the U.S. presidential election in 2016.

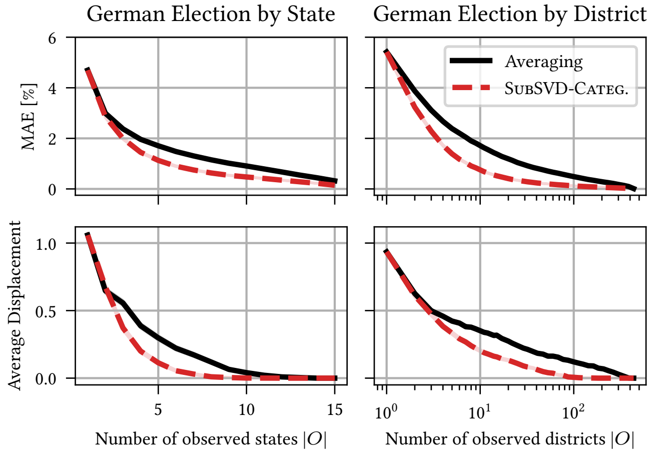

Figure 6: MAE (top) and average displacement (bottom) of German legislative elections at state level in 2009 (left) and district level in 2005 (right).

[3] CDU/CSU (christian democracy), SPD (social liberalism), FDP (conservative liberalism), the Green party (ecological), and the Left party (radical left).

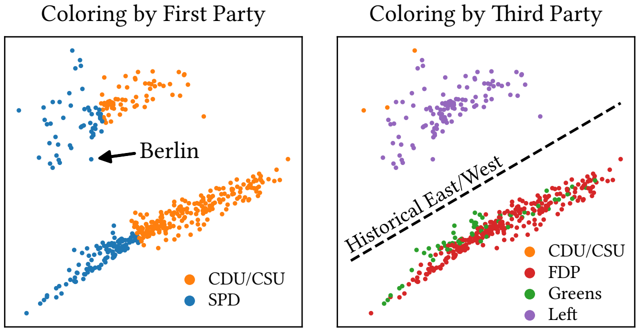

Figure 7: Projection of German district on the first two singular vectors of election matrix Y. (Left) Districts are colored according to the first party elected in each of them. (Right) Districts are colored according to the third party elected. This coloring reveals the historical East/West separation.

In Figure 7, we plot the first two columns of $\boldsymbol{X}$, *i.e.*, a projection of the districts on the first two singular vectors of the vote representations. We color the points according to the first party elected in the corresponding districts (left). With no exception, either the CDU/CSU or the SPD is elected. The two clusters are each separated in half: The districts on the right side of their cluster vote in majority for the CDU/CSU. For the lower cluster, those districts also belong to Southern Germany. The districts on the left side of this cluster (which vote in majority for the SPD) belong to North-Western Germany.

The CDU/CSU and the SPD have the top two ranks in all districts. Therefore, it is interesting to color the points according to the party in third place. This clearly separates the two clusters. The cluster at the top corresponds to the Left party.[4] The top cluster contains only districts that belong to historical East Germany (formerly the GDR, before the reunification in 1990), such as Potsdam, Leipzig, and Dresden. The cluster at the bottom corresponds to the Green party and the FDP and contains only districts that belong to historical West Germany (the former BDR), such as Frankfurt, Munich, and Hamburg. Interestingly, Berlin lies in the cluster that corresponds to historical East Germany, but seems slightly isolated.

4 DEPLOYED SYSTEM

We deploy a Web platform to provide real-time predictions for Swiss referenda[5]. Four Sundays a year, Swiss citizens are called on to vote on at least one item in a referendum. These items can cover a broad range of topics, from joining the European Union to subsidizing railways and roads, from banning the use of fossil fuels to cutting taxes, and even forbidding Swiss farmers to remove horns from cows and goats. A month prior to a referendum vote day, eligible voters receive official ballots, together with useful documentation. To cast their vote, they can either send their ballot by post or bring it to the ballot office on the referendum vote day, up to 11:59am. Starting at 12pm, each municipality is in charge of counting both the remote ballots and the ballots they collected on the same day. Once they have finished counting, they report the result to their canton whose administration communicates the official count.

Table 3: True outcome y^*_{nat}, earliest prediction y_{nat}, and absolute difference $\Delta = |y^*_{\text{nat}} - y_{\text{nat}}|$ for referenda with real data.

Date	Item	y^*_{nat} [%]	y_{nat} [%]	Δ
May 19	Tax Reform	66.38	67.90	1.52
May 19	Weapon Regulation	63.73	63.52	0.21
Feb 9	Affordable Houses	42.95	41.57	1.38
Feb 9	Ban on Homophobia	63.09	62.94	0.15

4.1 Implementation Details

In 2019, the Swiss Federal Statistical Office released a public API to access vote data, both historical and real-time, for all municipalities in a standardized format [31]. This enabled us to obtain sequential results in all municipalities on the referendum vote days and made it possible to use our algorithm to predict the outcome of referenda starting at 12pm. We use the dataset described in Table 2 for Switzerland, which contains $R = 2196$ municipalities. We predict the outcome of two items on May 19, 2019, using $V = 326$ votes and two items on February 9, 2020, using $V = 328$ votes[6]. We summarize these four items in Table 3. The turnout was about 44% on May 19 and about 42% on February 9020. For each referendum, about 2.2 million valid ballots were counted.

For a vote $V + 1$, we use the historical data up to vote V to learn the feature matrix $\boldsymbol{X}$ from the sub-matrix $\boldsymbol{Y}_V$. We use a Bernoulli likelihood to define our GLM with $D = 25$ latent dimensions and a regularization factor $\lambda = 0.01$. We fetch municipal results from the API every two minutes[7]. If new results are available, we learn the optimal parameters $\boldsymbol{w}_*$ by optimizing the negative log-likelihood using Newton's method, and we predict the unobserved municipal results as $\boldsymbol{y}^{(\mathcal{U})}_{V+1} = \sigma(X^{(\mathcal{U})}\boldsymbol{w}_*)$. We predict the national outcome $y_{\text{nat}} \in [0, 1]$ by aggregating our prediction of unobserved results $\mathcal{U}$ with the observed results $\mathcal{O}$ as

$$y_{\text{nat}} = \frac{1}{N}\left(\sum_{r\in\mathcal{U}} N_r^{(\mathcal{U})} y_{r,V+1}^{(\mathcal{U})} + \sum_{r\in\mathcal{O}} N_r^{(\mathcal{O})} y_{r,V+1}^{(\mathcal{O})}\right), \qquad (14)$$

where $N_r^{(\mathcal{U})}$ is the number of valid ballots in municipality r from the previous vote (used as proxy for the current vote), $N_r^{(\mathcal{O})}$ is the number of valid ballots in municipality r for the current vote, and $N = \sum_{r\in\mathcal{U}} N_r^{(\mathcal{U})} + \sum_{r\in\mathcal{O}} N_r^{(\mathcal{O})}$ is the total number of valid ballots. As the number of unobserved results $|\mathcal{U}|$ tends to 0 with time and the number of observed results $|\mathcal{O}|$ tends to the total number of regions R, the prediction for the national outcome y_{nat} converges to the true outcome $y^*_{\text{nat}} \in [0, 1]$.

4.2 Real-Time Predictions

In Figure 8, we show the evolution of our predictions (solid red line), together with the weighted averaging (solid black line), and the

[4] The three exceptions with CDU/CSU voted the Left party in second place.

[5] The platform is available on www.predikon.ch.

[6] The two referendum vote days between May 19, 2019, and February 9, 2020, were replaced by the Swiss legislative elections in Fall 2019. The referendum vote days after February 9, 2020, were cancelled due to the COVID-19 crisis in Spring 2020.

[7] Schedule suggested by the Swiss Federal Statistical Office.

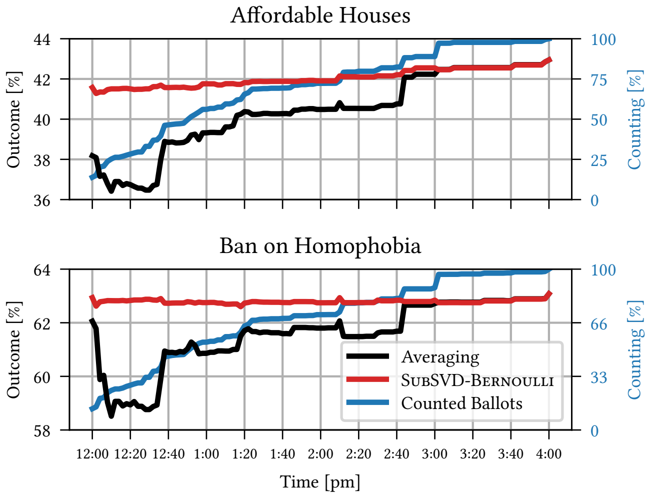

Figure 8: Evolution of predictions (red) and weighted averaging (black) on real, sequential data for the two referenda of February 9, 2020, together with the progress of the ballot counting (blue).

progress of the ballot counting (solid blue line) for the two referenda on February 9, 2020. The ballot counting starts at 12pm and ends at 4pm, after all municipalities reported their results Looking at the trajectory of the counting progress, the jumps occurring at several timestamps correspond to the publication of the results of the whole canton of Wallis and the city of Basel at 12:35pm, of Thun at 2:10pm, of Geneva at 2:40pm, and of Bern at 3pm, all of which are major cities in Switzerland. The large municipality of Zurich is split into nine districts that published their results independently, thus diluting its effect on the counting.

At 12pm, using the results of 531 municipalities (23.9% of all municipalities) representing 13.2% of the total population, we predict 41.57% for the "Affordable Housing" and 62.94% for the "Ban on Homophobia". This corresponds to a mean absolute error of 1.38% and 0.15% to the true outcome, respectively. The weighted average for the current count varies up to a difference of 6.5% and 4.5%, respectively, whereas our prediction is stable over time. To provide a robust estimation of the final outcome, our algorithm takes advantage of the correlation across municipalities and votes. The performance of our approach for the two referenda of May 19, 2019, is similar; but we do not show them[8], as early counting data were not available due to a bug on the API side (they published the first results at 12:35pm). Furthermore, the results of the nine districts of Zurich, which cumulatively form the largest municipality in Switzerland, were incorrectly reported on that day. Consequently, we could not reliably use the results in our algorithm. We report only the earliest prediction, made at 12:40pm, in Table 3.

5 RELATED WORK

We base the present paper on the work of Etter et al. [14] and build on their approach proposed in [15]. They combine matrix factorization and Gaussian processes (GP) to understand what features of the votes and of the municipalities contribute the most to the predictive performance. They develop an expectation-maximization algorithm to learn both latent features and the GP parameters jointly. They show that the geographical location of municipalities is the most important feature for making predictions, an aspect that is in part captured by the feature matrix $\boldsymbol{X}$ of Equation (4) in our algorithm and illustrated in Figure 3: Municipalities that are geographically close tend to speak the same language. They also show that they are able to make accurate predictions of Swiss referenda. In comparison, our method is more efficient, as it learns the latent features of municipalities $\boldsymbol{X}$ through singular value decomposition offline, and it learns the latent features of a vote through a GLM. The GLM also provides more flexibility: Our algorithm could conceivably be used to make prediction for other types of observations, *e.g.*, count data, and works for non-binary outcomes. We developed our algorithm with applicability in mind. Our main goal was to make real-time predictions for Swiss referenda, with all the constraints that come with this problem.

The problem we address, *i.e.*, predicting unobserved entries of a new column of a matrix from partial observations of that column, is most similar to the problem of missing-data imputation. The use of SVD for data imputation has been studied in the context of genomics [17, 32]. In gene matrices, missing entries are common, and the authors propose an algorithm based on SVD to impute missing data. Their algorithm iteratively computes the SVD of an approximation to the full matrix and predicts the missing values with a regression by using the non-missing values to refine the approximation. An extensive literature review of predictive methods for data imputation is available in Bertsimas et al. [5]. Incremental SVD revisions have been studied in the context of computer vision [8] and recommender systems [9]. In this latter work, the author proposes algorithms to compute the SVD of a matrix when new columns arrive sequentially and are corrupted by some noise (*e.g.*, some entries are missing). Their solution is equivalent to our SubSVD-Gaussian algorithm without regularization, *i.e.*, $\lambda = 0$, for which a closed form solution is provided in Equation (7).

A whole body of work in the political science community exists on election forecasting [21], *i.e.*, predicting the outcome of an election before it happens. The seminal work of Bean [2], who first studied this problem in 1948, looked at using historical data to find U.S. states that were the most predictive of the national outcome. Statistical models for election forecasting have since been developed in many contexts for Germany [33], France [3], the U.K. [16], and the U.S. [18, 27]. The prediction of U.S. elections has been popularized by the blogger and statistician Nate Silver in 2008 as he predicted Barack Obama's victory in the Democratic Party primaries using a statistical model of historical data [6], and as he predicted Barack Obama's victory in the presidential election from polling data [28]. In the computer science community, algorithms for election forecasting have also been developed using social media data in Denmark [20], Finland [34], the U.S. [10, 26], and the developing world [12]. To the best of our knowledge, except for the work mentioned at the beginning of this section, we are the first to study real-time outcome predictions of elections and referenda, and to deploy a system for making predictions of Swiss referenda in real-time.

[8] The interested reader can access these predictions on our Web platform.

6 CONCLUSION

We have proposed an algorithm to predict national vote results from regional results that are observed sequentially. Our approach learns a representation for each region by factorizing the sub-matrix of historical data and approximating the representation of a new vote as the optimal parameters of a generalized linear model. The predictions for unobserved results are obtained through the link function of the GLM, and national predictions are obtained by aggregating observed and unobserved regional results. We are able to predict both referenda with binary outcomes and elections with categorical outcomes. We have shown that our approach outperforms the (weighted) average of partial results on three datasets of Swiss referenda, U.S. presidential elections, and German legislative elections. We have explored the regional representations in their latent space and have shown that they capture ideological and cultural patterns. Finally, we have deployed a Web platform to provide real-time vote predictions for Swiss referenda. Our algorithm is able to predict the final outcome of four real votes with an absolute error of about 1% after observing only 13% of the ballots.

Future Work. We plan to further develop our approach in three directions. First, Bayesian inference in our generalized linear model would enable uncertainty quantification of our predictions in a principled way. This could be beneficial for predictions, especially during the early counting phase. Bayesian inference for GLMs has been widely studied in the literature [24]. Second, our algorithm is capable of making predictions only with at least one observed regional result. In the spirit of Etter et al. [15], we plan to augment our algorithm with features from the vote and the municipalities to make predictions prior to referenda in Switzerland. One limitation of their work lies in the lack of systematic availability of the features they include in their model. In particular, every Swiss citizen receives documentation about each referendum. These explanatory documents provide a valuable source of information about a vote, one that could be incorporated in a predictive model. The actual text of the proposed laws would provide another source of relevant information. Finally, by collecting the sequential order by which regional results arrive in Swiss referenda, we obtain data about the true reveal order. We plan to explore whether the true sequential order can be exploited to learn the schedule by which results arrive and, therefore, further improve the earliest predictions.

ACKNOWLEDGMENTS

We thank Holly Cogliati-Bauereis, Vincent Etter, Julien Herzen, and the anonymous reviewers for careful proofreading and constructive feedback. We thank Corinne Straub of the Swiss Federal Statistical Office for setting up the API and helping us understand their data.

REFERENCES

[1] R. Bamler and S. Mandt. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 380–389, 2017.

[2] L. H. Bean. How to predict elections. 1948.

[3] E. Belanger. Finding and using empirical data for vote and popularity functions in France. *French Politics*, 2(2):235–244, 2004.

[4] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 43–52. IEEE, 2007.

[5] D. Bertsimas, C. Pawlowski, and Y. D. Zhuo. From predictive methods to missing data imputation: an optimization approach. *The Journal of Machine Learning Research*, 18(1):7133–7171, 2017.

[6] M. Blumenthal. The poblano model, 2008. URL https://web.archive.org/web/20090414152429/http://www.nationaljournal.com/njonline/mp_20080507_8254.php. Accessed: 2020-02-13.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[8] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, pages 707–720. Springer, 2002.

[9] M. Brand. Fast online svd revisions for lightweight recommender systems. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 37–46. SIAM, 2003.

[10] M. Choy, M. Cheong, M. N. Laik, and K. P. Shung. US presidential election 2012 prediction using census corrected Twitter model. *arXiv preprint arXiv:1211.0938*, 2012.

[11] P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2):262–268, 1977.

[12] N. Dwi Prasetyo and C. Hauff. Twitter-based election prediction in the developing world. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, pages 149–158, 2015.

[13] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 1936.

[14] V. Etter, J. Herzen, M. Grossglauser, and P. Thiran. Mining democracy. In *Proceedings of the second ACM Conference on Online Social Networks*, 2014.

[15] V. Etter, M. E. Khan, M. Grossglauser, and P. Thiran. Online collaborative prediction of regional vote results. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016.

[16] F. Franch. (wisdom of the crowds) 2: 2010 uk election prediction with social media. *Journal of Information Technology & Politics*, 10(1):57–71, 2013.

[17] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein. Imputing missing data for gene expression arrays. 1999.

[18] R. Kennedy, S. Wojcik, and D. Lazer. Improving election prediction internationally. *Science*, 355(6324):515–520, 2017.

[19] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[20] J. B. Kristensen, T. Albrechtsen, E. Dahl-Nielsen, M. Jensen, M. Skovrind, and T. Bornakke. Parsimonious data: How a single Facebook like predicts voting behavior in multiparty systems. *PloS one*, 12(9), 2017.

[21] M. S. Lewis-Beck. Election forecasting: Principles and practice. *The British Journal of Politics and International Relations*, 7(2):145–164, 2005.

[22] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[23] MIT Election Data and Science Lab. U.S. President 1976–2016, 2017. URL https://doi.org/10.7910/DVN/42MVDX. Accessed: 2020-02-06.

[24] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[25] Norwegian Centre for Research Data. German parliamentary elections, 2020. URL https://nsd.no/european_election_database/country/germany/parliamentary_elections.html. Accessed: 2020-06-16.

[26] J. Ramteke, S. Shah, D. Godhia, and A. Shaikh. Election result prediction using Twitter sentiment analysis. In *2016 international conference on inventive computation technologies (ICICT)*, volume 1, pages 1–5. IEEE, 2016.

[27] S. E. Rigdon, S. H. Jacobson, W. K. Tam Cho, E. C. Sewell, and C. J. Rigdon. A Bayesian prediction model for the US presidential election. *American Politics Research*, 37(4):700–724, 2009.

[28] N. Silver. Pollster ratings v3.0, 2008. URL https://fivethirtyeight.com/features/pollster-ratings-v30/. Accessed: 2020-02-13.

[29] The Swiss Confederation. Democracy, 2019. URL https://www.ch.ch/en/demokratie/. Accessed: 2020-02-04.

[30] The Swiss Confederation. Popular vote, 2019. URL https://www.admin.ch/gov/en/start/documentation/votes.html. Accessed: 2020-02-04.

[31] The Swiss Federal Statistical Office (via opendata.swiss). Real-time data on referenda on vote days, 2020. URL https://opendata.swiss/en/dataset/echtzeitdaten-am-abstimmungstag-zu-eidgenoessischen-abstimmungsvorlagen. Accessed: 2020-02-11.

[32] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[33] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election forecasts with Twitter: How 140 characters reflect the political landscape. *Social science computer review*, 29(4):402–418, 2011.

[34] T. Vepsäläinen, H. Li, and R. Suomi. Facebook likes and public opinion: Predicting the 2015 Finnish parliamentary elections. *Government Information Quarterly*, 34 (3):524–532, 2017.

Table 4: Ranges of hyperparameters for different datasets.

Country	Region	λ	D
Switzerland	Munic.	$\{0.001, 0.01, 0.1\}$	$\{10, 25, 100, 250\}$
U.S.	State	$\{0.001, 0.01, 0.1\}$	$\{3, 5, 7\}$
Germany	State	$\{0.001, 0.01, 0.1\}$	$\{3, 7, 11\}$
Germany	District	$\{0.001, 0.01, 0.1\}$	$\{3, 7, 11\}$

A EXPERIMENTAL SETTING

We describe in details the experimental setting and the choice of hyperparameters. As explained in Section 3, we evaluate the predictive performance of our models and of the baselines after observing a fraction of regional outcomes. For each dataset, we use the training set as validation set to find the best hyperparameters from a range of possible values. We preserve the temporal order of the data, and we use the first V votes for training while testing on the next vote $V + 1$. This simulates a real setting where we use all available data prior to an election or referendum of interest.

For each test vote, we simulate several random reveal orders of regional results. For the Swiss referenda, the U.S. presidential election, and the German parliamentary election by district, we simulate reveal orders on a logarithmic space. This emphasizes the importance of early results, *i.e.*, when a small number of results are available, in the prediction performance. For the German parliamentary election by states, the number of regions is small enough ($R = 16$) to simulate reveal orders on a linear space.

We evaluate the performance of our models and of the baselines with different combinations of the hyperparameters. The hyperparameters in our method are the ℓ_2-regularization parameter λ and the number of dimensions D of the latent factors. We report in Table 4 the ranges of hyperparameters for each dataset and in Table 5 the best combinations in terms of MAE. In our experiments, we generally observed that the performance of an algorithm was robust to different combinations of hyperparameters. In light of Occam's razor, we chose the simplest model, *i.e.*, the lowest number of latent dimensions D and the lowest value of regularization parameter λ, when different combinations reached equal values of MAE.

Finally, we measure the performance of an algorithm given some hyperparameters in terms of the MAE and the ℓ_1-norm, as defined in Equation (11) of Section 3. We use the MAE (and the ℓ_1-norm) as it measures the error in percentage points and provides, therefore, some interpretability. For the binary datasets, *i.e.*, for the Swiss referenda and for the U.S. presidential election, we measure the performance of a combination in terms of the MAE. For the categorical datasets, *i.e.*, for the German elections where we try to predict the fractions of votes that parties obtain, we measure the performance in terms of the ℓ_1-norm to sum up the prediction errors across different parties.

A.1 Swiss Referenda

In Switzerland, referenda occur when 50 000 people petition against a law that has been accepted by the Parliament (optional referendum) or when the Swiss Constitution is modified (mandatory referendum). Popular initiatives occur when 100 000 people suggest a new law. For simplicity, we refer to optional referenda, mandatory referenda, and popular initiatives as *referenda.*

Table 5: Best hyperparameters for each model and dataset.

Country	Region	Likelihood	λ	D
Switzerland	Munic.	Gaussian	0.1	25
		Bernoulli	0.01	25
U.S.	State	Gaussian	0.01	5
		Bernoulli	0.01	7
Germany	State	Gaussian	0.01	7
		Bernoulli	0.01	7
Germany	District	Gaussian	0.01	11
		Bernoulli	0.01	11

We collected the data about Swiss referenda between 1981 and 2020 from the Swiss Federal Statistical Office. The data are published through an API on the Swiss Open Data platform[9] We pre-process the data as follows: First, we remove 12 regions (*i.e.*, the municipalities) with missing values. This may happen because, each year, some municipalities are merged or split, and some results might not exist for some votes. Second, we merge the regions that change their name, and we average their results.

In total, there are $V = 326$ referenda and $R = 2186$ municipalities in the data set used for the evaluation. The validation set consists of referenda 275 to 300, and we use it to find the best hyperparameters. The test set consists of referenda 301 to 326, and we use it to report the results in Section 3. We test values for $\lambda \in \{0.001, 0.01, 0.1\}$ and for $D \in \{10, 25, 100, 250\}$. The best model with Gaussian likelihood uses $\lambda = 0.1$ and $D = 25$ while the best model with Bernoulli likelihood uses $\lambda = 0.01$ and $D = 25$. We tune the hyperparameters over 10 random reveal orders per referendum, and we evaluate the performance of our algorithm over 100 random reveal orders. For the matrix factorization baseline, we use the best hyperparameters as reported by Etter et al. [15], *i.e.*, $\lambda_U = 31.0$, $\lambda_V = 0.03$, and $D = 25$.

A.2 US Presidential Election

The U.S. presidential election relies on the Electoral College system. In this system, 538 delegates are assigned to each state proportionally to their population, and a candidate who obtains the majority of votes in a state wins all the delegates in that state[10]. The candidate who wins the majority of the delegates among all the states, *i.e.*, at least 270 delegates, is elected president. Because a candidate wins the same number of delegates whether it receives 99% of the votes or 51% of the votes, the collegial system leads to some unexpected behaviour: A candidate may win the popular vote but lose the collegial vote. This happened only in two elections in our dataset: in 2000 and in 2016. This special structure adds one level of complexity to the prediction task, and it requires further modeling assumptions. To keep our approach general and because a mismatch between the popular vote and the collegial vote is rare, we keep this specificity of the U.S. electoral system for future work.

[9] https://opendata.swiss/en/dataset/echtzeitdaten-am-abstimmungstag-zu-eidgenoessischen-abstimmungsvorlagen

[10] With the exception of Maine and Minnesota, which have a different rules.

We use the U.S. presidential election data between 1976 and 2016. The data is publicly available on Harvard Dataverse [23]. The data reports state-level election outcomes with the number of votes received by each candidate. We transform the outcome of the election into a binary outcome of Democrat candidate and Republican candidate. Candidates from other parties are ignored, and we normalize the results of the candidates from the two major parties so that the sum of their votes is 1. In comparison to the Swiss referenda, aggregating by number of voters will, therefore, be inexact for the U.S. presidential elections. This is nonetheless a reasonable approximation, as the number of votes received by candidates from other parties are marginal compared to the candidates from the Republican party and from the Democrat party.

In total, there are $V = 11$ elections. The data include the results of the District of Columbia (*i.e.*, Washington D.C.), which has a special status and is not considered a state; hence, we have $R = 51$ regions, combining 50 states and the District of Columbia. We find the best hyperparameters using the vote prior to the 2012 election, and we evaluate the model on the 2016 election. We test values for $\lambda \in \{0.001, 0.01, 0.1\}$ and for $D \in \{3, 5, 7\}$ as we only have 9 elections prior to 2012. For both the Gaussian and Bernoulli likelihoods, the best model uses $\lambda = 0.01$. The best model with Gaussian likelihood uses $D = 5$ and the best model with Bernoulli likelihood uses $D = 7$. We tune the hyperparameters over 100 random reveal orders per election, and we evaluate the performance of our algorithm over 10000 random reveal orders.

A.3 German Parliamentary Election

We use the German parliamentary elections data published by the European Elections Database (EED) [25]. In Germany, parliamentary elections take place every 4 years. The EED reports results between 1990 and 2009 on state level and between 1990 and 2005 on district level. Similarly to the U.S. presidential elections, we normalize the results per region by keeping the main five parties in Germany (CDU/CSU, SPD, FDP, the Greens, and the Left).

In total, there are $V = 6$ state-level elections and $V = 5$ district-level elections. For state-level elections, we find the best hyperparameters using the votes prior to the 2005 elections and we evaluate the model on the 2009 elections. For district-level elections, we find the best hyperparameters using the votes prior to the 2002 elections and we evaluate the model on the 205 elections. We test values for $\lambda \in \{0.001, 0.01, 0.1\}$ and for $D \in \{3, 7, 11\}$. For both datasets, both the Gaussian and the Bernoulli likelihoods provide the same results. For state-level elections, the best model uses $\lambda = 0.01$ and $D = 7$. For district-level elections, the best model uses $\lambda = 0.01$ and $D = 11$. In both cases, we tune the hyperparameters over 100 random reveal orders per election, and we evaluate the performance of our algorithm over 1000 random reveal orders.