EPFL

Thèse n° 7186

Discrete-Choice Mining of Social Processes

Présentée le 24 juin 2021

Faculté informatique et communications
Laboratoire de la Dynamique de l'Information et des Réseaux 2
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Victor KRISTOF

Acceptée sur proposition du jury

Prof. K. Aberer, président du jury
Prof. P. Thiran, Prof. M. Grossglauser, directeurs de thèse
Prof. S. Hale, rapporteur
Prof. R. Ghani, rapporteur
Prof. R. West, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

The purpose of computing is insight, not numbers.
— Richard W. Hamming

To my family
In loving memory
of Dado and Tousi

Acknowledgments

This thesis would not have been possible without the contributions, explicit or implicit, of many. First and foremost, I would like to thank my advisors, Patrick Thiran and Matthias Grossglauser, for their mentorship. Their creativity, rigour, curiosity, and generosity were a true inspiration. Thank you, Patrick and Matthias, for believing in me when I did not.

I had the honor of having Bob West, Scott Hale, Rayid Ghani, and Karl Aberer as members of my jury committee. Thank you all for your time and your excellent comments and suggestions. I really enjoyed discussing the five and half years of my work with you.

I have been fortunate to be surrounded by a fantastic staff at the lab. Holly Cogliati-Bauereis provided invaluable help by proof-reading and improving my manuscripts. I became a much better writer thanks to her feedback (and her crisp humour!). Patricia Hjelt and Angela Devenoge assisted me with the administrative tasks—always with utmost kindness. I really enjoyed our personal discussions and their genuine interest in other people's lives. Marc-André Lüthi and Yves Lopes helped me with our IT infrastructure, which made tuning hyperparameters of predictive models an (almost) pleasant task. I thank them all.

I am very grateful to two special people, whom I consider my academic role models. Their guidance was invaluable to my obtention of this degree. Vincent Etter, I thank you for being an influential variable in my decision to begin my Ph.D. research. Since then, you have always been available to discuss my ideas and me give me honest feedback. Lucas Maystre, I thank you for your kindness, patience, positivity, and knowledge. I learned so much from you about so many aspects of doing research.

I also want to express my gratitude to my academic collaborators, in particular to Batuhan Yardim, Alexander Immer, and Aswin Suresh for being the best co-authors I could ever hope for. Working with all of you was a pure joy and very inspiring. I thank Roy Gava and Steven Eichenberger for co-organizing the AI & Democracy conference, Marlene Kammerer and Paula Castro for our project about international climate negotiations, Corinne Straub for building the API that made Predikon a reality, Jérôme Payet for our co-supervision and joint efforts to create Climpact. I also thank all the students who were involved in my projects and whose contributions helped me make progress in my research.

Acknowledgments

I am very grateful to Augustin Chaintreau and Ana-Andreea Stoica who welcomed me with kindness and generosity at Columbia University during the Summer.

I was delighted to work with a group of smart and fun people who gave supportive advice throughout my time in the INDY Lab. Thank you Alexandre, Arnout, Brunella, Christina, Daniyar, Ehsan, Emti, Farnood, Greg, Julien, Lars, Ljudmila, Mahsa, Mladen, Mohamed, and Sébastien. A very special thank you goes to William for being my fellow traveller during the sinuous journey of our theses. You were a motivation to come to the lab every day and were always there to support me when times were tough. I will keep our coffees in the lounge, beers at Sat', lunches at the food trucks, and discussions about work and life as some of the best memories of these years.

I am also grateful to everyone at Swiss Youth for Climate during the four years I was part of this amazing adventure. Making the world a better place is an arduous task, but being surrounded by incredible people makes it fun. This experience also taught me invaluable skills that I would never have learned behind my computer in the lab. I also extend thank yous to everyone involved in the Divest campaign at EPFL. I am proud that our efforts contributed to making our university more environmentally friendly.

Finally, I express my deep gratitude to my family. Thank you for your trust, understanding, and support. I am grateful, most of all, to Lydie-Line. Your empathy, sensitivity, humour, generosity, and love make everything in life enjoyable and possible—even a Ph.D. degree.

Lausanne, May 31, 2021 V. K.

Abstract

Poor decisions and selfish behaviors give rise to seemingly intractable global problems, such as the lack of transparency in democratic processes, the spread of conspiracy theories, and the rise in greenhouse gas emissions. However, people are more predictable than we think, and with machine-learning algorithms and sufficiently large datasets, we can design accurate models of human behavior in a variety of settings. In this thesis, to gain insight into social processes, we develop highly interpretable probabilistic choice-models. We draw from the econometrics literature on discrete-choice models and combine them with matrix factorization methods, Bayesian statistics, and generalized linear models. These predictive models enable interpretability through their learned parameters and latent factors.

First, we study the social dynamics behind group collaborations for the collective creation of content, such as in Wikipedia, the Linux kernel, and the European Union law-making process. By combining the Bradley-Terry and Rasch models with matrix factorization and natural language processing, we develop a model of edit acceptance in peer-production systems. We discover controversial components (*e.g.*, Wikipedia articles and European laws) and influential users (*e.g.*, Wikipedia editors and parliamentarians), as well as features that correlate with a high probability of edit acceptance. The latent representations capture non-linear interactions between components and users, and they cluster well into different topics (*e.g.*, historical figures and TV characters in Wikipedia, business and environment in European laws).

Second, we develop an algorithm for predicting the outcome of elections and of referenda by combining matrix factorization and generalized linear models. Our algorithm learns representations of votes and regions, which capture ideological and cultural voting patterns (*e.g.*, liberal/conservative, rural/urban), and it predicts the vote results in unobserved regions from partial observations. We test our model on voting data in Germany, Switzerland, and the US, and we deploy it on a Web platform to predict Swiss referendum votes in real-time. On average, our predictions reach a mean absolute error of 1% after observing only 5% of the regions.

Abstract

Third, we study how people perceive the carbon footprint of their day-to-day actions. We cast this problem as a comparison problem between pairs of actions (*e.g.*, the difference between flying across continents and using household appliances), and we develop a statistical model of relative comparisons reminiscent of the Thurstone model in psychometrics. The model learns the users' perception as the parameters of a Bayesian linear regression, which enables us to derive an active-learning algorithm to collect data efficiently. Our experiments show that users overestimate the emissions of low-footprint actions and underestimate those of high-footprint actions.

Finally, we design a probabilistic model of pairwise-comparison outcomes that capture a wide range of time dynamics. We achieve this by replacing the static parameters of a class of popular pairwise-comparison models with continuous-time Gaussian processes. We also develop an efficient inference algorithm that computes, with only a few linear-time iterations over the data, an approximate Bayesian posterior distribution.

Keywords discrete-choice models, matrix factorization, Bayesian statistics, generalized linear models, comparisons, choices, probabilistic models, data mining, machine learning, computational social science

Résumé

Les problèmes globaux, tels que le manque de transparence des processus démocratiques, la propagation de théories conspirationnistes ou l'augmentation des gaz à effet de serre, peuvent paraître imprévisibles et insolubles. Par contre, les êtres humains sont–heureusement–plus prévisibles que l'on ne pense. Grâce à des jeux de données massifs et de puissants algorithmes d'apprentissage automatique, il devient possible de modéliser une multitude de comportements sociaux. Dans cette thèse, nous développons des modèles probabilistes de choix individuels afin d'analyser ces comportements. Nous puisons dans la littérature des modèles de choix discrets, forts utilisés en économétrie, afin de rendre nos modèles interprétables, et nous les combinons avec des méthodes computationnelles, telles que la factorisation matricielle, les statistiques bayésiennes et les modèles linéaires généralisés, afin de les rendre plus performants.

Premièrement, nous étudions la dynamique des systèmes collaboratifs de création de contenu, tels que Wikipédia, le système d'opération Linux, et les lois du Parlement européen. Nous combinons les modèles de Bradley-Terry et de Rasch en un nouveau modèle qui nous permet de prédire si les modifications de contenu sont acceptées ou non (par la communauté Wikipédia ou par les autres parlementaires, par exemple). Ce modèle révèle quels sont les composants importants de ces systèmes, tels que les articles de Wikipédia controversés ou les parlementaires influents, ainsi que les facteurs qui augmentent la probabilité qu'une modification soit acceptée. Notre modèle inclut également des facteurs latents qui améliorent les performances de prédictions.

Deuxièmement, nous développons un algorithme de prédiction des résultats du vote populaire d'élections et de référendums à partir d'observations régionales partielles. Notre approche combine la factorisation matricielle et les modèles linéaires généralisés afin d'apprendre des représentations vectorielles des votes et des régions. Ces représentations capturent les biais d'influence, comme les biais culturels, linguistiques ou idéologiques. Nous appliquons notre modèle à des données de vote pour l'Allemagne, les États-Unis et la Suisse, et nous le déployons sur une plateforme en ligne pour prédire les votations suisses en temps réel. En moyenne, nos prédictions sont correctes à moins de 1% d'erreur en utilisant les résultats de seulement 5% des communes.

Troisièmement, nous nous intéressons à la perception que les gens ont de leur empreinte carbone. Nous formalisons ce problème sous forme de comparaisons entre deux actions (par exemple, prendre l'avion et utiliser un séchoir) et développons un modèle inspiré par l'approche de Thurstone en psychométrie. Le modèle apprend la perception générale d'une population d'individus en estimant les paramètres d'une régression linéaire bayésienne. Nos expériences montrent que les individus ont tendance à sur-estimer les actions à faible empreinte carbone et sous-estimer les actions à forte empreinte.

Finalement, nous développons un modèle probabiliste dynamique de comparaison par paires. Nous remplaçons les paramètres statiques d'une famille de modèles de comparaison par des processus gaussiens à temps continu. Nous développons également un algorithme d'inférence qui calcule une approximation bayésienne de la distribution postérieure du modèle de manière efficace, en quelques itérations à temps linéaire sur les données.

Mots-clés modèles de choix discrets, factorisation matricielle, statistiques bayésiennes, modèles linéaires généralisés, comparaisons, choix, modèles probabilistes, analyse de données, apprentissage automatique, sciences sociales computationnelles

Contents

Mathematical Notation

Symbol	Description
x	Plain lowercase letters denote scalar values.
$\boldsymbol{x} = [x_i]$	Boldface lowercase letter denote column vectors.
$\boldsymbol{X} = [x_{ij}]$	Boldface uppercase letters denote matrices.
$\mathcal{X}$	Calligraphic uppercase letters denote sets.
$\mathbf{R}, \mathbf{R}_{>0}, \mathbf{N}$	Number types: real, positive real and natural numbers, respectively.
$[N]$	Set of consecutive natural numbers $\{1, \ldots, N\}$.
$i \succ j$	Pairwise comparison outcome "i is chosen over j".
$i \succ \mathcal{A}$	Multiway comparison outcome "i is chosen among alternatives $\mathcal{A}$".
$\mathbf{P}(\mathcal{A})$	Probability of the event $\mathcal{A}$.
$\mathbf{1}_{\{\mathcal{A}\}}$	Indicator variable of the event $\mathcal{A}$.
$\mathbf{E}[x]$	Expectation of the random variable x.
$\mathbf{Var}[x]$	Variance of the random variable x.
$\mathbf{Cov}[x, y]$	Covariance of the random variables x and y.
$\sigma(x)$	Sigmoid function $\sigma : \mathbf{R} \to [0, 1]$, $\sigma(x) = 1/[1 + \exp(-x)]$.
$\mathcal{S}(x)$	Softmax function $\mathcal{S} : \mathbf{R}^K \to [0, 1]^K$, $\mathcal{S}(\boldsymbol{x})_i = \exp(x_i)/\sum_{k=1}^{K} \exp(x_k)$.
$O(f(x))$	$g(x) = O(f(x)) \iff \limsup_{x \to \infty} \lvert g(x) \rvert / f(x) < \infty$.

Distribution	Domain	Density or mass function $f(x)$
$\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	$\mathbf{R}^D$	$\frac{1}{\sqrt{2\pi\lvert\boldsymbol{\Sigma}\rvert}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$
$\mathrm{Gumbel}(\mu, \beta)$	$\mathbf{R}$	$\frac{1}{\beta}\exp\{-[z+\exp(-z)]\}$, where $z = \frac{x-\mu}{\beta}$
$\mathrm{Logistic}(\mu, \beta)$	$\mathbf{R}$	$\frac{\exp(-z)}{\beta\left[1+\exp(-z)\right]^2}$, where $z = \frac{x-\mu}{\beta}$
$\mathrm{Bernoulli}(p)$	$\{0, 1\}$	$p^x(1-p)^{1-x}$
$\mathrm{Categorical}(\boldsymbol{p}, K)$	$\{0, \ldots, K\}$	$\prod_{i=1}^{K} \mathbf{1}_{\{x=i\}} p_i$

1 Introduction

1.1 Motivation

Since the seminal work of Alan Turing establishing the foundations of modern computer science [185] and artificial intelligence [186], computers and algorithms have repeatedly accelerated the progress in science and engineering. Today, however, their effect on society is mixed. The US presidential election and the UK Brexit referendum in 2016 were marred by allegations of manipulation by the algorithms of the political consulting firm Cambridge Analytica. Misinformation, fake news, and conspiracy theories spread to millions of people within algorithmically recommended echo chambers on social media [107, 62, 156, 35], thus shaping collective action and political participation [119]. Although it reduces costs, increases economic outputs, and facilitates decision-making, the democratization of machine-learning algorithms in health, finance, surveillance, marketing, justice, and policy-making also tends to reinforce and exacerbate social biases [74, 171, 159]. Major breakthroughs in computer vision and natural language processing are obtained at a high environmental cost [172].

The lack of transparency in democratic processes, the spread of conspiracy theories, and the rise in greenhouse gas emissions are examples of seemingly intractable and unpredictable global problems stemming from the poor decisions and selfish behaviors of people. Fortunately, a century of research in econometrics and psychometrics has taught us that human decisions are more predictable than we think: From choosing between drinking tea or coffee in the morning to selecting which book to read before going to bed, human behavior is often reduced to making choices between a finite number of alternatives. Rooted in the work of Thurstone [179] and of Zermelo [206] in the 1920s, and later earning Daniel McFadden his Nobel Prize in economics [128], *discrete-choice theory* provides us with a toolset of statistical models for analyzing and forecasting decision-making processes.

Despite the progress that discrete-choice models made possible in studying consumer choices of transportation modes [11, 127], household energy suppliers [67], and college choices [61], their early application was mostly restricted to small-scale problems due to lack of data and of computational power. Coincidently, the emergence of the Internet and the World Wide Web in the second half of the 20^{th} century led to the collection of large datasets of human behavior. At work and at home, people spend countless hours behind their computers and their smartphones, where every click, tap, and mouse movement is recorded. The World Economic Forum [199] estimates that by 2025, the world will generate 463 exabytes[1] of data every day. In parallel, the rapid increase of computational power and the development of machine-learning algorithms have made it possible to process and analyze considerable amounts of data.

However, the architecture of modern machine-learning methods, belonging to the class of deep-learning algorithms, consists of many layers of non-linear transformations that progressively extract higher-level features from the data, and each layer consists of many parameters. This complex structure makes interpretation challenging: It is unclear what patterns the model has learned exactly [58, 70, 143, 83]. Hence, although these algorithms offer unprecedented predictive powers [109], they offer little insight into the problem itself, limiting any in-depth understanding of human behavior. Often, they are used as black boxes, *i.e.*, oracles that gobble up datasets and spit out predictions.

In this thesis, we focus on designing probabilistic models of decision-making that are highly interpretable. To study social processes, we draw from the literature on discrete-choice models and incorporate ideas from matrix factorization, Bayesian statistics, and generalized linear models. In particular, we ask the following research questions:

RQ1 Who are the important users and components in peer-production systems?

RQ2 What features of parliamentarians and laws increase the probability of law amendments being accepted?

RQ3 What ideological patterns are contained in voting data and how can they help predict elections and referenda?

RQ4 How do people perceive the carbon footprint of their actions?

RQ5 How can we learn pairwise-comparison models of time-dependent data?

We answer each question by designing a tailor-made probabilistic choice model. The learned parameters of each model enable us to interpret their predictions, thereby shedding light on the problem at hand. These models are also sufficiently general to be applicable in other contexts. Finally, we made our approach practical and our results useful by developing interactive Web platforms for RQ3, RQ4, and RQ5. These platforms are available to the general public and contribute to the global endeavour of opening science.

[1] This is 463×10^{18} bytes or 463 billion gigabytes, enough data to fill almost 100 billions DVDs.

1.2 Probabilistic Choice Models

1.2.1 A Brief History

The history of studying choices to understand human behaviour has its roots in the 1920s in the psychometrics community. Thurstone [179] pioneered the "law of comparative judgment" that established the methodology of measuring the perception of physical *stimuli* (*e.g.*, the weight of different objects) from pairwise comparisons. That same year, he used his new approach [180], today known as the *probit model*, to study people's perception of the seriousness of crimes, a notion for which no physical scale exists. Almost concurrently, Zermelo [206] proposed a similar model, known as the *logit model*, to rank chess players from outcomes of matches[2]. Zermelo's model was then independently rediscovered in the early 1950s in the statistics community by Bradley and Terry [19].

In the late 1950s, Marschak [120] introduced Thurstone's work to the econometrics community by interpreting the psychological stimuli of Thurstone's model as economic *random utility*. In parallel, Luce [114] proposed his *choice axiom* and the hypothesis of *independence of irrelevant alternatives* (IIA) that states that the relative comparison of two alternatives is unaffected by additions and subtractions of other alternatives. In other words, it assumes that the alternatives are uncorrelated. This property enabled Luce to extend the logit model to multi-way comparisons. This extension was also proposed by McFadden [126] to introduce the *multinomial logit model*[3] from a random utility viewpoint.

The subsequent decades were dedicated to extending discrete-choice models. In particular, to relax the (rather restrictive) IIA hypothesis, Ben-Akiva [11] and Williams [198] developed the *nested logit model* that encodes correlation between alternatives through their joint distribution. Similarly, Boyd and Mellman [17] and Cardell and Dunbar [25] developed the *mixed logit model*, which encodes correlation by assigning a probability distribution to the parameters of the (multinomial) logit model [80]. Some efforts were also deployed by Yellot [204] to unify the different formulations of the logit model. In parallel, pairwise-comparison data started to be exploited for *ranking* [59, 24, 148, 192, 135], a model often referred to as the *Plackett-Luce model* [80]. Research addressing the inference of discrete-choice models was also conducted for *sampling and simulations* [118, 39], *maximum likelihood estimation* [77, 89, 122, 190], and *Bayesian inference* [71, 26, 87].

Today, the availability of unprecedented computational power and of large-scale datasets has enabled new applications of discrete-choice models. In reinforcement learning, Sadigh et al. [161] and Christiano et al. [31] propose to use pairwise-comparison models to incorporate feedback from human supervisors into the reward function. Ammar [5] suggests using these models to make personalized recommendations. Chumbalov et al.

[2]This approach is still used today by the World Chess Federation [53].
[3]This model was first introduced as the *conditional logit model*.

[33] propose a search algorithm for navigating large-scale databases of complex items (*e.g.*, images) from pairwise comparisons. To make algorithmic policies, Lee et al. [110] apply the Plackett-Luce model in a *virtual democracy* setting to learn people's preferences. Noothigattu et al. [139] also use this model to train autonomous vehicles to make ethical decisions. Finally, Salganik and Levy [162] implemented the probit model into the online platform *All Our Ideas*[4] to help the New York City Mayor's Office of Long-Term Planning and Sustainability understand New Yorkers' preferences for developing the city sustainably.

A history of the development of discrete-choice models in econometrics is given by McFadden [128] in his Nobel-Prize lecture. The curious reader will find more details about random utility models in the books of Train [181, Chapter 1] and Hensher et al. [81, Chapter 3]. An introduction to probabilistic models of choice from a statistical perspective is given by Maystre [121, Chapter 1]. In the next section, we introduce discrete-choice models from a *random utility* perspective.

1.2.2 Random Utility Models

Choice Set

Discrete-choice models capture people's preferences that drive their choices. When facing a set of (at least two) alternatives, a decision-maker chooses one of the alternatives over the other(s). This set of alternatives is defined as the *choice set*.

Definition (Choice Set)**.** Given the set of all possible alternatives $\mathcal{A}$, the *choice set* $\mathcal{C} \subseteq \mathcal{A}$ is the set of alternatives faced by a decision-maker. It has the following three characteristics:

1. The alternatives are *mutually exclusive*.
2. The choice set $\mathcal{C}$ is *exhaustive*.
3. The number of alternatives is *finite*.

The exclusiveness of alternatives means that, when choosing alternative $i \in \mathcal{C}$, the other alternatives of $\mathcal{C}$ are left aside. The exhaustiveness of the choice set means that the decision-maker faces all possible alternatives at decision time. Finally, the number of alternatives must be finite: The decision-maker can count the alternatives.

The first two characteristics are not restrictive, because it is always possible to add artificial alternatives. For example, if $\mathcal{C} = \{i, j\}$, exclusiveness can be ensured by adding a third alternative $k =$ "choose i and j". This enables the decision-maker to choose both

[4] http://www.allourideas.org

alternatives at the same time. Similarly, exhaustiveness can be ensured by adding another alternative $l =$ "none of the alternatives". This enables the decision-maker to choose none of the alternatives, hence making the choice set exhaustive.

The third characteristic is, however, restrictive. A finite number of alternatives is actually the defining characteristic of discrete-choice models. This contrasts with regression models, in which the target variable is continuous, hence the number of alternatives is infinite. The choice set can also vary for each choice faced by a decision-maker. For example, $\mathcal{C}_1 = \{i, j\}$ and $\mathcal{C}_2 = \{i, j, k\}$. This contrasts with classification models, in which the choice set, *i.e.*, the domain of the target variable, is identical for every observation. For example, in the context of email classification, $\mathcal{C}_1 = \mathcal{C}_2 = \{\text{“spam”}, \text{“ham”}\}$.

Random Utility

Without loss of generality, we introduce the random utility models from an econometrics viewpoint, *i.e.*, by analyzing the behavior of decision-makers facing choices. These methods can obviously be used to model other processes. In particular, and as we will see in this thesis, they can model *implicit* choices. For example, in the context of collective-sport matches, if Team A wins against Team B, then Team A is implicitly chosen over Team B (*e.g.*, because it played better or had good lucky).

In econometrics, we posit that a decision-maker is rational and chooses the alternative that maximizes its personal gain. For example, let us consider a sleepy Ph.D. student who needs a hot beverage in the morning in order to start working on their research. Let $\boldsymbol{x}_i \in \mathbf{R}^M$ be a vector of M observable features that might influence a person's decision to choose alternative $i \in \mathcal{C}$, and let $\boldsymbol{w} \in \mathbf{R}^M$ be the associated M-dimensional parameter vector. In our example, the choice set is $\mathcal{C} = \{\text{“espresso”}, \text{“cappuccino”}, \text{“Earl Grey tea”}\}$ and the features could include the type of beverage (coffee or tea), the level of caffeine, and the preparation time. The feature vector could also include features of the student, such as their age, their gender, and their baseline level of glucose.

It is impossible to characterize *all* features that influence a decision-maker's choice. Therefore, we capture the effect of the unobserved features in a random noise variable ϵ_i, whose probability distribution is to be defined. In our example, the noise could capture the effect of the atmospheric pressure on the quality of the brew and the influence of the student's personal history on their choice[5].

To analyze the decision-maker's behaviour, economists posit that an alternative i has a *random utility* U_i given by

$$U_i = \boldsymbol{x}_i^\intercal \boldsymbol{w} + \epsilon_i,$$

[5]The student could have a preference for Earl Grey tea because this reminds them of their grandfather.

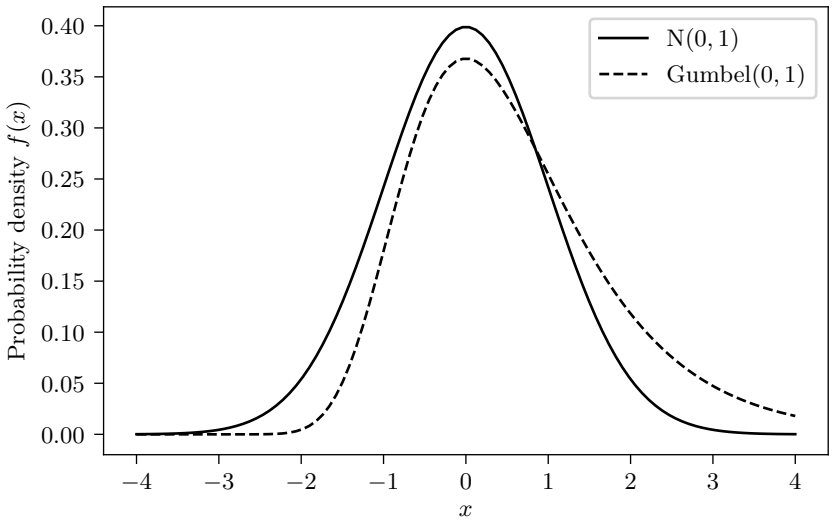


Figure 1.1 – Probability density functions of Gaussian and Gumbel distributions.

and the decision-maker chooses alternative i if $U_i > U_j$, for all $j \neq i$, $i, j \in \mathcal{C}$. Hence, the probability that a decision-maker chooses i over j is

$$\begin{aligned}\mathbf{P}\left(i \succ j\right) &:= \mathbf{P}\left(U_i > U_j\right) \\ &= \mathbf{P}\left(\boldsymbol{x}_i^\intercal \boldsymbol{w} + \epsilon_i > \boldsymbol{x}_j^\intercal \boldsymbol{w} + \epsilon_j\right) \\ &= \mathbf{P}\left(\epsilon_i - \epsilon_j > \boldsymbol{x}_j^\intercal \boldsymbol{w} - \boldsymbol{x}_i^\intercal \boldsymbol{w}\right). \end{aligned} \tag{1.1}$$

The probability $\mathbf{P}\left(i \succ j\right)$ is called the *choice probability*. The notation "$i \succ j$" reads as "alternative i is chosen over alternative j" or, equivalently, as "i wins over j". In our example, we are interested in the probability $\mathbf{P}$ ("capuccino" $\succ$ "espresso") that the student will choose to drink a cappuccino instead of an espresso.

From (1.1), the characterization of a discrete-choice model depends on the researcher's hypotheses on the noise model, *i.e.*, on the probability distribution that captures the unobserved features best. Two popular choices of distribution, which we describe below, are (i) the Gaussian distribution $\mathrm{N}(\mu, \sigma^2)$ with mean μ and variance σ^2, and (ii) the Gumbel distribution $\mathrm{Gumbel}(\mu, \beta)$ with location μ and scale β. The probability density function of these two distributions are

$$\begin{aligned} f_{\text{Gaussian}}(x) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \\ f_{\text{Gumbel}}(x) &= \frac{1}{\beta} \exp\{-[z + \exp(-z)]\}, \end{aligned}$$

where $z = \frac{x-\mu}{\beta}$. We show in Figure 1.1 an example of these two distributions with $\mu = 0$ and $\sigma^2 = \beta = 1$.

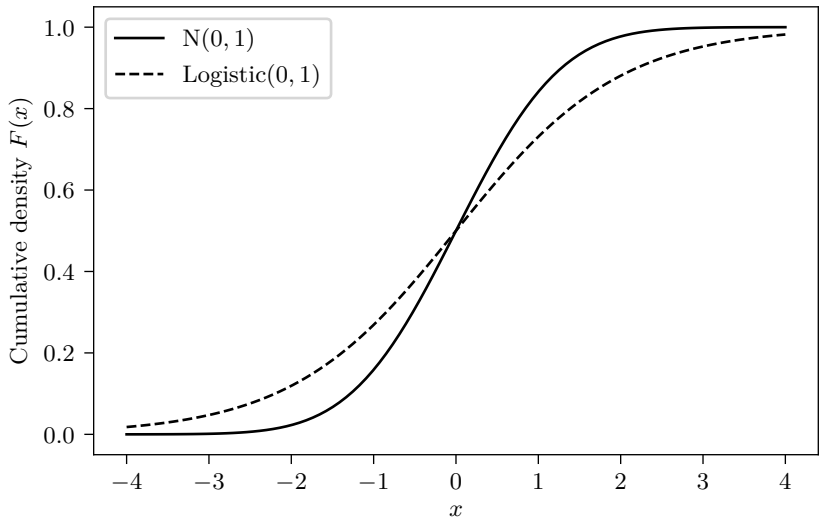


Figure 1.2 – Cumulative density functions of Gaussian and logistic distributions.

Probit Model The probit model was first introduced by Thurstone [179] in the context of psychometrics. In this model, the random noise is independently and identically distributed (i.i.d.) with a Gaussian distribution $\epsilon_i, \epsilon_j \sim \mathrm{N}(0, 0.5)$. As the difference of two Gaussian random variables is also Gaussian, *i.e.*, $\epsilon_i - \epsilon_j \sim \mathrm{N}(0, 1)$ in this special case, the choice probability for the probit model is

$$\mathbf{P}\left(i \succ j\right) = \mathbf{P}\left(\epsilon_i - \epsilon_j > \boldsymbol{x}_j^\intercal \boldsymbol{w} - \boldsymbol{x}_i^\intercal \boldsymbol{w}\right) = \Phi\left(\boldsymbol{x}_i^\intercal \boldsymbol{w} - \boldsymbol{x}_j^\intercal \boldsymbol{w}\right), \tag{1.2}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, as shown in Figure 1.2. When the random utility is parameterized by only one parameter, each alternative i is represented by a one-dimensional parameter $w_i \in \mathbf{R}$. The feature vector $\boldsymbol{x}_i$ becomes a one-hot vector that is 0 everywhere except in i, where it is 1, *i.e.*, it "selects" the parameter associated with alternative i. Then, $U_i = \boldsymbol{x}_i^\intercal \boldsymbol{w} + \epsilon_i = w_i + \epsilon_i$, and the model

$$\mathbf{P}\left(i \succ j\right) = \Phi\left(w_i - w_j\right)$$

is called the *Thurstone model*. The M parameters $\boldsymbol{w} = [w_1 \cdots w_M]^\intercal$ represent a *score* for each of the M alternative. They can be interpreted as the *perceived psychological stimuli* of the alternatives and induce a natural ranking.

Logit Model The logit model was introduced by Zermelo [206] and rediscovered two decades later by Bradley and Terry [19]. In this model, the random noise is assumed to follow a Gumbel distribution[6] $\epsilon_i \sim \mathrm{Gumbel}(\mu_i, \beta_i)$ (see Figure 1.1). The Gumbel distribution has the property that the difference of two Gumbel random variables G_1

[6]This distribution is also called the (Type I) extreme value distribution.

and G_2 with locations μ_1 and μ_2, and scales $\beta_1 = \beta_2 = \beta$, follows a logistic distribution $G_1 - G_2 \sim \text{Logistic}(\mu', \beta')$, whose cumulative density function is

$$F_{\text{Logistic}}(x) = \sigma\left(\frac{x-\mu'}{\beta'}\right) = \frac{1}{1+\exp\left[-\frac{x-\mu'}{\beta'}\right]},$$

where $\mu' = \mu_1 - \mu_2$ and $\beta' = \beta$, and where $\sigma(\cdot)$ is the logistic function. We show, in Figure 1.2, an example of the logistic cumulative distribution function with $\mu' = 0$ and $\beta' = 1$, which we compare with that of the standard normal distribution.

In the logit model, the random noise is assumed to be i.i.d. with a Gumbel distribution $\epsilon_i, \epsilon_j \sim \text{Gumbel}(0, 1)$. Hence, $\epsilon_i - \epsilon_j \sim \text{Logistic}(0, 1)$, and the choice probability for this model is

$$\begin{aligned}\mathbf{P}\left(i \succ j\right) &= \mathbf{P}\left(\epsilon_i - \epsilon_j > \boldsymbol{x}_j^\intercal \boldsymbol{w} - \boldsymbol{x}_i^\intercal \boldsymbol{w}\right) \\ &= \frac{1}{1+\exp[-(\boldsymbol{x}_i^\intercal \boldsymbol{w} - \boldsymbol{x}_j^\intercal \boldsymbol{w})]} \\ &= \frac{\exp(\boldsymbol{x}_i^\intercal \boldsymbol{w})}{\exp(\boldsymbol{x}_i^\intercal \boldsymbol{w}) + \exp(\boldsymbol{x}_j^\intercal \boldsymbol{w})}. \end{aligned} \tag{1.3}$$

When the random utility is parameterized by only one parameter, the model

$$\mathbf{P}\left(i \succ j\right) = \frac{1}{1+\exp[-(w_i - w_j)]} \tag{1.4}$$

is called the *Bradley-Terry model*. In this scenario, the parameters $\boldsymbol{w}$ can be interpreted as the intrinsic *strengths* of each alternative.

Multinomial Logit Model The multinomial logit model, also called conditional logit model, was introduced by Luce [114] and by McFadden [126]. In the probit and logit models, the decision-maker faces a binary choice, *i.e.*, the size of the choice set is $|\mathcal{C}| = 2$. In the multinomial logit model, the decision-maker faces multiple alternatives, and the choice set $\mathcal{C} = \{i, j, \ldots, k\}$ has more than two elements. The random noise is also assumed to be i.i.d. with the Gumbel distribution, so that the choice probability is

$$\mathbf{P}\left(i \succ \mathcal{C}\right) = \mathbf{P}\left(U_i > U_j, \ldots, U_i > U_k\right) = \frac{\exp(\boldsymbol{x}_i^\intercal \boldsymbol{w})}{\sum_{j\in\mathcal{C}} \exp(\boldsymbol{x}_j^\intercal \boldsymbol{w})}. \tag{1.5}$$

The notation "$i \succ \mathcal{C}$" reads as "alternative i is chosen among all alternatives in the choice set $\mathcal{C}$".

Rasch Model Although not categorized as a discrete-choice model, the Rasch model [152] is closely related to the Bradley-Terry model. We present it here because in Chapter 3 we combine it with the multinomial logit model. This model was introduced in the context of *item response theory* in order to measure people's ability to answer tests and understand the traits that explain their performance. It assumes that an individual u taking a test has an intrinsic strength $s_u \in \mathbf{R}$, and that a question i in the test has an intrinsic difficulty $d_i \in \mathbf{R}$. The probability that individual u answers question i correctly is

$$\mathbf{P}\left(u \succ i\right) = \frac{1}{1 + \exp[-(s_u - d_i)]}. \tag{1.6}$$

The relation with the Bradley-Terry model is obvious comparing (1.4) and (1.6).

Independence of Irrelevant Alternatives The hypothesis property of *independence of irrelevant alternatives* (IIA) was first formulated by Luce [114]. It states that, for any two alternatives i and j, the ratio of the multinomial logit probabilities from (1.5) is independent of alternatives other than i and j, *i.e.*,

$$\frac{\mathbf{P}\left(i \succ \mathcal{C}\right)}{\mathbf{P}\left(j \succ \mathcal{C}\right)} = \frac{\exp(\boldsymbol{x}_i^\intercal \boldsymbol{w}) / \sum_{k \in \mathcal{C}} \exp(\boldsymbol{x}_k^\intercal \boldsymbol{w})}{\exp(\boldsymbol{x}_j^\intercal \boldsymbol{w}) / \sum_{k \in \mathcal{C}} \exp(\boldsymbol{x}_k^\intercal \boldsymbol{w})} = \exp(\boldsymbol{x}_i^\intercal \boldsymbol{w} - \boldsymbol{x}_j^\intercal \boldsymbol{w}). \tag{1.7}$$

As this ratio depends only on alternatives i and j, adding or removing alternatives from the choice set $\mathcal{C}$ will leave it unchanged. This is a powerful result, because it implies that not all alternatives are necessary in order to obtain an estimate of the associated parameters. As a result, under the multinomial logit model, (i) the computational cost of estimating the parameters of many alternatives can be reduced by sub-sampling the alternatives and (ii) if one is interested only in analyzing alternatives $i, j \in \mathcal{A}$, the other alternatives $k \in \mathcal{A} - \{i, j\}$ are irrelevant. If the multinomial logit model exhibits this property, it is also because Luce proved that this model stems for the IIA hypothesis.

As shown bellow by the blue-bus/red-bus paradox, however, the IIA assumption is restrictive. Suppose that a population of suburban residents must choose a mode of transportation for they daily daily commute. Their probability of taking the bus, which is blue, compared to commuting by car is $\mathbf{P}\left(\text{“blue bus”} \succ \text{“car”}\right) = 1/3$, hence $\mathbf{P}\left(\text{“car”} \succ \text{“blue bus”}\right) = 2/3$. The ratio between these two probabilities is equal to 2. Suppose now that the city adds a new red bus to their fleet. Although this should not affect the probability of commuters to use their car[7], the multinomial logit model predicts that

$$\mathbf{P}\left(\text{“blue bus”} \succ \text{“car”}\right) = \mathbf{P}\left(\text{“red bus”} \succ \text{“car”}\right) = \frac{1}{4},$$
$$\mathbf{P}\left(\text{“car”} \succ \text{“blue/red bus”}\right) = \frac{1}{2},$$

[7] Assuming, of course, that the bus frequency remains the same.

so that the ratio is still equal to 2. It should be expected, however, that the probability of a person using their car is unaffected by this new bus, *i.e.*,

$$\mathbf{P}\left(\text{“blue bus”} \succ \text{“car”}\right) = \mathbf{P}\left(\text{“red bus”} \succ \text{“car”}\right) = \frac{1}{6},$$
$$\mathbf{P}\left(\text{“car”} \succ \text{“blue/red bus”}\right) = \frac{2}{3}.$$

In this case, the ratio is equal to 4.

To circumvent this issue, new choice models were proposed in the econometrics literature. For example, the *nested logit model*, *multinomial probit model*, and *mixed logit model* all relax the IIA hypothesis by enabling correlated alternatives. The nested logit and multinomial probit models assume that the random noise terms ϵ_i are correlated through their joint distribution. The mixed logit model enables the parameters $\boldsymbol{w}$ to be random by assigning them a probability distribution. An introduction to these models is given by Train [181].

1.3 Outline and Contributions

In this thesis, we seek to gain new insight into the structure and dynamics of social processes, such as peer-production, law-making, and voting. To answer the research questions of Section 1.1, we

1. collect *rich* datasets,
2. build *interpretable* predictive models, and
3. design *efficient* learning algorithms.

Our models are tailored to the datasets, and the learned parameters enable us to interpret their predictions, thereby gaining insight into the studied processes. We also make our approaches practical and our results useful by deploying them on online Web platforms.

More specifically, in Chapter 2, we ask who are the important users and components in online peer-production systems. We take a predictive viewpoint and posit that the probability of acceptance of user contributions depends on the skill of users and the inertia of components resisting to change. We model this probability with a discrete-choice model inspired from the Rasch model, and we include latent factors reminiscent of collaborative filtering to capture non-linear interactions between users and components. We apply our model to Wikipedia and to the Linux kernel, two examples of large-scale peer-production systems, and we discover interesting structures in the data: We identify controversial Wikipedia articles and core Linux components that are crucial to the functioning of

the system. Finally, the latent factors boost the predictive performance and cluster well according to topics of the Wikipedia articles.

In Chapter 3, we shift our attention to law-making processes that we study through the lens of peer-production systems. In the European Union, parliamentarians shape policies by proposing amendments to law drafts. We look for features of the parliamentarians, the amendments, and the laws that increase the probability of amendments being accepted. We start by collecting a new dataset of 450 000 legislative edits proposed by European parliamentarians between 2009 and 2019. Then, we predict the acceptance probability of amendments by building a model inspired from the multinomial logit model and the Rasch model. Our approach takes advantage of the conflictive structure of amendments that modify the same parts of the same laws. We identify that being in charge of a law draft and that proposing shorter amendments are among the features that correlate with highest probability of acceptance. We also discover words and bigrams that are predictive of acceptance or rejection when inserted or deleted, such as the term "human rights" that predicts acceptance when deleted from the law.

In Chapter 4, we study one of the most fundamental choice processes in our society: voting. To understand voting patterns, we develop an algorithm for predicting aggregate vote outcomes (*e.g.*, national) from partial results (*e.g.*, regional) that are revealed sequentially. We combine matrix factorization and generalized linear models to obtain a flexible, efficient, and accurate algorithm. Our experiments show that this approach accurately predicts the outcomes of Swiss referenda, U.S. presidential elections, and German legislative elections. We also show that the learned latent factors correspond to clear ideological and cultural patterns, such as conservative/liberal and rural/urban patterns. Finally, we deploy our algorithm on an online Web platform to provide real-time vote predictions in Switzerland and a data-visualization tool to explore voting behavior.

In Chapter 5, we study people's perception of their carbon footprint. Driven by the observation that few people think of CO_2 impact in absolute terms, we design a system to probe their perception from simple pairwise comparisons of the relative carbon footprint of their actions. We design a Web interface to collect 2000 answers from 200 users on our university campus. We develop a Bayesian model inspired from the probit model that enables us to take an active-learning approach to selecting the pairs of actions that are maximally informative about the model parameters, hence making data collection more efficient. The parameters capture the perceived carbon footprint of the actions and induce a natural ranking to compare them with the true values. This reveals an interesting pattern: Low-impact actions are usually overestimated and high-impact actions are usually underestimated.

Finally, in Chapter 6, we address the problem of learning choices in a dynamic setting, where alternatives are correlated over time. We solve this by replacing the static parameters of the logit model by continuous-time Gaussian processes, whose covariance

function enables expressive time dynamics. We develop an efficient inference algorithm that computes an approximate Bayesian posterior distribution. Despite the flexibility of our model, our inference algorithm requires only a few linear-time iterations over the data. We apply our model to several historical databases of sports outcomes and find that our approach (a) outperforms competing approaches in terms of predictive performance, (b) scales to millions of observations, and (c) generates compelling visualizations that help in understanding and interpreting the data. We also develop a Web platform that uses our algorithm to make predictions for football matches in European leagues and international competitions.

2 Peer-Production Systems

In this chapter[1], we develop a discrete-choice model inspired from the Rasch model and including ideas reminiscent of collaborative filtering to predict user contributions to online peer-production systems. As the number of contributors to these systems grows, it becomes increasingly important to predict whether the edits that users make will eventually be beneficial to the project. Existing solutions either rely on a user reputation system or consist of a highly specialized predictor that is tailored to a specific peer-production system. We explore a different point in the solution space that goes beyond user reputation but does not involve any content-based feature of the edits. We posit that the probability that an edit is accepted is a function of the editor's skill, of the difficulty of editing the component and of a user-component interaction term. Our model is broadly applicable, as it only requires observing data about *who* makes an edit, *what* the edit affects and whether the edit survives or not. We apply our model on Wikipedia and the Linux kernel, two examples of large-scale peer-production systems, and we seek to understand whether it can effectively predict edit survival: in both cases, we provide a positive answer. Our approach significantly outperforms those based solely on user reputation and bridges the gap with specialized predictors that use content-based features. It is simple to implement, computationally inexpensive, and in addition it enables us to discover interesting structure in the data[2].

2.1 Introduction

Over the last two decades, the number and scale of online peer-production systems has become truly massive, driven by better information networks and advances in collaborative software. At the time of writing, 128 643 editors contribute regularly to 5+ million articles of the English Wikipedia [197] and over 15 600 developers have authored code for the

[1] This chapter is based on Yardım et al. [201].

[2] Data and code publicly available on https://github.com/lca4/interank.

Linux kernel [37]. On GitHub, 24 million users collaborate on 25.3 million active software repositories [63].

In order to ensure that such projects advance towards their goals, it is necessary to identify whether edits made by users are beneficial. As the number of users and components of the project grows, this task becomes increasingly challenging. In response, two types of solutions are proposed. On the one hand, some advocate the use of *user reputation systems* [155, 2]. These systems are general, their predictions are easy to interpret and can be made resistant to manipulations [44]. On the other hand, a number of highly specialized methods are proposed to automatically predict the quality of edits in particular peer-production systems [48, 75]. These methods can attain excellent predictive performance [79] and usually significantly outperform predictors that are based on user reputation alone [48], but they are tailored to a particular peer-production system, use domain-specific features and rely on models that are difficult to interpret.

In this work, we set out to explore another point in the solution space. We aim to keep the generality and simplicity of user reputation systems, while reaching the predictive accuracy of highly specialized methods. We ask the question: Can one predict the outcome of contributions simply by observing *who edits what* and whether the edits eventually survive? We address this question by proposing a novel statistical model of edit outcomes. We formalize the notion of collaborative project as follows. N users can propose edits on M distinct items (components of the project, such as articles on Wikipedia or a software's modules), and we assume that there is a process for validating edits (either immediately or over time). We observe triplets (u, i, q) that describe a user $u \in \{1, \dots, N\}$ editing an item $i \in \{1, \dots, M\}$ and leading to outcome $q \in \{0, 1\}$; the outcome $q = 0$ represents a rejected edit, whereas $q = 1$ represents an accepted, beneficial edit. Given a dataset of such observations, we seek to learn a model of the probability $\mathbf{P}(u \succ i)$ that an edit made by user u on item i is accepted. This model can then be used to help moderators and project maintainers prioritize their efforts once new edits appear: For example, edits that are unlikely to survive could be sent out for review immediately.

Our approach borrows from probabilistic models of pairwise comparisons [206, 151]. These models learn a real-valued score for each object (user or item) such that the difference between two objects' scores is predictive of comparison outcomes. We take a similar perspective and view each edit in a collaborative project as a game between the user who tries to effect change and the item that resists change[3]. Similarly to pairwise-comparison models, our approach learns a real-valued score for each user and each item. In addition, it also learns latent features of users and items that capture interaction effects.

[3]Obviously, items do not really "resist" by themselves. Instead, this notion should be taken as a proxy for the combined action of other users (e.g., project maintainers) who can accept or reject an edit depending, among others, on standards of quality.

In contrast to quality-prediction methods specialized on a particular peer-production system, our approach is general and can be applied to any system in which users contribute by editing discrete items. It does not use any explicit content-based features: instead, it simply learns by observing triplets $\{(u, i, q)\}$. Furthermore, the resulting model parameters can be interpreted easily. They enable a principled way of (a) ranking users by the quality of their contributions, (b) ranking items by the difficulty of editing them and (c) understanding the main dimensions of the interaction between users and items.

We apply our approach on two different peer-production systems. We start with Wikipedia and consider its Turkish and French editions. Evaluating the accuracy of predictions on an independent set of edits, we find that our model approaches the performance of the state of the art. More interestingly, the model parameters reveal important facets of the system. For example, we characterize articles that are easy or difficult to edit, respectively, and we identify clusters of articles that share common editing patterns. Next, we turn our attention to the Linux kernel. In this project, contributors are typically highly skilled professionals, and the edits that they make affect 394 different subsystems (kernel components). In this instance, our model's predictions are *more accurate* than a random forest classifier trained on domain-specific features. In addition, we give an interesting qualitative description of subsystems based on their difficulty score.

In short, our paper (a) gives evidence that observing *who edits what* can yield valuable insights into peer-production systems and (b) proposes a statistically grounded and computationally inexpensive method to do so. The analysis of two peer-production systems with very distinct characteristics demonstrates the generality of the approach.

Organization of the Paper We start by reviewing related literature in Section 2.2. In Section 2.3, we describe our statistical model of edit outcomes and briefly discuss how to efficiently learn a model from data. In Sections 2.4 and 2.5, we investigate our approach in the context of Wikipedia and of the Linux kernel, respectively. Finally, we conclude in Section 2.6.

2.2 Related Work

With the growing size and impact of online peer-production systems, the task of assessing contribution quality has been extensively studied. We review various approaches to the problem of quantifying and predicting the quality of user contributions and contrast them to our approach.

User Reputation Systems Reputation systems have been a long-standing topic of interest in relation to peer-production systems and, more generally, in relation to

online services [155]. Adler and de Alfaro [2] propose a point-based reputation system for Wikipedia and show that reputation scores are predictive of the future quality of editing. As almost all edits to Wikipedia are immediately accepted, the authors define an *implicit* notion of edit quality by measuring how much of the introduced changes is retained in future edits. The ideas underpinning the computation of implicit edit quality are extended and refined in subsequent papers [3, 44]. This line of work leads to the development of WikiTrust [45], a browser add-on that highlights low-reputation texts in Wikipedia articles. When applying our methods to Wikipedia, we follow the same idea of measuring quality implicitly through the state of the article at subsequent revisions. We also demonstrate that by automatically learning properties of the *item* that a user edits (in addition to learning properties of the user, such as a reputation score) we can substantially improve predictions of edit quality. This was also noted by Tabibian et al. [176] in a setting similar to ours, but using a temporal point process framework.

Specialized Classifiers Several authors propose quality-prediction methods tailored to a specific peer-production system. Typically, these methods consist of a machine-learned classifier trained on a large number of content-based and system-based features of the users, the items and the edits themselves. Druck et al. [48] fit a maximum entropy classifier for estimating the lifespan of a given Wikipedia edit, using a definition of edit longevity similar to that of Adler and de Alfaro [2]. They consider features based on the edit's content (such as: number of words added / deleted, type of change, capitalization and punctuation, etc.) as well as features based on the user, the time of the edit and the article. Their model significantly outperforms a baseline that only uses features of the user. Other methods use support vector machines [22], random forests [22, 92] or binary logistic regression [149], with varying levels of success. In some cases, content-based features are refined using natural-language processing, leading to substantial performance improvements. However, these improvements are made to the detriment of general applicability. For example, competitive natural language processing tools have yet to be developed for the Turkish language (we investigate the Turkish Wikipedia in Section 2.4). In contrast to these methods, our approach is general and broadly applicable. Furthermore, the use of black-box classifiers can hinder the interpretability of predictions, whereas we propose a statistical model whose parameters are straightforward to interpret.

Truth Inference In crowdsourcing, a problem related to ours consists of *jointly* estimating (a) model parameters (such as user skills or item difficulties) that are predictive of contribution quality, and (b) the quality of each contribution, without ground truth [43]. Our problem is therefore easier, as we assume access to ground-truth information about the outcome (quality) of past edits. Nevertheless, some methods developed in the crowdsourcing context [195, 193, 208] provide models that can be applied to our setting as well. In Sections 2.4 and 2.5, we compare our models to GLAD [195].

Pairwise Comparison Models Our approach draws inspiration from probabilistic models of pairwise comparisons, as described in Section 1.2. The main paradigm posits that every object i has a latent *strength* (skill or difficulty) parameter w_i, and that the probability $\mathbf{P}(i \succ j)$ of observing object i "winning" over object j increases with the distance $w_i - w_j$. Conceptually, our model is closest to that of Rasch [151].

Collaborative Filtering Our method also borrows from collaborative filtering techniques popular in the recommender systems community. In particular, some parts of our model are remindful of matrix-factorization techniques [99]. These techniques automatically learn low-dimensional embeddings of users and items based on ratings, with the purpose of producing better recommendations. Our work shows that these ideas can also be helpful in addressing the problem of predicting outcomes of edits in peer-production systems. Like collaborative-filtering methods, our approach is exposed to the *cold-start* problem: with no (or few) observations about a given user or item, the predictions are notably less accurate. In practice, this problem can be addressed, e.g., by using additional features of users and / or items [164, 108] or by clustering users [112].

2.3 Statistical Models

In this section, we describe and explain two variants of a statistical model of edit outcomes based on *who* edits *what*. In other words, we develop models that are predictive of the outcome $q \in \{0, 1\}$ of a contribution of user u on item i. To this end, we represent the probability $\mathbf{P}(u \succ i)$ that an edit made by user u on item i is successful. In collaborative projects of interest, most users typically interact with only a small number of items. In order to deal with the sparsity of interactions, we postulate that the probabilities $\{\mathbf{P}(u \succ i)\}$ lie on a low-dimensional manifold and propose two model variants of increasing complexity. In both cases, the parameters of the model have intuitive effects and can be interpreted easily.

Basic Variant The first variant of our model is directly inspired by the Rasch model [151]. The probability that an edit is accepted is defined as

$$\mathbf{P}(u \succ i) = \frac{1}{1 + \exp[-(s_u - d_i + b)]}, \tag{2.1}$$

where $s_u \in \mathbf{R}$ is the *skill* of user u, $d_i \in \mathbf{R}$ is the *difficulty* of item i, and $b \in \mathbf{R}$ is a global parameter that encodes the overall skew of the distribution of outcomes. We call this model variant INTERANK *basic*. Intuitively, the model predicts the outcome of a "game" between an item with inertia and a user who would like to effect change. The *skill* quantifies the ability of the user to enforce a contribution, whereas the *difficulty* quantifies how "resistant" to contributions the particular item is.

Similarly to reputation systems [2], INTERANK *basic* learns a score for each user; this score is predictive of edit quality. However, unlike these systems, our model also takes into account that some items might be more challenging to edit than others. For example, on Wikipedia, we can expect high-traffic, controversial articles to be more difficult to edit than less popular articles. As with user skills, the article difficulty can be inferred *automatically* from observed outcomes.

Full Variant Although the *basic* variant is conceptually attractive, it might prove to be too simplistic in some instances. In particular, the *basic* variant implies that if user u is more skilled than user v, then $\mathbf{P}(u \succ i) > \mathbf{P}(v \succ i)$ for *all* items i. In many peer-production systems, users tend to have their own specializations and interests, and each item in the project might require a particular mix of skills. For example, with the Linux kernel, an engineer specialized in file systems might be successful in editing a certain subset of software components, but might be less proficient in contributing to, say, network drivers, whereas the situation might be exactly the opposite for another engineer. In order to capture the multidimensional interaction between users and items, we add a bilinear term to the probability model (2.1). Letting $\boldsymbol{x}_u, \boldsymbol{y}_i \in \mathbf{R}^D$ for some dimensionality $D \in \mathbf{N}_{>0}$, we define

$$\mathbf{P}(u \succ i) = \frac{1}{1 + \exp[-(s_u - d_i + \boldsymbol{x}_u^\top \boldsymbol{y}_i + b)]}. \tag{2.2}$$

We call the corresponding model variant INTERANK *full*. The vectors $\boldsymbol{x}_u$ and $\boldsymbol{y}_i$ can be thought of as embedding users and items as points in a latent D-dimensional space. Informally, $\mathbf{P}(u \succ i)$ increases if the two points representing a user and an item are close to each other, and it decreases if they are far from each other (e.g., if the vectors have opposite signs). If we slightly oversimplify, the parameter $\boldsymbol{y}_i$ can be interpreted as describing the set of skills needed to successfully edit item i, whereas $\boldsymbol{x}_u$ describes the set of skills displayed by user u.

The bilinear term is reminiscent of matrix-factorization approaches in recommender systems [99]; indeed, this variant can be seen as a *collaborative-filtering* method. In true collaborative-filtering fashion, our model is able to learn the latent feature vectors $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{y}_i\}$ *jointly*, by taking into consideration all edits and without any additional content-based features.

Finally, note that the skill and difficulty parameters are retained in this variant and can still be used to explain first-order effects. The bilinear term explains only the additional effect due to the user-item interaction.

2.3.1 Learning the Model

From (2.1) and (2.2), it should be clear that our probabilistic model assumes no data other than the identity of the user and that of the item. This makes it generally applicable to any peer-production system in which users contribute to discrete items.

Given a dataset of K independent observations $\mathcal{D} = \{(u_k, i_k, q_k) \mid k = 1, \ldots, K\}$, we infer the parameters of the model by maximizing their likelihood under $\mathcal{D}$. That is, collecting all model parameters into a single vector $\boldsymbol{\theta}$, we seek to minimize the negative log-likelihood

$$-\ell(\boldsymbol{\theta}; \mathcal{D}) = \sum_{(u,i,q)\in\mathcal{D}} \left[-q \log \mathbf{P}\left(u \succ i\right) - (1-q) \log(1 - \mathbf{P}\left(u \succ i\right))\right], \tag{2.3}$$

where $\mathbf{P}\left(u \succ i\right)$ depends on $\boldsymbol{\theta}$. In the *basic* variant, the negative log-likelihood is convex, and we can easily find a global maximum by using standard methods from convex optimization. In the *full* variant, the bilinear term breaks the convexity of the objective function, and we can no longer guarantee that we will find parameters that are global minimizers. In practice, we do not observe any convergence issues but reliably find good model parameters on all datasets.

Note that (2.3) easily generalizes from binary outcomes ($q \in \{0, 1\}$) to continuous-valued outcomes ($q \in [0, 1]$). Continuous values can be used to represent the *fraction* of the edit that is successful.

Implementation We implement the models in Python by using the TensorFlow library [1]. Our code is publicly available online at `https://github.com/lca4/interank`. In order to avoid overfitting the model to the training data, we add a small amount of ℓ_2 regularization to the negative log-likelihood. We minimize the negative log-likelihood by using stochastic gradient descent [14] with small batches of data. For INTERANK *full*, we set the number of latent dimensions to $D = 20$ by cross-validation.

Running Time Our largest experiment consists of learning the parameters of INTERANK *full* on the entire history of the French Wikipedia (c.f. Section 2.4), consisting of over 65 million edits by 5 million users on 2 million items. In this case, our TensorFlow implementation takes approximately 2 hours to converge on a single machine. In most other experiments, our implementation takes only a few minutes to converge. This demonstrates that our model effortlessly scales, even to the largest peer-production systems.

2.3.2 Applicability

Our approach models the difficulty of effecting change through the affected item's identity. As such, it applies particularly well to peer-production systems where users *cooperate* to improve the project, *i.e.*, where each edit is judged independently against an item's (latent) quality standards. This model is appropriate for a wide variety of projects, ranging from online knowledge bases (such as Wikipedia, c.f. Section 2.4) to open source software (such as the Linux kernel project, c.f. Section 2.5). In some peer-production systems, however, the contributions of different users *compete* against each other, such as multiple answers to a single question on a Q&A platform. In these cases, our model can still be applied, but fails to capture the fact that edit outcomes are interdependent.

2.4 Wikipedia

Wikipedia is a popular free online encyclopedia and arguably one of the most successful peer-production systems. In this section, we apply our models to the French and Turkish editions of Wikipedia.

2.4.1 Background & Datasets

The French Wikipedia is one of the largest Wikipedia editions. At the time of writing, it ranks in third position both in terms of number of edits and number of users[4]. In order to obtain a complementary perspective, we also study the Turkish Wikipedia, which is roughly an order of magnitude smaller. Interestingly, both the French and the Turkish editions score very highly on Wikipedia's *depth* scale, a measure of collaborative quality [196].

The Wikimedia Foundation releases periodically and publicly a database dump containing the successive revisions to all articles[5]. We use a dump that contains data starting from the beginning of the edition up to the fall of 2017: The French Wikipedia contains edits between August 4, 2001, and September 2, 20017, and the Turkish Wikipedia between December 5, 2002, and October 1, 2017.

Computation of Edit Quality

On Wikipedia, any user's edit is immediately incorporated into the encyclopedia[6]. Therefore, in order to obtain information about the quality of an edit, we have to consider the

[4] We chose the French edition over the English one because our computing infrastructure could not support the $\approx$ 15 TB needed to store the entire history of the English Wikipedia. The French edition contains roughly 5$\times$ fewer edits.

[5] See: `https://dumps.wikimedia.org/`.

[6] Except for a small minority of protected articles.

implicit signal given by subsequent edits to the same article. If the changes introduced by the edit are preserved, it signals that the edit was beneficial, whereas if the changes are reverted, the edit likely had a negative effect. A formalization of this idea is given by Adler and de Alfaro [2] and Druck et al. [48]; see also de Alfaro and Adler [44] for a concise explanation. In this work, we essentially follow their approach.

Consider a particular article and denote by v_k its k-th revision (*i.e.*, the state of the article after the k-th edit). Let $d(u, v)$ be the Levenshtein distance between two revisions [106]. We define the *quality* of edit k from the perspective of the article's state after $\ell \geq 1$ subsequent edits as

$$q_{k|\ell} = \frac{1}{2} + \frac{d(v_{k-1}, v_{k+\ell}) - d(v_k, v_{k+\ell})}{2d(v_{k-1}, v_k)}.$$

By properties of distances, $q_{k|\ell} \in [0, 1]$. Intuitively, the quantity $q_{k|\ell}$ captures the proportion of work done in edit k that remains in revision $k + \ell$. It can be understood as a *soft* measure of whether edit k has been reverted or not. We compute the unconditional quality of the edit by averaging over multiple future revisions:

$$q_k = \frac{1}{L} \sum_{\ell=1}^{L} q_{k|\ell}, \tag{2.4}$$

where L is the minimum between the number of subsequent revisions of the article and 10 (we empirically found that 10 revisions is enough to accurately assess the quality of an edit). Note that even though q_k is no longer binary, our models naturally extend to continuous-valued $q_k \in [0, 1]$ (c.f. Section 2.3.1).

In practice, we observe that edit quality is bimodal and asymmetric. Most edits have a quality close to either 0 or 1 and a majority of edits are of high quality. The two rightmost columns of Table 2.1 quantify this for the French and Turkish editions.

Dataset Preprocessing

We consider all edits to the pages in the main namespace (*i.e.*, articles), including those from anonymous contributors identified by their IP address[7]. Sequences of consecutive edits to an article by the same user are collapsed into a single edit in order to remove bias in the computation of edit quality [2]. To evaluate methods in a realistic setting, we split the data into a training set containing the first 90 % of edits, and we report results on an independent validation set containing the remaining 10 %. Note that the quality is computed based on subsequent revisions of an article: In order to guarantee that the two sets are truly independent, we make sure that we never use any revisions from the

[7]Note, however, that a large majority of edits are made by registered users (82.7 % and 76.6 % for the French and Turkish editions, respectively).

Table 2.1 – Summary statistics of Wikipedia datasets after preprocessing.

Edition	# users N	# articles M	# edits	$q < 0.2$	$q > 0.8$
French (2001–2017)	5 460 745	1 932 810	65 430 838	6.4%	72.2%
Turkish (2002–2017)	1 360 076	310 991	8 768 258	11.6%	60.5%

validation set to compute the quality of edits in the training set. A short summary of the data statistics after preprocessing is provided in Table 2.1.

2.4.2 Evaluation

In order to facilitate the comparison of our method with competing approaches, we evaluate the performance on a binary classification task consisting of predicting whether an edit is of poor quality. To this end, we assign binary labels to all edits in the validation set: the label *bad* is assigned to every edit with $q < 0.5$, and the label *good* is assigned to all edits with $q \geq 0.5$. The predictions of the classifier might help Wikipedia administrators to identify edits of low quality; these edits might then be sent to domain experts for review.

As discussed in Section 2.3, we consider two versions of our model. The first one, INTERANK *basic*, simply learns scalar user skills and article difficulties. The second one, INTERANK *full*, additionally includes a latent embedding of dimension $D = 20$ for each user and article.

Competing Approaches

To set our results in context, we compare them to those obtained with four different baselines.

Average The first approach always outputs the marginal probability of a bad edit in the training set, *i.e.*,

$$p = \frac{\#\text{ bad edits in training set}}{\#\text{ edits in training set}}$$

This is a trivial baseline, and it gives an idea of what results we should expect to achieve without any additional information on the user, article or edit.

User-Only The second approach models the outcome of an edit using only the user's identity. In short, the predictor learns skills $\{s_u \mid u = 1, \ldots, N\}$ and a global offset b such

that, for each user u, the probability

$$\mathbf{P}(u) = \frac{1}{1 + \exp[-(s_u + b)]}$$

maximizes the likelihood of that user's edits in the training set. This baseline predictor is representative of user reputation systems such as that of Adler and de Alfaro [2].

GLAD In the context of crowdsourcing, Whitehill et al. [195] propose the GLAD model that postulates that

$$\mathbf{P}(u \succ i) = \frac{1}{1 + \exp(-s_u/d_i)},$$

where $s_u \in \mathbf{R}$ and $d_i \in \mathbf{R}_{>0}$. This reflects a different assumption on the interplay between user skill and item difficulty: under their model, an item with a large difficulty value makes every user's skill more "diffuse". In order to make the comparison fair, we add a global offset parameter b to the model (similarly to INTERANK and the user-only baseline).

ORES reverted The fourth approach is a state-of-the-art classifier developed by researchers at the Wikimedia Foundation as part of Wikipedia's Objective Revision Evaluation Service [75]. We use the two classification models specifically developed for the French and Turkish editions. Both models use over 80 content-based and system-based features extracted from the user, the article and the edit to predict whether the edit will be reverted, a target which essentially matches our operational definition of *bad* edit. Features include the number of vulgar words introduced by the edit, the length of the article and of the edit, etc. This predictor is representative of specialized, domain-specific approaches to modeling edit quality.

Results

Table 2.2 presents the average log-likelihood and the area under the precision-recall curve (AUPRC) for each method. INTERANK *full* has the highest average log-likelihood of all models, meaning that its predictive probabilities are well calibrated with respect to the validation data.

Figure 2.1 presents the precision-recall curves for all methods. The analysis is qualitatively similar for both Wikipedia editions. All non-trivial predictors perform similarly in the high-recall regime, but present significant differences in the high-precision regime, on which we will focus. The ORES predictor performs the best. INTERANK comes second, reasonably close behind ORES, and the *full* variant has a small edge over the *basic* variant. GLAD is next, and the user-only baseline is far behind. This shows that (a) incorporating

Table 2.2 – Predictive performance on the *bad edit* classification task for the French and Turkish editions of Wikipedia. The best performance is highlighted in bold.

Edition	Model	Avg. log-likelihood	AUPRC
French	INTERANK *basic*	−0.339	0.399
	INTERANK *full*	**−0.336**	0.413
	Average	−0.389	0.131
	User-only	−0.346	0.313
	GLAD	−0.344	0.369
	ORES reverted	−0.469	**0.453**
Turkish	INTERANK *basic*	−0.380	0.494
	INTERANK *full*	**−0.379**	0.503
	Average	−0.461	0.168
	User-only	−0.390	0.410
	GLAD	−0.387	0.471
	ORES reverted	−0.392	**0.552**

information about the article being edited is crucial for achieving a good performance on a large portion of the precision-recall trade-off, and (*b*) modeling the outcome probability by using the *difference* between skill and difficulty (INTERANK) is better than by using the *ratio* (GLAD).

We also note that in the validation set, approximately 20 % (15 %) of edits are made by users (respectively, on articles) that are never encountered in the training set (the numbers are similar in both editions). In these cases, INTERANK reverts to average predictions, whereas content-based methods can take advantage of other features of the edit to make an informed prediction. In order to explore this *cold-start* effect in more detail, we group users and articles into bins based on the number of times they appear in the training set, and we compute the average log-likelihood of validation examples separately for each bin. Figure 2.2 presents the results for the French edition; the results for the Turkish edition are similar. Clearly, predictions for users and articles present in the training set are significantly better. In a practical deployment, several methods can help to address this issue [164, 108, 112]. A thorough investigation of ways to mitigate the cold-start problem is beyond the scope of this work.

In summary, we observe that our model, which incorporates the articles' identity, is able to bridge the gap between user-only prediction approach and a specialized predictor (ORES reverted). Furthermore, modeling the interaction between user and article (INTERANK *full*) is beneficial and helps further improve predictions, particularly in the high-precision regime.

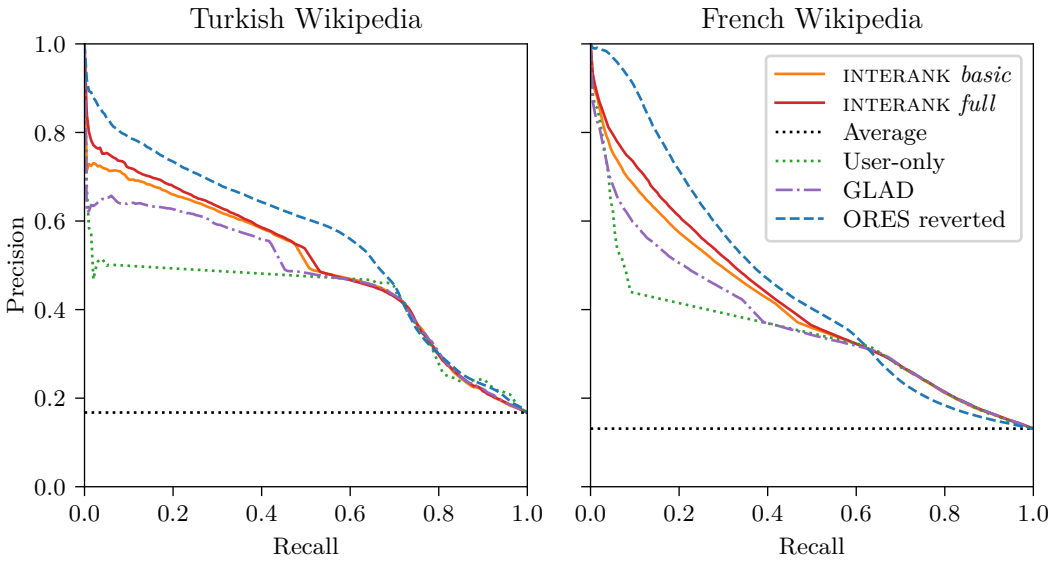


Figure 2.1 – Precision-recall curves on the *bad edit* classification task for the Turkish and French editions of Wikipedia.

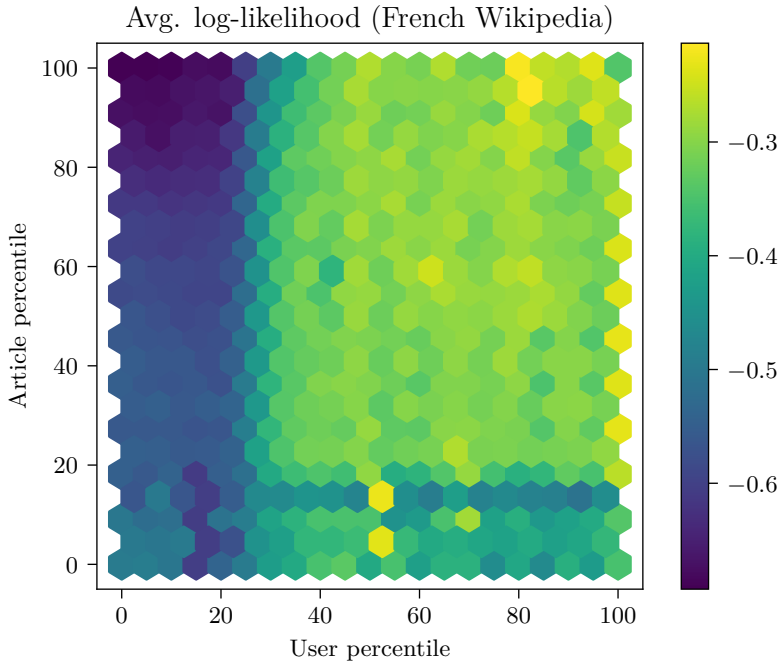


Figure 2.2 – Average log-likelihood as a function of the number of observations of the user and item in the training set of the French Wikipedia.

Table 2.3 – The ten most controversial articles on the French Wikipedia according to Yasseri et al. [203]. For each article i, we indicate the percentile of its corresponding parameter d_i.

Rank	Title	Percentile of d_i
1	Ségolène Royal	99.840 %
2	Unidentified flying object	99.229 %
3	Jehovah's Witnesses	99.709 %
4	Jesus	99.953 %
5	Sigmund Freud	97.841 %
6	September 11 attacks	99.681 %
7	Muhammad al-Durrah incident	99.806 %
8	Islamophobia	99.787 %
9	God in Christianity	99.712 %
10	Nuclear power debate	99.304 %
	Median	99.710 %

2.4.3 Interpretation of Model Parameters

The parameters of INTERANK models, in addition to being predictive of edit outcomes, are also very interpretable. In the following, we demonstrate how they can surface interesting characteristics of the peer-production system.

Controversial Articles

Intuitively, we expect an article i whose difficulty parameter d_i is large to deal with topics that are potentially controversial. We focus on the French Wikipedia and explore a list of the ten most controversial articles given by Yasseri et al. [203]. In this 2014 study, the authors identify controversial articles by using an ad-hoc methodology. Table 2.3 presents, for each article identified by Yasseri et al., the percentile of the corresponding difficulty parameter d_i learned by INTERANK *full*. We analyze these articles approximately four years later, but the model still identifies them as some of the most difficult ones. Interestingly, the article on Sigmund Freud, which has the lowest difficulty parameter of the list, has become a *featured* article since Yasseri et al.'s analysis—a distinction awarded only to the most well-written and neutral articles.

Latent Factors

Next, we turn our attention to the parameters $\{\boldsymbol{y}_i\}$. These parameters can be thought of as an embedding of the articles in a latent space of dimension $D = 20$. As we learn a model that maximizes the likelihood of edit outcomes, we expect these embeddings to capture latent article features that explain edit outcomes. In order to extract the one or

Table 2.4 – A selection of articles of the Turkish Wikipedia among the top-20 highest and lowest coordinates along the first principal axis of the matrix $\boldsymbol{Y}$.

Direction	Titles
Lowest	Harry Potter's magic list, List of programs broadcasted by Star TV, Bursaspor 2011-12 season, Kral Pop TV Top 20, Death Eater, Heroes (TV series), List of programs broadcasted by TV8, Karadayı, Show TV, List of episodes of Kurtlar Vadisi Pusu.
Highest	Seven Wonders of the World, Thomas Edison, Cell, Mustafa Kemal Atatürk, Albert Einstein, Democracy, Isaac Newton, Mehmed the Conqueror, Leonardo da Vinci, Louis Pasteur.

two directions that explain most of the variability in this latent space, we apply principal component analysis [14] to the matrix $\boldsymbol{Y} = [\boldsymbol{y}_i]$.

In Table 2.4, we consider the Turkish Wikipedia and list a subset of the 20 articles with the highest and lowest coordinates along the first principal axis of $\boldsymbol{Y}$. We observe that this axis seems to distinguish articles about popular culture from those about "high culture" or timeless topics. This discovery supports the hypothesis that users have a propensity to successfully edit *either* popular culture *or* high-culture articles on Wikipedia, but not *both*.

Finally, we consider the French Wikipedia. Once again, we apply principal component analysis to the matrix $\boldsymbol{Y}$ and keep the first two dimensions. We select the 20 articles with the highest and lowest coordinates along the first two principal axes[8]. A two-dimensional t-SNE plot [188] of the 80 articles selected using PCA is displayed in Figure 2.3. The plot enables identifying meaningful clusters of related articles, such as articles about tennis players, French municipalities, historical figures, and TV or teen culture. These articles are representative of the latent dimensions that separate editors the most: a user skilled in editing pages about ancient Greek mathematicians might be less skilled in editing pages about *anime*, and vice versa.

2.5 Linux Kernel

In this section, we apply the INTERANK model to the Linux kernel project, a well-known open-source software project. In contrast to Wikipedia, most contributors to the Linux kernel are highly skilled professionals who dedicate a significant portion of their time and efforts to the project.

[8]Interestingly, the first dimension has a very similar interpretation to that obtained on the Turkish edition: it can also be understood as separating popular culture from high culture.

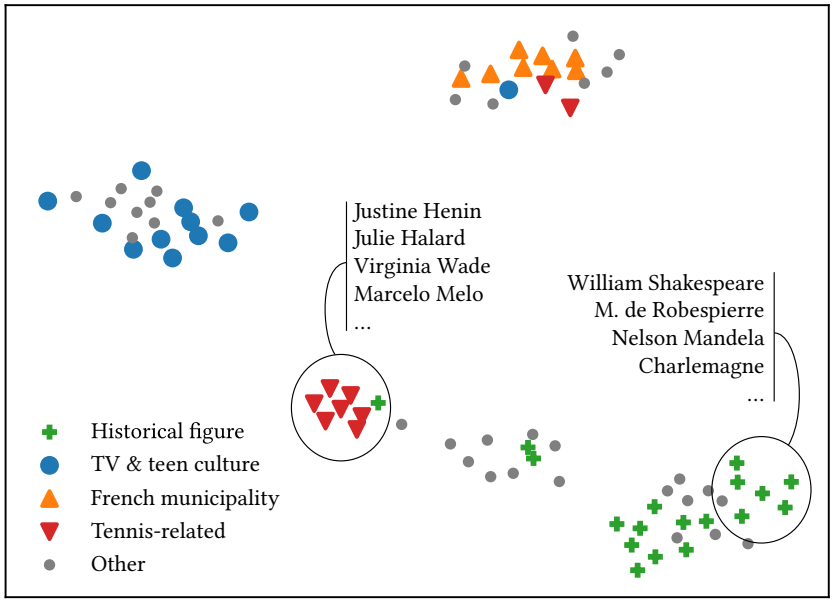


Figure 2.3 – t-SNE visualization of 80 articles of the French Wikipedia with highest and lowest coordinates along the first and second principal axes of the matrix $\boldsymbol{Y}$.

2.5.1 Background & Dataset

The Linux kernel has fundamental impact on technology as a whole. In fact, the Linux operating system runs 90% of the cloud workload and 82% of the smartphones [37]. To collectively improve the source code, developers submit bug fixes or new features in the form of a *patch* to collaborative repositories. Review and integration time depend on the project's structure, ranging from a few hours or days for Apache Server [157] to a couple of months for the Linux kernel [93]. In particular for the Linux kernel, developers submit patches to subsystem mailing lists, where they undergo several rounds of reviews. After suggestions are implemented and if the code is approved, the patch can be committed to the subsystem maintainer's software repository. Integration conflicts are spotted at this stage by other developers monitoring the maintainer's repository and any issues must be fixed by the submitter. If the maintainer is satisfied with the patch, she commits it to Linus Torvalds' repository, who decides to include it or not with the next Linux release.

Dataset Preprocessing

We use a dataset collected by Jiang et al. [93] which spans Linux development activity between 2005 and 2012. It consists of 670 533 patches described using 62 features derived from e-mails, commits to software repositories, the developers' activity and the content of the patches themselves. Jiang et al. scraped patches from the various mailing lists and matched them with commits in the main repository. In total, they managed to trace back 75% of the commits that appear in Linus Torvalds' repository to a patch submitted to a

mailing list. A patch is labeled as *accepted* ($q = 1$) if it eventually appears in a release of the Linux kernel, and *rejected* ($q = 0$) otherwise. We remove data points with empty subsystem and developer names, as well as all subsystems with no accepted patches. Finally, we chronologically order the patches according to their mailing list submission time.

After preprocessing, the dataset contains $K = 619\,419$ patches proposed by $N = 9672$ developers on $M = 394$ subsystems. 34.12% of these patches are accepted. We then split the data into training set containing the first 80% of patches and a validation set containing the remaining 20%.

Subsystem-Developer Correlation

Given the highly complex nature of the project, one could believe that developers tend to specialize in few, independent subsystems. Let $X_u = \{X_{ui}\}_{i=1}^{M}$ be the collection of binary variables X_{ui} indicating whether developer u has an accepted patch in subsystem i. We compute the sample Pearson correlation coefficient $r_{uv} = \rho(X_u, X_v)$ between X_u and X_v. We show in Figure 2.4 the correlation matrix $\boldsymbol{R} = [r_{uv}]$ between developers patching subsystems. Row $\boldsymbol{r}_u$ corresponds to developer u, and we order all rows according to the subsystem each developer u contribute to the most. We order the subsystems in decreasing order by the number of submitted patches, such that larger subsystems appear at the top of the matrix $\boldsymbol{R}$. Hence, the blocks on the diagonal roughly correspond to subsystems and their size represents the number of developers involved with the subsystem. As shown by the blocks, developers tend to specialize into one subsystem. However, as the numerous non-zero off-diagonal entries reveal, they still tend to contribute substantially to other subsystems. Finally, as highlighted by the dotted, blue square, subsystems number three to six on the diagonal form a cluster. In fact, these four subsystems (`include/linux`, `arch/x86`, `kernel` and `mm`) are core subsystems of the Linux kernel.

2.5.2 Evaluation

We consider the task of predicting whether a patch will be integrated into a release of the kernel. Similarly to Section 2.4, we use INTERANK *basic* and INTERANK *full* with $D = 20$ latent dimensions to learn the developers' skills, the subsystems' difficulty, and the interaction between them.

Competing Approaches

Three baselines that we consider—*average*, *user-only* and *GLAD*—are identical to those described in Section 2.4.2. In addition, we also compare our model to a random forest classifier trained on domain-specific features similar to the one used by Jiang et al. [93]. In

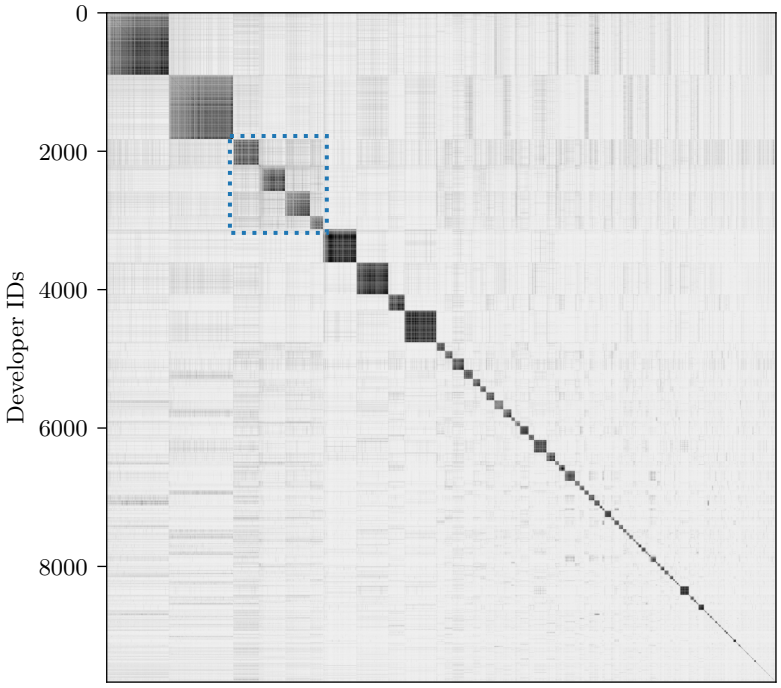


Figure 2.4 – Correlation matrix $\boldsymbol{R}$ between developers ordered according to the subsystem they contribute to the most. The blocks on the diagonal correspond to subsystems. Core subsystems form a strong cluster (blue square).

total, this classifier has access to 21 features for each patch. Features include information about the developer's experience up to the time of submission (e.g., number of accepted commits, number of patches sent), the e-mail thread (e.g., number of developers in copy of the e-mail, size of e-mail, number of e-mails in thread until the patch) and the patch itself (e.g., number of lines changed, number of files changed). We optimize the hyperparameters of the random forest using a grid-search. As the model has access to domain-specific features about each edit, it is representative of the class of specialized methods tailored to the Linux kernel peer-production system.

Results

Table 2.5 displays the average log-likelihood and area under the precision-recall curve (AUPRC). INTERANK *full* performs best in terms of both metrics. In terms of AUPRC, it outperforms the random forest classifier by 4.4%, GLAD by 5%, and the *user-only* baseline by 7.3%.

We show the precision-recall curves in Figure 2.5. Both INTERANK *full* and INTERANK *basic* perform better than the four baselines. Notably, they outperform the random forest in the high-precision regime, even though the random forest uses content-based features about developers, subsystems and patches. In the high-recall regime, the random forest

Table 2.5 – Predictive performance on the *accepted patch* classification task for the Linux kernel. The best performance is highlighted in bold.

Model	Avg. log-likelihood	AUPRC
INTERANK *basic*	-0.589	0.525
INTERANK *full*	**-0.588**	**0.527**
Average	-0.640	0.338
User-only	-0.601	0.491
GLAD	-0.598	0.502
Random forest	-0.599	0.505

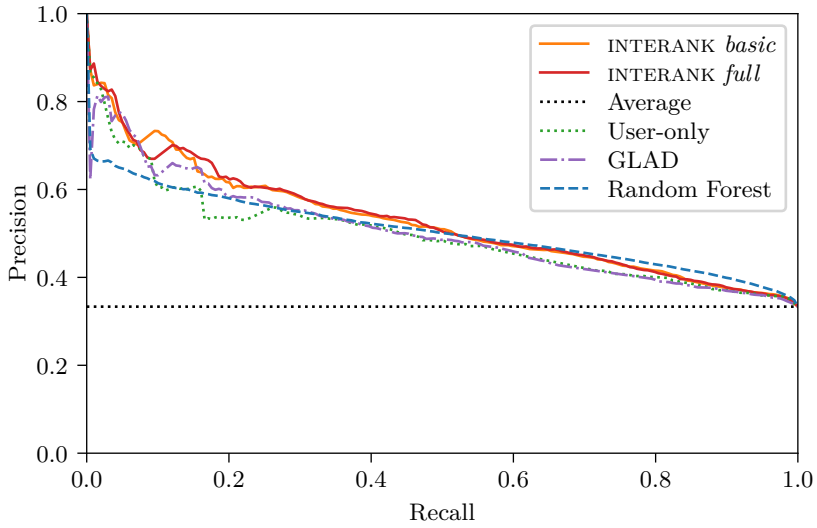


Figure 2.5 – Precision-recall curves on the bad edit classification task for the Linux kernel. INTERANK (solid orange and red) outperforms the user-only baseline (dotted green), the random forest classifier (dashed blue), and GLAD (dash-dotted purple).

attains a marginally better precision. The *user-only* and GLAD baselines perform worse than all non-trivial models.

2.5.3 Interpretation of Model Parameters

We show in Table 2.6 the top-five and bottom-five subsystems according to difficulties $\{d_i\}$ learned by INTERANK *full*. We note that even though patches submitted to difficult subsystems have in general low acceptance rate, INTERANK enables a finer ranking by taking into account *who* is contributing to the subsystems. This effect is even more noticeable with the five subsystems with smallest difficulty value.

The subsystems i with largest d_i are *core* components, whose integrity is crucial to the system. For instance, the `usr` subsystem, providing code for RAM-related instructions

Table 2.6 – Top-five and bottom-five subsystems according to their difficulty d_i.

Difficulty	Subsystem	% Acc.	# Patch	# Dev.
+2.664	`usr`	1.88%	796	70
+1.327	`include`	7.79%	398	101
+1.038	`lib`	15.99%	5642	707
+1.013	`drivers/clk`	34.34%	495	81
+0.865	`include/trace`	17.73%	547	81
-1.194	`drivers/addi-data`	78.31%	272	8
-1.080	`net/tipc`	43.11%	573	44
-0.993	`drivers/ps3`	44.26%	61	9
-0.936	`net/nfc`	73.04%	204	26
-0.796	`arch/mn10300`	45.40%	359	63

at booting time, has barely changed in the last seven years. On the other hand, the subsystems i with smallest d_i are *peripheral* components serving specific devices, such as digital signal processors or gaming consoles. These components can arguably tolerate a higher rate of bugs, and hence they evolve more frequently.

Jiang et al. [93] establish that a high prior subsystem churn (*i.e.*, high number of previous commits to a subsystem) leads to lower acceptance rate. We approximate the number of commits to a subsystem as the number of patches submitted multiplied by the subsystem's acceptance rate. The first quartile of subsystems according to their increasing difficulty, *i.e.*, the least difficult subsystems, has an average churn of 687. The third quartile, *i.e.*, the most difficult subsystems, has an average churn of 833. We verify hence that higher churn correlates with difficult subsystems. This corroborates the results obtained by Jiang et al.

As shown in Figure 2.5, if false negatives are not a priority, INTERANK will yield a substantially higher precision. In other words, if the task at hand requires that the patches classified as accepted are actually the ones integrated in a future release, then INTERANK will yield more accurate results. For instance, it would be efficient in supporting Linus Torvalds in the development of the Linux kernel by providing him with a restricted list of patches that are likely to be integrated in the next release of the Linux kernel.

2.6 Summary

In this chapter, we have introduced INTERANK, a model of edit outcomes in peer-production systems. Predictions generated by our model can be used to prioritize the work of project maintainers by identifying contributions that are of high or low quality.

Similarly to user reputation systems, INTERANK is simple, easy to interpret and applicable to a wide range of domains. Whereas user reputation systems are usually not competitive with specialized edit quality predictors tailored to a particular peer-production system, INTERANK is able to bridge the gap between the two types of approaches, and it attains a predictive performance that is competitive with the state of the art—without access to content-based features.

We have demonstrated the performance of the model on two peer-production systems exhibiting different characteristics. Beyond predictive performance, we can also use model parameters to gain insight into the system. On Wikipedia, we have shown that the model identifies controversial articles, and that latent dimensions learned by our model display interesting patterns related to cultural distinctions between articles. On the Linux kernel, we have shown that inspecting model parameters enables to identify core subsystems (large difficulty parameters) from peripheral components (small difficulty parameters).

Perspective One direction to explore is the idea of using the latent embeddings learned by our model in order to recommend items to edit. Ideally, we could match items that need to be edited with users that are most suitable for the task. For Wikipedia, an ad-hoc method called "SuggestBot" was proposed by Cosley et al. [38]. We believe it would be valuable to propose a method that is applicable to peer-production systems in general.

3 Law-Making Processes

Comparable to peer-production systems, a body of law is an example of a dynamic corpus of text documents that are jointly maintained by a group of editors who compete and collaborate in complex constellations. In this chapter[1], we develop predictive models for this process, thereby shedding light on the competitive dynamics of parliamentarians who make laws. For this purpose, we curated a rich dataset[2] of 450 000 law edits introduced by European parliamentarians over ten years. An *edit* modifies the status quo of a law, and could be in competition with another edit if it modifies the same part of that law. We adapt the INTERANK model from Chapter 2 for predicting the success of such edits, in the face of both the *inertia* of the status quo and the *competition* between overlapping edits. This model combines three different categories of features: (a) *Explicit* features extracted from data related to the edits, the parliamentarians, and the laws, (b) *latent* features that capture bi-linear interactions between parliamentarians and laws, and (c) *text* features of the edits. We show experimentally that this combination enables us to accurately predict the success of the edits. The parameters of this model can be interpreted in terms of the influence of parliamentarians and of the controversy of laws. They also help us understand what explicit and text features contribute to the acceptance of edits. The latent features cluster well into distinct topics discussed in the European Parliament.

3.1 Introduction

The process of maintaining a body of law in a democratic society shares many features with peer-production systems. The work of parliaments is governed by complex rules, processes, and conventions, in order to foster compromises among competing viewpoints and priorities. How well this process works, to what extent it is subject to biases and to benign or undue influences is of obvious concern to citizens and to scientists alike. An exciting recent development in this regard is the adoption of *open government initiatives*,

[1] This chapter is based on Kristof et al. [104, 105].

[2] Data and code publicly available on https://github.com/indy-lab/war-of-words-2.

such as in the United States [88], Switzerland [69], Brazil [47], and the European Union [187]. Open-government data published on the Web are of great interest to citizens, companies, sub- and supra-government entities, and researchers. These initiatives aim to improve the transparency of the law-making process and the accountability of its protagonists.

Not surprisingly, the dynamics of this process is complex, given the confluence of many stakeholders, topics, special interests, and lobbying groups. Until open-government was introduced, the work of parliaments had not been systematically accessible to the general public, and internal documents – when they existed – were difficult to find. The European Union (EU), however, has been a pioneer in opening the mechanics of its parliament. It publishes detailed records of the process by which bills are written and amended, until they finally become law. Once an initial draft of a new law has been published, parliamentarians (MEPs, for Members of the European Parliament) in one or several specialized committees examine the draft and propose amendments. Several amendments can be in conflict if they attempt to modify the same part of the law draft. To be instituted, an amendment needs to be approved by the committee in charge, and ultimately by the full plenary. The European Parliament publishes every proposed amendment and its authorship, along with various other details. This makes it possible to build detailed models of the interplay between MEPs, laws, amendments, and committees.

In this work, we (i) curate a large-scale dataset of amendments proposed by MEPs over two legislature periods (2009–2019) and (ii) develop a predictive model for the success and failure of proposed amendments. Specifically, we collect explicit features for each MEP, including their party membership, country of origin, and gender. We also collect explicit features of the amendments and dossiers (law drafts), including their type and the committee in charge. Finally, we extract the actual text of the amendments, which consists of *edits* of the proposed law. Our dataset contains 449 493 edits proposed by 1 214 parliamentarians on 1 889 dossiers

Our model relies mostly on the structure of incompatible edits, which can be viewed as a *conflict graph* among all edits that target the same law. We posit a measure of *strength* for each parliamentarian, and an edit inherits the strengths of its supporters. There are two sources of competition in the process. First, a proposed edit competes with the status quo, because the edit can be rejected in favor of not changing the existing state of a law. Our model incorporates this by endowing each law with a measure of *inertia* that represents the level of controversy of a law. Second, proposed edits of a law are frequently mutually exclusive, because they overlap and are incompatible. These edits then compete against each other, as well as against the status quo.

We further include explicit features and text features into the model. This combination gives rise to models with improved predictive performance and enables us to make predictions for unseen laws. We also endow our model with a set of latent features

for both laws and MEPs, which capture richer interactions between them. Indeed, it would seem plausible that an MEP might be an expert in one subject matter, but less knowledgeable in another, which would bear upon their effectiveness in promoting a particular amendment.

The remainder of this chapter is structured as follows. We set the framework by giving some background on the European legislative process in Section 3.2. We state the problem and provide a detailed description of our dataset in Section 3.3. In Section 3.4, we use our dataset to describe the evolution of a law via a graph-theoretical viewpoint. We describe our statistical models in Section 3.5. We give the results and interpretations of our experiments in Section 3.6. We describe related work in Section 3.7 and conclude in Section 3.8.

3.2 The European Law-Making Process

3.2.1 Representative Democracies

In representative democracies, citizens elect politicians to represent them in the various branches of the government. The executive branch is in charge of executing and enforcing the laws. Representatives of the executive branch can also propose new laws, but, to avoid a concentration of power, they cannot pass new legislation without the approval of the legislative branch. The legislative branch, typically a parliament, represents both the people and the sub-governmental entities (such as states and municipalities). Parliamentarians can propose new legislation or amend propositions made by the executive branch. Finally, the judicial branch balances the power of the executive branch and the legislative branch through its ability to decide whether the laws are constitutional.

Here, we focus on the European Union (EU). The EU is a political and economic union of 28 countries called member states. This union enables them to share their markets, to ease mobility across borders, to favor economic development, and to harmonize laws. The EU covers an estimated population of 513 million, and up to 84% of member states' national laws emanate from the EU [130]. Hence, EU laws have a significant impact on the life of many people. European institutions make efforts to be transparent. They make a lot of valuable data available online: parliamentary amendments, meetings by the commissioners with civil society, and a transparency register to monitor interest groups.

The EU political system is broadly similar to that of a regular state. The 751 parliament representatives (MEPs, for Member of the European Parliament) are elected every five years by universal suffrage. The executive branch is called the *European Commission*. The legislative branch consists of the *European Parliament* and of the *Council of Ministers*. The Parliament is divided into 20 committees, comprising sub-sets of MEPs and specialized in some particular policy area (such as fisheries, judiciary affairs, transportation, and

trade). Each MEP is a member of at least one committee. The myriad of national parties aggregate into a small number of political groups.

3.2.2 The Ordinary Legislative Procedure

We now describe the EU law-making process in some detail, leading up to our modeling assumptions. Under the Treaty of Lisbon [141], which marks the beginning of the 7^{th} legislature in 2009, the Parliament's powers were increased. The Parliament became central in the process through which new laws are created. This process can take the form of various procedures, the main one being the *ordinary legislative procedure* (OLP) [144]. Through the OLP, the Commission initiates a legislative proposal, and the Parliament must adopt it in order for the proposal to become a law. Other procedures exist, where the Parliament is not necessarily involved. Since 2009, the Parliament has dealt with 90% of all new laws via the OLP. In this regard, we focus on the dynamics of the legislative process in the Parliament. A sketch of the OLP is illustrated in Figure 3.1 and described in the next paragraphs.

To create a new law, (A) the Commission drafts a legislative *proposal* and transfers it to the corresponding committee of the Parliament. For instance, if the proposal introduces regulations on greenhouse-gas emissions, it is transferred to the Environment Committee. The committee appoints a *rapporteur* to lead the debate. The role of the committee is to write a *report* in the form of *amendments* to the proposal, *i.e.*, insertions in or deletions of parts of the proposal. The rapporteur first seeks external expertise to draft a report. Then, (B) other MEPs on the committee can in turn propose amendments to the proposal. To constitute the final report to be submitted to the whole Parliament, each amendment by the rapporteur or by other MEPs is therefore voted on within the committee. Once the committee finds a consensus, (C) they transfer the report to the whole Parliament.

In the plenary session, the Parliament holds a vote on the report. (D) If rejected, the proposal is abandoned; (E) if accepted, the report, establishing the Parliament's position on the proposal, is transferred to the Council of Ministers. The report is therefore an important document and the rapporteur has an important role to play. The ministers (of the different EU countries) can accept the report, (F) in which case, the proposal is adopted with the Parliament's amendments and a new law is created; or they can make amendments, (G) in which case it is transferred back to the parliamentary committee. At this stage, we say that a law has gone through the first *reading*.

Other committees can also independently decide to address an *opinion* to the reporting committee. For instance, the Transportation Committee might consider that it is also concerned by greenhouse-gas emissions and that it is entitled to give its opinion to the Environment Committee. An opinion is similar to a report in that it contains amendments

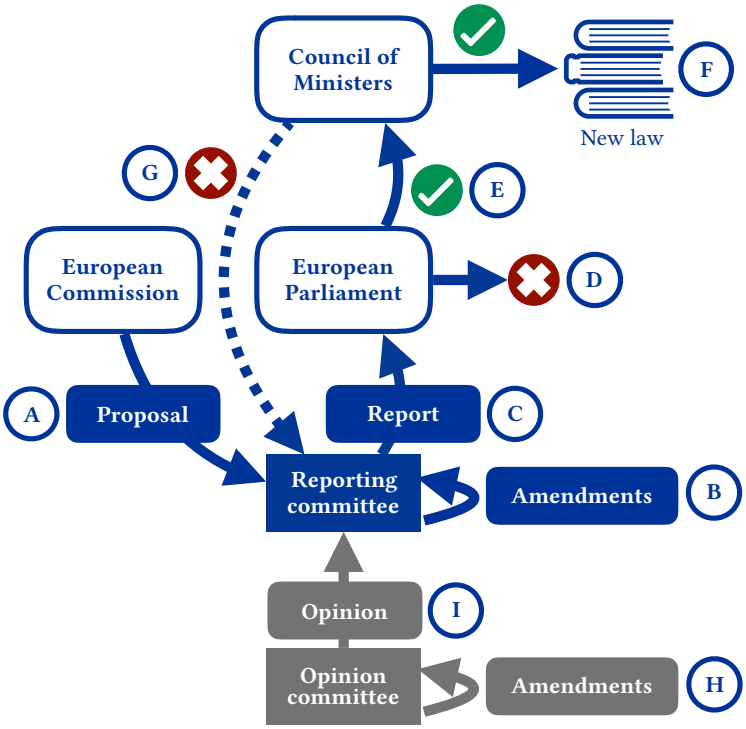


Figure 3.1 – Sketch of the *ordinary legislative procedure*. (A) The Commission submits a legislative proposal to one of the Parliament committees. (B) The proposal is amended and (C) submitted to vote to the whole Parliament. (D) If it is rejected, the proposal is abandoned. (E) If it is accepted, it is transferred to the Council. (F) If the Council accepts the amended proposal, a new law is adopted. (G) If the Council amends it, it is sent back to the committee. (H) Other committees can optionally make amendments and (I) suggest them to the reporting committee.

to the proposal. It is created similarly to a report, *i.e.*, (H) the opinion committee appoints a rapporteur to draft an opinion, and other MEPs can propose amendments. (I) The opinion committee then transfers its opinion to the reporting committee. An opinion differs from a report in that it is not voted by the whole Parliament (only the report is), and the reporting committee is free to take into account the amendments from the opinion. Amendments from the opinion committee can, however, be in conflict with amendments from the reporting committee, and MEPs from the reporting committee will also have to vote on those. We refer to reports and opinions as *dossiers*.

This iterative process can be repeated up to three times (three readings). The third reading, called conciliation, involves a negotiation between the Parliament and the Council. During the 8$^{\text{th}}$ legislature for example, 99% of all laws were adopted after the first reading, *i.e.*, after amendments made by both the Parliament and the Council, and 89% were adopted directly after amendments by the Parliament, *i.e.*, the Council accepted it without making amendments.

3.3 Dataset

3.3.1 Amendments & Edits

We collected a dataset of 237 177 legislative amendments from the European Parliament website.[3] The dataset spans the 7$^{\text{th}}$ legislature (referred to as EP7), from 2009 to 2014, and the 8$^{\text{th}}$ legislature (EP8), from 2014 to 2019. MEPs come from 28 different countries, and they belong to one of the 8 (EP7) or 9 (EP8) political groups. An amendment consists of (i) one or several authors, (ii) the original text by the European Commission, and (iii) the amended text by the author(s). We show an example of two amendments in their raw format in Figure 3.2. The two amendments are proposed on Article 13 of a proposal about copyrights on the Internet. Amendment 802 is proposed by three MEPs and consists of three edits: (a) Inserting "copyright" (in green), (b) replacing "by" by "uploaded by users of" (in yellow), and (c) deleting the end of the title after "providers" (in red). Amendment 803 is proposed by two other MEPs and consists of two edits: (d) Replacing "large" by "significant" (in yellow) and (e) inserting "copyright protected" (in green). There are two conflicts in this amendment: Edit (c) of the first amendment is in conflict with Edit (d), and it is also in conflict with Edit (e). All these edits are also implicitly in conflict with the original text proposed by the European Commission. Out of these five edits, only Edit (d) was accepted. All other edits were rejected, *i.e.*, the status quo was voted and the text proposed by the Commission was maintained.

Edits MEPs propose amendments on a specific article of the legislation, and they can modify several parts within a single amendment. As a result, we decompose the difference

[3]Data and code publicly available on https://github.com/indy-lab/war-of-words.

Amendment 802
Lidia Joanna Geringer de Oedenberg, Catherine Stihler, Victor Negrescu
Article 13 – title

Text proposed by the Commission	*Amendment*
Use of protected content ***by*** information society service providers ***storing and giving access to large amounts of works and other subject-matter uploaded by their users***	Use of ***copyright*** protected content ***uploaded by users of*** information society service providers

Amendment 803
Tadeusz Zwiefka, Bogdan Brunon Wenta
Article 13 – title

Text proposed by the Commission	*Amendment*
Use of protected content by information society service providers storing and giving access to ***large*** amounts of works and other subject-matter uploaded by their users	Use of protected content by information society service providers storing and giving access to ***significant*** amounts of ***copyright protected*** works and other subject-matter uploaded by their users

Figure 3.2 – Example of two conflicting amendments in their raw format on the title of Article 13 of a proposal about copyrights on the Internet. (Top) Amendment 802 is proposed by three MEPs and consists of three edits. (Bottom) Amendment 803 is proposed by two other MEPs on the same text, and it consists of two edits. The last edit of Amendment 802 (deleting the end of the title) conflicts with both edits of Amendment 803. Only the first edit of Amendment 803 (replacing "large" by "significant") was accepted, and all other edits were rejected.

between the original and the amended text into one or several *edits*, as defined below. An edit is a sequence of words that are inserted or deleted or both. We extract edits by computing the *diff*, *i.e.*, the difference between the words in two texts, between the original and the amended text of each amendment. We normalize the texts by removing special characters and by putting the words in lower case. We keep punctuation because the structure of sentences is important in legal texts. We merge identical edits proposed by different MEPs, thus considering them as one edit proposed by all authors together. This is in line with Rule 174 of the Rules of Procedure of the Parliament [145]. We extract 200 407 edits for EP7 and 249 086 edits for EP8. On average, there are 1.85 and 1.93 edits per amendment for EP7 and EP8, respectively. There are also more dossiers in EP7 than in EP8, which means that there are proportionally more edits per dossier in EP8.

Conflicts There exists an inherent competition between the MEPs in the amending process, as amendments are vehicles of political ideas and interests. We are therefore interested in the conflicts between edits. We define a *conflict* as a set of edits that overlap. Edits overlap because they modify parts of the text at the same position. We extract 40 302 conflicts for EP7 and 56 298 for EP8. Adding the conflicts to isolated edits, we obtain a dataset of 126 417 data points for EP7 and 141 034 data points for EP8.

Labels The votes on each edit are not publicly available, and we need to infer their outcomes from the raw data. Reports and opinions contain only the amendments accepted within the committees. Draft reports, draft opinions, and other documents containing all proposed amendments are published separately. Therefore, if the edits extracted from the latter documents appear in the former documents, we label them as *accepted*, *i.e.*, the committee votes to include these edits in their report or opinion. Otherwise, we label them as *rejected*. Out of the proposed edits, 37.7% are accepted for EP7 and 25.7% for EP8.

Timestamps The timeline of the legislative process described in Section 3.2 varies from one dossier to another. Depending on the dossier, MEPs can propose edits during a window of one to six months, after which all the edits related to that dossier are published together. As a result, the actual, detailed chronology of the edits is unfortunately hidden, and we do not have access to the precise time the edits are proposed and when they are voted. Furthermore, there is a delay between the time an edit is proposed and the time it is voted: recent edits might be voted *before* older ones. The timestamps associated with each edit are, therefore, noisy.

Table 3.1 – Descriptive statistics of our extended dataset.

	EP7 (2009–2014)	EP8 (2014–2019)
# amendments	108 292	128 885
# edits	200 407	249 086
# conflicts	126 417	141 034
# MEPs	761	791
# dossiers	1 089	800
% accepted	37.7%	25.7%
% inserted	37.8%	37.9%
% deleted	22.0%	22.4%
% replaced	40.2%	39.7%

In total, we collect 449 493 edits from 237 177 amendments in the European Parliament during the 7$^{\text{th}}$ and the 8$^{\text{th}}$ legislature periods[4] (referred to as EP7 and EP8), between 2009 and 2019 (each period lasts 5 years). After gathering the edits according to the conflicts, we obtain 267 451 conflicts for both EP7 and EP8, covering 1889 dossiers. We summarize this dataset in Table 3.1.

3.3.2 Explicit Features

We extract explicit (meta) features of the MEPs, the edits, and the dossiers, as well as text features. For each MEP, we collect their nationality (one of 28), their EU political group (one of 8 or 9), and their gender. A political group clusters national parties that share similar political ideologies. For each edit, we identify whether it is an insertion, a deletion, or a replacement of some words in the proposal, and we compute its length. We also collect information about where in the law the edit was proposed: in an article (in the body of the proposal), in a recital (in the preamble of the proposal), in an annex, or in other more specific but less frequent parts of a law. We determine whether an edit in a reporting committee comes from an opinion committee (in which case it is an "outsider"). Finally, we note whether an edit comes with an optional justification. For each dossier, we identify its type (report or opinion) and the committee that is in charge. We also note if the proposal is a regulation (legally binding for all member states of the EU), a directive (sets general goals that member states can implement however they want), or a decision (binding to one member state or company only). We describe these explicit features in Table 3.2.

[4]We do not collect data from EP9 (2019 – 2023), as the amount of published data is too small at this time: The legislature period started in Fall 2019 and the Parliament's activities were slowed down due to the COVID-19 crisis in Spring 2020.

Table 3.2 – List of features for MEPs and edits.

Category	Feature	Type [Values]
MEP	Nationality	Categorical [28]
	Political group	Categorical [8 or 9]
	Gender	Categorical [2]
	Rapporteur	Binary
Edit	Edit type	Categorical [3]
	Log-length (+)	Numerical [$\mathbf{R}_{\geq 0}$]
	Log-length (-)	Numerical [$\mathbf{R}_{\geq 0}$]
	Article type	Categorical [7]
	Outsider committee	Binary
	Justification	Binary
Dossier	Type	Categorical [2]
	Committee	Categorical [35]
	Legal act	Categorical [3]

3.3.3 Text Features

We further augment the dataset by collecting text features of the edit itself. It is reasonable to expect that certain words and phrases are predictive of the success of an edit. We extract the deleted words w_- from the proposal and the inserted words w_+ from the amendment. In Figure 3.2, for example, Edit (b) of Amendment 802 has w_- = *"by"* and w_+ = *"uploaded by users of"*. We also consider the context of an edit by extracting the original text of the whole amended article. For Amendment 802, the context is the portion of text labelled as *"Text proposed by the Commission"*. Finally, we also extract the title of the law proposal; we will use it as a text feature of the dossier. For Amendments 802 and 803, the title is *"Copyright in the Digital Single Market"*. We map all words to lower case, and we replace digits in the title by the letter "D", as there are many reference numbers that are unlikely to be useful for our task.

We give some statistics of the distribution of the length of the deleted text w_-, the inserted text w_+, the context, and the title in Table 3.3. We report the lower quartile Q_1 and the upper quartile Q_3, as well as the median. About half of the inserted and deleted texts are short (7 words or less), but the distribution of lengths has a long tail, as shown by the larger values of the upper quartile Q_3. The context provides large portions of text (the median is at 42 for EP7 and 49 for EP8), which will be useful for making predictions. In Section 3.5, we describe how we incorporate the explicit features and the text features into our models.

Table 3.3 – Distribution of text lengths in number of words.

Legislature	Type	Q_1	Median	Q_3
EP7	Insertion w_+	2	7	20
	Deletion w_-	2	6	26
	Context	15	42	79
	Title	6	12	19
EP8	Insertion w_+	2	6	17
	Deletion w_-	2	6	28
	Context	20	49	93
	Title	6	10	22

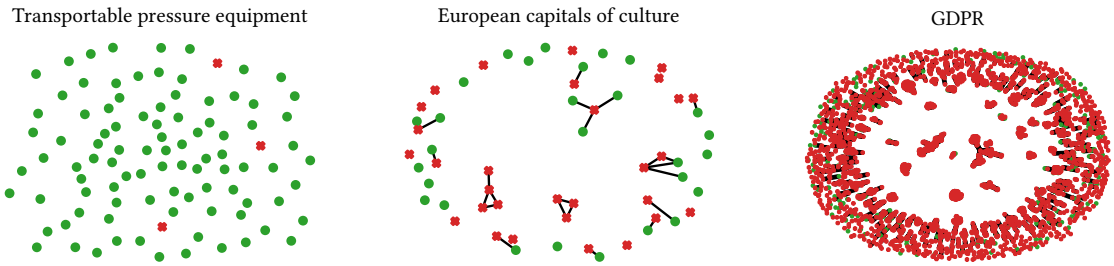


Figure 3.3 – (Left) The "transportable pressure equipment" edit graph contains 96 edits (97% accepted) and no conflicts. (Center) The "European capitals of culture" edit graph contains 58 edits (48% accepted) and 16 conflicts. (Right) The GDPR edit graph contains 3154 edits (9% accepted) and 1298 conflicts.

3.4 Edit Graph

We describe the dynamics of the legislative process in terms of the conflicts between edits. For each dossier, we construct the edit graph $G = (V_G, E_G)$, such that each node $v \in V_G$ is an edit and such that there is an undirected edge $(u, v) \in E_G$ if edits u and v overlap. A component of size at least 2 in G is therefore a group of overlapping edits. An isolated node corresponds to an edit that does not overlap with any other edit.

In Figure 3.3, we show the edit graphs of three regulations of EP7. We depict each node with a green dot if the edit is accepted, and with a red cross if the edit is rejected. The "transportable pressure equipment" (left), a very specific legislation, exhibits a graph with 96 nodes, among which 97% are accepted. The graph contains only isolated nodes, meaning that no edits overlap: all its components are size 1. The "European capitals of culture" (center), which can affect some cities of member states, exhibits a graph with 58 nodes, among which 48% are accepted. The graph contains 16 cliques and the average component size is 1.49. The GDPR (right), with high stakes for both businesses and consumers, exhibits a graph with 3154 nodes, among which only 9% are accepted. The graph contains 1298 cliques, meaning that many edits are conflicting, and has an average component size of 3.44.

3.4.1 Conflicts

Conflicts are inherent in the ordinary legislative procedure defined in Section 3.2, as every proposed edit reflects a disagreement with the initial law proposal. A first class of conflicts occur between the proposal and each edit proposed by MEPs. These conflicts appear as components of any size in G. Hence, every isolated node and every clique in G are such conflicts. We call them "conflicts with the status quo", as they are in disagreement with the proposal. For example, each edit of Amendments 108 and 5 in Figure 3.2 is such a conflict. In Figure 3.3 (left), each green node is an edit accepted over the status quo, and each red node is an edit rejected over the status quo. Similarly, in Figure 3.3 (center), the cliques with all red nodes are rejected over the status quo.

Another class of conflicts occur between two or more edits proposed by MEPs. If several MEPs propose different edits on the same part of a text, they compete with each other for the acceptance of their suggestions. In this case, the edits conflict with the status quo *and* with edits proposed by other MEPs. These conflicts appear as a clique of size at least 2 in G, as there is an edge between overlapping edits. For example, in Figure 3.2, the first edit in Amendment 108 and the first edit in Amendment 5 form such a conflict. It corresponds to a clique of size 2. In Figure 3.3 (left), there are no such conflicts. As no edge links any two nodes, all conflicts are only with the status quo. In Figure 3.3 (center), however, the cliques with one green node and one or more red nodes are conflicts between several edits, where one edit is accepted over the others and over the status quo.

In G, two green nodes cannot appear at both ends of the same edge, as only one edit can be accepted among those that are conflicting. Hence, green nodes can only appear as an independent set on the components. Two red nodes, however, can appear at both ends of the same edge, as they can both be rejected: this is the case with the first edit in Amendments 108 and 5.

Conflicts between edits can be easily projected to conflicts between MEPs, as we know the authors of each edit. We compare the conflictive dynamics between MEPs by comparing the distribution of (i) the number of cliques and (ii) the size of cliques in the edit graph G of each dossier. The median number of cliques in EP7 is 14, which is smaller than 32 in EP8. The median size of cliques in EP7 is 2.23, which is smaller than 2.38 in EP8. There are therefore (i) more conflicts and (ii) conflicts of larger size in EP8, compared to EP7. This increased heterogeneity in the clique structures of edit graph G suggests that predicting the outcome of edits is more difficult for EP8.

3.4.2 Collaboration

Here, we construct the collaboration graph $H = (V_H, E_H)$ by projecting edits onto the space of MEPs. Each node $v \in V_H$ is a MEP and there is a weighted edge $(u, v) \in E_H$ if MEPs u and v co-sign an edit, where the weights count the collaborations. The node-

degree distribution, *i.e.*, the distribution of number of collaborators, is well fitted by a power-law distribution whose median is 61 for EP7 and 136 for EP8. Hence, MEPs tend to collaborate with many colleagues in general, and more so in EP8.

We quantify (i) national and (ii) political collaborations by computing the modularity [136] in graph H when defining communities by nationality or by political group. Modularity is a measure of the strength of the community structure in a graph. It takes values between -1 and 1, with a higher positive value indicating stronger community structure. In order to obtain comparable measurements, we merge the two right-wing populist, euroskeptic groups of EP8 to obtain 8 political groups, as in EP7[5]. We compute the modularity $Q_n^{(l)}$ when clustering MEPs by nationality in the l-th legislature and $Q_p^{(l)}$ when clustering MEPs by political group. Computing the modularities in both legislatures, we obtain

$$Q_n^{(7)} = 0.17 > 0.05 = Q_n^{(8)},$$
$$Q_p^{(7)} = 0.22 > 0.18 = Q_p^{(8)}.$$

This suggests that political affinity is more important than national affinity to drive collaboration in EP8 compared to EP7. The political science has not settled on this point: political cohesion is stronger than national cohesion in the EU Parliament in some works [84, 86, 125], and national cohesion is stronger than political cohesion in other works [34, 85, 27]. To the best of our knowledge, however, all previous work about political and national cohesion is performed using vote outcome data rather than amendment outcome, an inherently different setting.

3.5 Statistical models

3.5.1 Problem Statement

We build a model that predicts the vote outcome of edits that will form the reports and the opinions. Formally, we take a supervised approach to solve the following prediction problem: Let $\mathcal{C} = \{a, b, \ldots\}$ be a set of conflictive edits proposed on a dossier i, for which we have observed other edits. Note that $\mathcal{C}$ forms a clique in the edit graph G of Section 3.4. We want to predict which of the conflictive edits in $\mathcal{C}$ or the status quo of the proposal for dossier i will be accepted within the committee. This task differs from multinomial classification as the number of classes varies for each data point: If an edit a is in conflict only with the original text proposed by the Commission, then $|\mathcal{C}| = 1$. If several edits $a, b, \ldots \in \mathcal{C}$ are in conflict against each other, then $|\mathcal{C}| > 1$.

According to Rule 180 of the Rules of Procedure of the European Parliament [147], the committee sets a deadline by which MEPs must propose amendments to a dossier. The

[5]Communities of equal size are required to enable fair comparison of modularities. One right-wing populist group in EP7 split into two at the beginning of EP8.

voting takes place after this time. Hence, at the time of voting, an edit is expected to confront all alternatives: If edits a, b, and c are in conflict, the MEPs vote on all three of them and the status quo to select only one outcome.

3.5.2 The War of Words Model

We propose a statistical model of edit outcomes from conflicts. We incorporate assumptions reminiscent of the Bradley-Terry model [19] and of the Rasch model [151], as follows. We model the amending process as a "game" between (a) the MEPs themselves (similar to the Bradley-Terry model) and (b) the MEPs and the status quo (similar to the Rasch model). For simplicity, let us suppose that an edit proposed by MEP u is accepted on dossier i over a conflicting edit proposed by MEP v. As an example, a MEP from one party might propose a modification favoring economic interests, whereas another MEP from another party proposes a modification at the same position in the proposal favoring social interests. We model the probability of the edit proposed by MEP u to be accepted over the edit proposed by MEP v on dossier i, *i.e.*, the probability of MEP u "winning" over MEP v on dossier i as

$$\begin{aligned}\mathbf{P}\left(u \succ_i v\right) &\coloneqq \frac{\exp(s_u)}{\exp(s_u) + \exp(s_v) + \exp(d_i + b)} \\ &= \frac{1}{1 + \exp[-(s_u - s_v)] + \exp[-(s_u - d_i) + b]},\end{aligned} \tag{3.1}$$

where $s_u, s_v \in \mathbf{R}$ are the *skills* of MEPs u and v, $d_i \in \mathbf{R}$ is the *inertia* of dossier i, and $b \in \mathbf{R}$ is a global bias parameter. The first exponential in the denominator of (3.1) encodes the MEP-MEP interaction. The second exponential encodes the MEP-dossier interaction. If an edit proposed by MEP u does not conflict with any other edits, the MEP-MEP term vanishes, leaving only the MEP-dossier term.

As explained in Section 3.3 and Section 3.4, one or more MEPs can propose an edit, and an edit can be in conflict with one or more other edits. It is easy to generalize (3.1) to multiple authors and multiple conflicts. To model multiple authors, we simply sum the skills of each author of an edit. To model multiple conflicts, we observe that each conflict generates a new MEP-MEP interaction term. Call $\mathcal{C} = \{a, b, \ldots\}$ the set of conflicting edits proposed by authors $\mathcal{A}_a, \mathcal{A}_b, \ldots$. The probability of edit a being accepted over edits $b, \ldots$ on dossier i is given by

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) \coloneqq \frac{\exp(s_a)}{\sum\limits_{c \in \mathcal{C}} \exp(s_c) + \exp(d_i + b)}, \tag{3.2}$$

where $s_a = \sum_{u \in \mathcal{A}_a} s_u$ is the cumulated skill of all authors of edit a. We refer to this model as the War of Words model, or simply as the WoW model. The probability

that all edits are rejected, *i.e.*, the status quo of dossier i wins, is given by

$$\mathbf{P}\left(i \succ \mathcal{C}\right) \coloneqq 1 - \sum_{a \in \mathcal{C}} \mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \frac{\exp(d_i + b)}{\sum\limits_{a \in \mathcal{C}} \exp(s_a) + \exp(d_i + b)}.$$

The parameters in this model enable interpretation. The skill s_u quantifies the ability of MEP u to pass an edit representing their views. We interpret a high skill as a high *influence*. The inertia d_i quantifies the resistance to change of dossier i. This resistance is not due to the dossier resisting *per se* but rather to the effect of other MEPs voting the edits or proposing conflicting edits. In this sense, we interpret a high inertia as a sign of possible high *controversy*. The general bias term b tunes the importance that the model gives to the MEP-MEP term relative to the MEP-dossier term. We conduct an in-depth analysis of the parameters in Section 3.6.

3.5.3 Enriched Models

Explicit Features We extend the WoW model by augmenting it with explicit features of the MEPs (*e.g.*, nationality), the edits (*e.g.*, length of inserted text), and the dossiers (*e.g.*, report or opinion), as described in Table 3.2. From (3.2), we replace the skill parameters s_a with the inner product between a feature vector $\boldsymbol{s}_a \in \mathbf{R}^{M_E}$ of M_E features of edit a and the associated parameter vector $\boldsymbol{w}_E \in \mathbf{R}^{M_E}$. We also replace the difficulty parameter d_i by the product of a feature vector $\boldsymbol{d}_i \in \mathbf{R}^{M_D}$ of M_D features of dossier i and its associated parameter vector $\boldsymbol{w}_D \in \mathbf{R}^{M_D}$. We then have

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \frac{\exp(\boldsymbol{s}_a^\intercal \boldsymbol{w}_E)}{\sum\limits_{c \in \mathcal{C}} \exp(\boldsymbol{s}_c^\intercal \boldsymbol{w}_E) + \exp(\boldsymbol{d}_i^\intercal \boldsymbol{w}_D + b)}. \tag{3.3}$$

We refer to this model as WoW(*Explicit*) (or WoW(X), for conciseness). In (3.2), the feature vector $\boldsymbol{s}_a$ is the indicator of the authors of an edit a: Its entries s_u are 1 for all $u \in \mathcal{A}_a$ and 0 otherwise. Similarly, the feature vector $\boldsymbol{d}_i$ is the indicator of dossier i. In (3.3), the feature vectors $\boldsymbol{s}_a$ and $\boldsymbol{d}_i$ represent features related to MEPs, edits, and dossiers derived from our dataset.

Latent Features Consider the simple case of an MEP u proposing an edit on dossier i, and suppose that this edit conflicts with another edit, proposed by MEP v. From (3.2), let $p(u \succ_i v)$ be the probability that, for dossier i, the edit proposed by MEP u is accepted over the edit proposed by MEP v. The assumption made in the WoW model is strong: It posits that if MEP u is more influential than MEP v, then, all other things being equal, $\mathbf{P}\left(u \succ_i v\right) > \mathbf{P}\left(v \succ_i u\right)$ for all dossiers i. This assumption is not always realistic: Dossiers span a vast amount of different topics, and the MEPs have their own

specializations and interests. For example, an MEP familiar with fisheries might not be knowledgeable about research and academia.

In order to capture these dependencies, we incorporate a bi-linear term into the WOW model. We assign a vector $\boldsymbol{x}_u \in \mathbf{R}^L$ to each MEP u, and a vector $\boldsymbol{y}_i \in \mathbf{R}^L$ to each dossier i, for some dimensionality $L > 0$. We then rewrite (3.2) as

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \frac{\exp(s_a + \boldsymbol{x}_a^\intercal \boldsymbol{y}_i)}{\sum\limits_{c \in \mathcal{C}} \exp(s_c + \boldsymbol{x}_c^\intercal \boldsymbol{y}_i) + \exp(d_i + b)}, \tag{3.4}$$

where $\boldsymbol{x}_a = \sum_{u \in \mathcal{A}_a} \boldsymbol{x}_u$ is the sum of the latent features $\boldsymbol{x}_u$ of each author u of edit a. We refer to this model as the WOW(*Latent*) model (or WOW(L)). The latent vectors $\boldsymbol{x}_u$ and $\boldsymbol{y}_i$ can be viewed as the embeddings of MEP u and of dossier i in a Euclidean latent space. Informally, the probability $\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right)$ increases when the MEP embedding $\boldsymbol{x}_a$ is co-linear with the dossier embedding $\boldsymbol{y}_i$ in the latent space. It decreases when the two vectors point in opposite directions. Furthermore, vector $\boldsymbol{x}_u$ can be interpreted as the set of skills of MEP u. Similarly, $\boldsymbol{y}_i$ can be interpreted as the set of skills required to edit dossier i.

Text Features The features described so far ignore the text content of the edit itself. It is reasonable to expect that the presence of certain words or phrases in the original or amended text of an edit, and in the title of the dossier, are predictive of the success of the edit. Hence, we incorporate text features to the WOW model by rewriting (3.2) as

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \frac{\exp(s_a + \boldsymbol{r}_a^\intercal \boldsymbol{w}_T)}{\sum\limits_{c \in \mathcal{C}} \exp(s_c + \boldsymbol{r}_c^\intercal \boldsymbol{w}_T) + \exp(d_i + \boldsymbol{r}_i^\intercal \boldsymbol{w}_{T'} + b)}, \tag{3.5}$$

where $\boldsymbol{r}_a \in \mathbf{R}^D$, $\boldsymbol{r}_i \in \mathbf{R}^{D'}$ are, respectively, representations of the text of the edit a and the title of dossier i, and $\boldsymbol{w}_T \in \mathbf{R}^D$, $\boldsymbol{w}_{T'} \in \mathbf{R}^{D'}$ are, respectively, the associated parameter vectors. We refer to this model as the WOW(*Text*) model (or WOW(T)).

We explore different ways of learning the representations $\boldsymbol{r}_a$ and $\boldsymbol{r}_i$ from (a) pre-trained word embeddings and (2) by training embeddings on our dataset. With pre-trained embeddings, $\boldsymbol{r}_a$ is the concatenation of three vectors that are the representations of the deleted text, inserted text, and the context of the edit, as explained in Section 3.3. Each of these vectors are the averages of the pre-trained word embeddings of the words in these parts of the text, and $\boldsymbol{r}_i$ is the average of the pre-trained embeddings of the words in the title of dossier i. We use two sets of pre-trained embeddings trained with the word2vec algorithm [129]: (a) 300-dimensional embeddings trained on Google News [68] and (b) 200-dimensional Law2Vec embeddings trained on legal texts of the EU, the US, the UK, Canada, and Japan[28].

Table 3.4 – Variations of our model by combination of features (explicit, latent, and text features).

Model	Equation	Explicit	Latent	Text
WOW	(3.2)	–	–	–
WOW(*Explicit*)	(3.3)	✓	–	–
WOW(*Latent*)	(3.4)	–	✓	–
WOW(*Text*)	(3.5)	–	–	✓
WOW(*XL*)	(3.3) & (3.4)	✓	✓	–
WOW(*XT*)	(3.3) & (3.5)	✓	–	✓
WOW(*LT*)	(3.4) & (3.5)	–	✓	✓
WOW(*XLT*)	(3.3), (3.4) & (3.5)	✓	✓	✓

We also learn embeddings from our dataset by using the supervised fastText model for text classification [95]. In the simplest version of this model, a D-dimensional embedding is learned for each word (and n-grams) in a dataset. A piece of text is then classified with a softmax layer by representing it as the average of the word embeddings. We use the learned word and bigram embeddings to construct $\boldsymbol{r}_a$ and $\boldsymbol{r}_i$.

The original fastText model is defined, however, for classification of homogeneous pieces of text into a fixed set of classes. This does not directly apply to our problem, as (a) the text features for the edit are of three types (deleted text, inserted text, and context) and (b) the size of a conflict $|\mathcal{C}| = K$ varies from a data point to another. We solve the first problem by prepending tags (`<del>`, `<ins>`, and `<con>`) to each word to enable the model to learn separate embeddings for the same word in different types of text feature. We solve the second problem by training the embeddings on a binary classification task of edit acceptance (based only on the text), and by using the embeddings learned on this ad-hoc task into the WOW models. We learn the embeddings for the words in the title by training a different fastText model to predict the acceptance of an edit from the title only. This is equivalent to predicting the probability of acceptance of the status quo for each dossier, given its title. For our experiments in Section 3.6, we use the fastText embeddings rather than pre-trained embeddings, because the former performed better on the ad-hoc binary classification task.

Hybrid Models We combine WOW(*Explicit*), WOW(*Latent*), and WOW(*Text*) together to obtain hybrid models with different components. This helps us understand the contribution of each type of features to the performance, in Section 3.6. We summarize all the possible combinations in Table 3.4, and we sort them by increasing levels of complexity. The WOW model has no features at all and will serve as a baseline. The WOW(*XLT*) combines explicit, latent, and text features together, and it has the highest complexity.

3.5.4 Learning the Parameters

Each observation n is a triplet $(\mathcal{C}_n, i_n, l_n)$ of (a) a set of conflicting edits $\mathcal{C}_n$ with $|\mathcal{C}_n| = K_n > 0$, (b) a dossier i_n on which the edits are proposed, and (c) a label $l_n \in \mathcal{C}_k \cup \{i_n\}$ indicating which of the K_n edits or the status quo is accepted. Given a dataset of N independent triplets $\mathcal{D} = \{(\mathcal{C}_n, i_n, l_n) \mid n = 1, ..., N\}$ and given a vector $\boldsymbol{\theta}$ of all the parameters in our model, we learn $\boldsymbol{\theta}$ by minimizing their negative log-likelihood under $\mathcal{D}$

$$-\ell(\boldsymbol{\theta}; \mathcal{D}) = \sum_{n=1}^{N} \sum_{a \in \mathcal{C}_n} \left[\mathbf{1}_{\{l_n = a\}} \log \mathbf{P}\left(a \succ_{i_n} \mathcal{C}_n - \{a\}\right) + \mathbf{1}_{\{l_n = i_n\}} \log \mathbf{P}\left(i_n \succ \mathcal{C}_n\right) \right],$$

where $\mathbf{P}\left(a \succ_{i_n} \mathcal{C}_n - \{a\}\right)$ and $\mathbf{P}\left(i_n \succ \mathcal{C}_n\right)$ depend on $\boldsymbol{\theta}$. In order to avoid overfitting, we add regularization to the negative log-likelihood. We pre-process our dataset by keeping only the dossiers for which more than 10 edits have been proposed and only the MEPs who have proposed more than 10 edits. Hence, we obtain a dataset of $N = 125733$ data points for EP7 and $N = 140763$ data points for EP8. In the WoW(*Explicit*) and the WoW(*Text*) models, the log-likelihood is convex, and we find optimal parameters by using an off-the-shelf convex optimizer (L-BFGS-B [23]). In the WoW(*Latent*) model, the bi-linear term breaks the convexity, and we can no longer ensure that we will find parameters that are global optimizers. In practice, by using a stochastic gradient descent algorithm (Adagrad [49]), we are still able to find good model parameters without convergence issues.

3.6 Experimental Results

3.6.1 Baselines

We start by introducing the baselines against which we compare our models. For each baseline and for our models, we assume a set of K conflicting edits $\mathcal{C} = \{a, b, \ldots\}$ proposed on dossier i, for which we want to model the probability that an edit $a \in \mathcal{C}$ is accepted over edits $b, \ldots$ on this dossier. We denote this probability by $\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right)$, and we denote the probability that the status quo wins, *i.e.*, that the original text proposed by the Commission is kept, by $\mathbf{P}\left(i \succ \mathcal{C}\right) = 1 - \sum_{a \in \mathcal{C}} \mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right)$.

Naive Classifier The *naive classifier* predicts a uniform probability for each outcome, *i.e.*, for each of the conflicting edits or the status quo to win, as

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \mathbf{P}\left(i \succ \mathcal{C}\right) = \frac{1}{K+1}.$$

Random Classifier The *random classifier* learns the prior probability $p^{(K)}$ that the status quo wins for each conflict size $|\mathcal{C}| = K$, and it predicts

$$\mathbf{P}\left(i \succ \mathcal{C}\right) = p^{(K)}.$$

It predicts uniformly each of the edits to win as

$$\mathbf{P}\left(a \succ_i \mathcal{C} - \{a\}\right) = \frac{1 - p^{(K)}}{K}.$$

3.6.2 Experimental Setting

We report the cross-entropy loss to evaluate the baselines and our models. Let $(\mathcal{C}_n, i_n, l_n)$ be an observation. We compute

$$\ell_n = \begin{cases} -\log p(l_n \succ_{i_n} \mathcal{C}_n - \{l_n\}) & \text{if } l_n \in \mathcal{C}_n, \\ -\log p(i_n \succ \mathcal{C}_n) & \text{if } l_n = i_n. \end{cases} \tag{3.6}$$

We report the average value for all N points in our test set as $\ell = \frac{1}{N}\sum_n \ell_n$. We randomize our dataset and we split it into 80% for training, 10% for validation, and 10% for the final evaluation. Note that an edit can be involved in several conflicts. For example, in Figure 3.2, edit c is in involved in two conflicts: $\mathcal{C}_1 = \{c, d\}$ and $\mathcal{C}_2 = \{c, e\}$. Hence, we assign conflicts to each set so that an edit is present in exactly one set. We combine both the training and the validation sets to fit our model before evaluating it on the test set. We set the number of latent dimensions L and the regularizers, and we choose the best word embeddings, by held-out validation. This results in fastText of dimension $D = D' = 10$, with bigrams.

3.6.3 Predictive Performance

We show in Figure 3.4 the overall performance of all variations of our model (with and without explicit, latent, and text features) over EP7 and EP8, and we compare them against the naive and the random predictors, as well as against the WOW model. All our models outperform the baselines, and WOW(XLT) outperforms all other models. Including explicit features improves the performance of the predictions in terms of the cross entropy by 7% for EP7 and 6% for EP8 over the simpler WOW model. On EP7, WOW(L) improves the performance by 12% and WOW(T) by 7%, whereas for EP8 the difference between the two models is smaller (10% increase for WOW(L) and 8% for WOW(T)). Hence, the text features provide a greater improvement for EP8 than for EP7, while the latent features provide a greater improvement for EP7 than for EP8. The difference between WOW(XL) and WOW(L) (0.010 for EP7 and 0.013 for EP8) is less than the difference between WOW(XT) and WOW(T) (0.034 for EP7 and 0.035 for

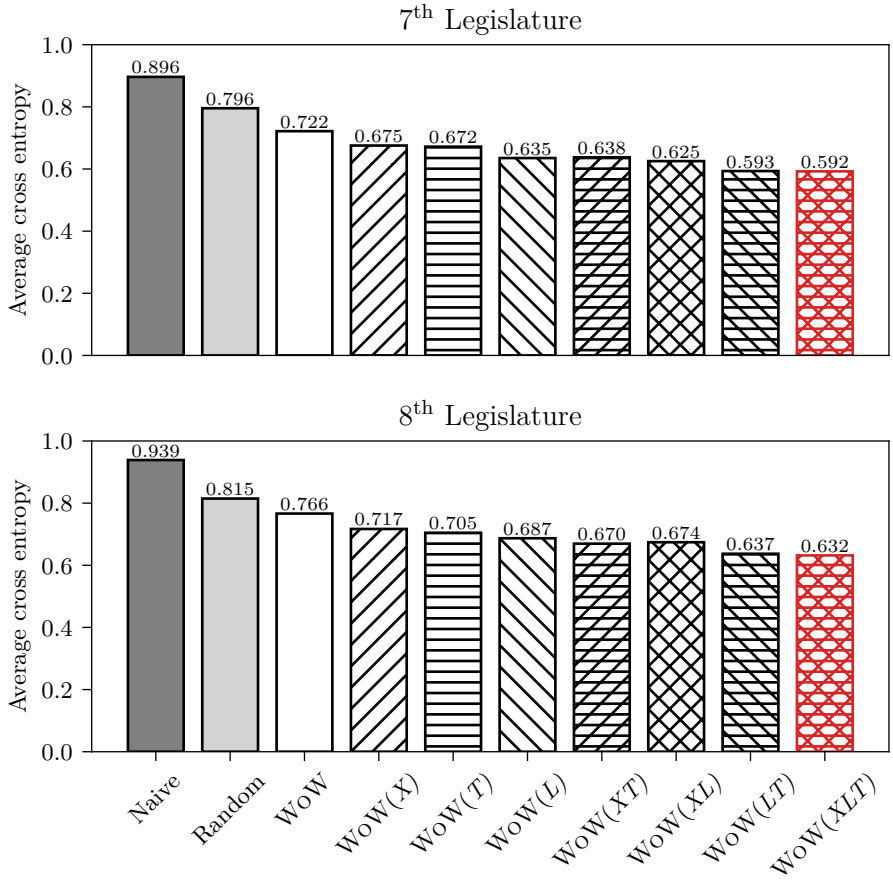


Figure 3.4 – Average cross-entropy loss of the baselines and our models. Combining the explicit, latent, and text features help obtain the best performance.

EP8), as the latent features absorb the effects of the explicit features more than the text features do. Finally, combining the text and latent features provides high performance, but further combining them with explicit features leads to the best performance.

3.6.4 Error Analysis by Conflict Size

We explore how the WoW(XLT) model performs on conflict of different sizes in the test set for EP8 (we observe a similar behaviour on EP7). We bin the conflict size so that there are at least 100 data points in each bin. The distribution of conflict size is exponentially decreasing: There are 8462 conflicts of size 1 (i.e., an edit is in conflict with the status quo only), 3063 conflicts of size 2 (i.e., two edits are in conflict, as well as with the status quo), and 140 conflicts of size 7 and more. We compare the average cross entropy of the WoW(XLT) model with that of the random predictor and that of the WoW model. In Figure 3.5, we see that while the loss generally increases with conflict size for all three models, it increases less rapidly for the WoW(XLT) model than for the WoW model. This suggests that the explicit, latent, and text features enable the model

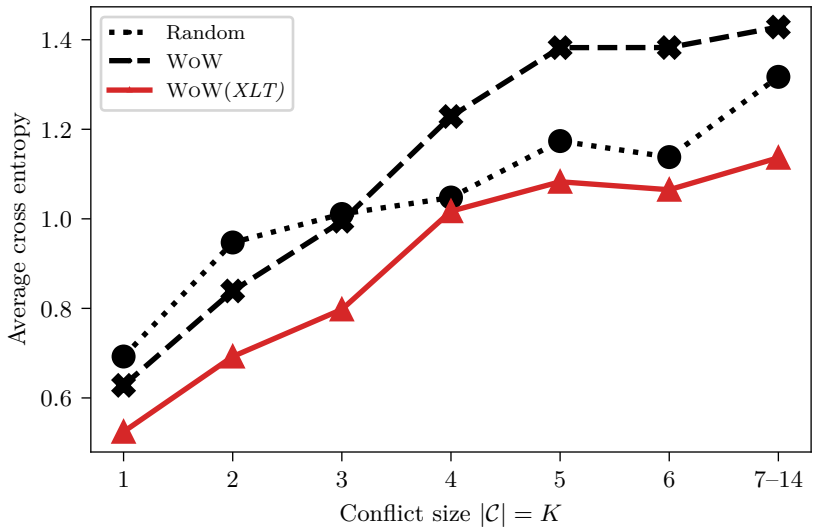


Figure 3.5 – Average cross-entropy loss per conflict size $|\mathcal{C}| = K$. The loss of the WoW(XLT) model increases less rapidly than the loss of the baselines.

to exploit the increasing complexity of data points to make more accurate predictions. We also see that for conflicts of size 4 and higher, the WoW model performs worse than the random predictor, but the WoW(XLT) model is able to outperform it.

3.6.5 Contribution of Explicit Features

To understand the contribution of the explicit features to the predictive performance, we show in Figure 3.6 the decrease in cross-entropy loss of WoW(*MEP*) (all MEP features but the rapporteur feature), WoW(*Rapporteur*) (rapporteur feature only), WoW(*Edit*), and WoW(*Dossier*) over WoW. The dossier features contribute virtually nothing to the predictive performance (the difference is at the fourth decimal point). Similarly, for EP7, the nationality, political group, and gender features of WoW(*MEP*) contribute very little. For EP8, these features improve the performance, but not as much as the edit features. This suggests that these features have limited influence on the predictions. Nationalities and political groups have been qualitatively analyzed in the literature in the context of their influence on MEPs' voting behaviour [84, 36, 133, 111]. To the best of our knowledge, there is no analysis of their effect on the amending process. Interestingly, for EP7, combining all features into the WoW(X) model leads to a performance boost that is greater than the sum of each individual feature groups.

3.6.6 Interpretation of Explicit Features

To get insights into the dynamics of the legislative process, we interpret the values of the parameters of WoW(XLT) trained on the full dataset for EP8 (combining training,

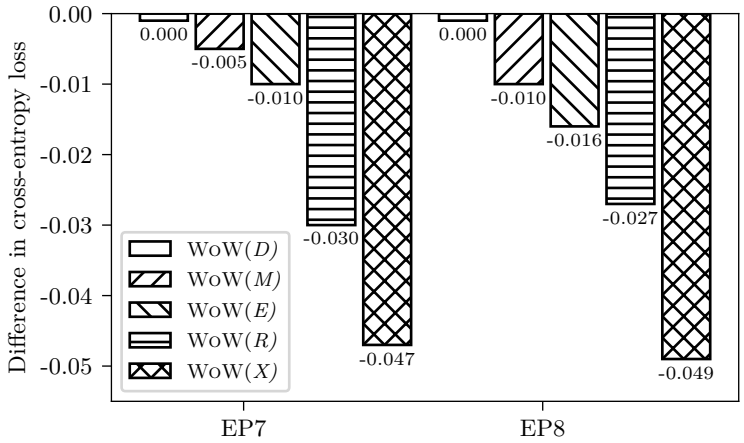


Figure 3.6 – Difference in cross-entropy loss over WoW of different models. The rapporteur feature and the edit features contribute more to the predictive performance than the MEP and dossier features.

validation, and test data). Let $w_f \in \mathbf{R}$ be the value of the parameter associated with feature f. The rapporteur feature r of WoW(*Rapp.*) provides a greater decrease in loss. This *rapporteur advantage* complements the findings of Costello and Thomson [40], conducted by interviewing key informants over EP5 (1999-2004) and EP6 (2004-2009). They show that the rapporteur, with their particular role, has some influence on the legislative process, albeit constrained. We note that, according to our model, the rapporteur advantage has slightly increased in EP8 ($w_r = 1.19$) compared to EP7 ($w_r = 1.12$).

These explicit features enable us to explain what contributes to the success of an edit. We report here (and in subsequent sections) the results for EP8 only. All other things being equal, a female ($w_{\text{fem}} = -0.02 > -0.04 = w_{\text{mal}}$) MEP from Latvia and whose party belongs to the group of the European People's Party (center-right) has the highest chance to see her edit accepted. This edit has even higher chances if it inserts ($w_{\text{ins}} = -0.03 > w_{\text{del}} = -0.13 > w_{\text{rep}} = -0.22$) a short portion of text (the feature associated with both insertion and deletion length is negative) in a part of the law that is not its body or its preamble (w_{art}, w_{rec} and w_{para} have the lowest value among the seven article types). Adding a justification also increases the probability of an edit being accepted ($w_{\text{jus}} = 0.08$), as well as edits from the opinion committee (referred to as the "outsider committee" feature in Table 3.2, $w_{\text{out}} = 0.16$).

For the dossier features, our model learns that it is harder to make edits on reports, as compared to opinions ($w_{\text{rep}} = 0.33 > -0.26 = w_{\text{opi}}$). As explained in Section 3.3, reports are voted by the whole Parliament. Therefore, they have a greater influence on the final law, and we expect that MEPs make it more difficult for competing edits to be accepted in reports. Finally, our model also learns that it is harder to make edits for decisions and directives, as compared to regulations ($w_{\text{dec}} = 0.25 > w_{\text{dir}} = 0.12 > w_{\text{reg}} = 0.10$).

Controversy of Dossiers Table 3.5 provides a list of the ten dossiers in EP8 with the highest inertia parameter d_i and the ten dossiers with the lowest d_i. Overall, the values of d_i correlate well with the number of nodes, the number of cliques, the average size of cliques, and the edit acceptance rate. These four metrics are a good proxy to the level of activity by MEPs in the amending process of a given dossier. Higher activity, possibly due to higher controversy, leads to higher value of d_i. We note, however, that some of the top-10 dossiers have a small number of edits. This shows that the inertia parameters capture more information than simply some of these descriptive statistics.

The top-ten dossiers include laws with high stakes about financial markets, the environment, vast investment programmes, and assistance to member states: The "Screening of foreign direct investments" sets a framework to better equip the EU for investments from non-EU countries. It has crucial implications for companies, workers, governments, and citizens. The "European Supervisory Authorities on financial markets" sets strict regulations for the financial markets. "InvestEU" and the "Horizon Programme" are vast investment programmes for innovation and research. The "Cost-effective emission reductions and low-carbon investments" is one of the implementations of the Paris Climate Agreement. Finally, The infamous "Copyright in the Digital Single Market", considered to be a threat to freedom of expression on the Web by its opponents, sparked public protests in several cities. The reporting committee publicized that "MEPs have rarely or never been subject to a similar degree of lobbying before" [146].

3.6.7 Interpretation of Text Features

In Figure 3.4, we observe that the text features contribute significantly to improving the performance. We use the learned parameter vectors $\boldsymbol{w}_T$ and $\boldsymbol{w}_{T'}$ of WOW(XLT) to identify words and bigrams that have the most predictive power. First, we rank the words and bigrams of the edit text, according to the dot product of their embeddings with $\boldsymbol{w}_T$. The top-k terms (having a positive dot product) contribute the most towards acceptance of the edit, whereas the bottom-k terms (having a negative dot product) contribute most towards rejection of the edit. The opposite holds for the terms of the title and their dot product with $\boldsymbol{w}_{T'}$.

We look at the top 50 terms for each feature and prediction outcome and find some interesting patterns among these terms, although not all of them are easy to interpret. Note that we have more than 10 000 unique terms for the edit text and more than 1 000 unique terms for the title, hence we consider only the most predictive terms near the ends of the ranking.

One of the bigrams that, when deleted, is predictive of acceptance is *any other*, which is commonly used to widen the scope of the law (as in "contractual or any other duty"). Interestingly, the bigrams *human rights* and *data protection* are also predictive of

Table 3.5 – Top-10 and bottom-10 inertia parameters d_i for dossiers in EP8.

d_i	Type	Comm.	Title	# edits	# conf.	avg. cf. sz.	% acc.
2.018	Rep.	INTA	Screening of foreign direct investments	1040	272	3.1	2.6
1.958	Opi.	ITRE	Cost-effective emission reductions and low-carbon investments	1756	385	4.2	5.1
1.879	Opi.	PETI	Discontinuing seasonal changes of time	81	25	2.9	6.2
1.619	Rep.	ENVI	Health technology assessment and amending	133	14	2.0	4.5
1.512	Rep.	ECON	European Supervisory Authorities on financial markets	48	12	2.2	10.4
1.447	Rep.	ECON	InvestEU Programme	1194	297	2.9	27.0
1.393	Rep.	ITRE	Horizon Europe	2013	467	3.0	9.8
1.386	Rep.	INTA	Macro-financial assistance to the Republic of Moldova	36	8	2.4	13.9
1.286	Rep.	AFET	Instrument for Pre-Accession Assistance	732	239	2.5	20.6
1.282	Rep.	JURI	Copyright in the Digital Single Market	2657	577	4.3	2.6
-1.651	Opi.	REGI	Common agricultural policy	105	4	2.0	82.9
-1.655	Opi.	DEVE	Promotion of the use of energy from renewable sources	62	3	2.0	90.3
-1.681	Opi.	AGRI	Establishing Horizon Europe	43	8	2.0	65.1
-1.686	Opi.	AGRI	Governance of the Energy Union	150	30	2.3	56.7
-1.754	Opi.	JURI	Insurance against civil liability with motor vehicles	29	2	2.0	89.7
-1.779	Opi.	BUDG	Common agricultural policy	15	0	0.0	100.0
-1.780	Opi.	AGRI	European Regional Development and Cohesion Fund	129	13	2.2	58.1
-1.812	Opi.	ECON	Prevention and prosecution of criminal offences	81	2	2.0	86.4
-2.065	Opi.	DEVE	Unfair trading practices in in the food industry	63	6	2.0	84.1
-2.284	Opi.	TRAN	Protection of the collective interests of consumers	121	26	2.1	66.9

acceptance when deleted. The word *should*, which is used to add recommendations, is predictive of acceptance when inserted, while adding *must*, which is used for obligations, is predictive of rejection. We see that *best* is predictive of acceptance, which is commonly used to make a requirement stronger (as in "best available scientific evidence", "best possible way"). Adding *positive* and *positive impact* predicts acceptance, whereas adding *negative* predicts rejection. Adding the word *inserted*, which commonly refers to inserting new articles in existing laws, is predictive of acceptance, whereas *deleted* is predictive of rejection.

Considering the words in the context, we see that *firearms*, *resettlement*, *terrorist* and *fingerprints* are predictive of rejection. This could be because the laws related to these topics are controversial, hence many edits are rejected due to conflicts. For the words in the title, we see that *customs*, *community*, *financial*, *fisheries*, and *general budget* are predictive of acceptance, whereas *market*, *framework*, *structural reform*, *emission*, and *greenhouse gas* are predictive of rejection. This suggests the relative ease or difficulty of editing laws related to these topics, and it correlates well with the values of the difficulty parameters d_i: The top-50 dossiers with the highest difficulty parameters contain high-controversy dossiers about establishing frameworks for the screening of foreign investments and vast public investment programs (InvestEU and Horizon Europe), as well as regulation of the financial market, copyright in the digital market, and carbon-emission reduction. The bottom-50 dossiers with the lowest difficulty parameters contain low-controversy dossiers about cohesion within the EU, financial rules, fisheries, and the community code on visas.

3.6.8 Interpretation of Latent Features

The latent features improve the predictions overall and help capture the complex dynamics of the legislative process. The best number of latent dimensions is $L = 20$ for the models including latent features. In order to interpret the latent features, we gather the latent vectors $\boldsymbol{y}_i$ learned by WoW(XLT) into a matrix $Y = [\boldsymbol{y}_i]$. We apply principal component analysis and keep the top-10 and bottom-10 dossiers from each of the first two principal components in EP8. We use t-SNE [115] to represent these forty dossiers in a two-dimensional space, and we show the projection in Figure 3.7.

We distinguish four clusters. The cluster at the top-left contains dossiers about fuel quality, renewable energy, trade of animals, and sustainable investments. It also contains dossiers about electronic communications, the processing of personal data, and sharing public information. We interpret this cluster as *environment and communications*, and we highlight with green triangles the corresponding dossiers. The cluster at the top-center contains dossiers about the establishment of defense funds, the prosecution of criminal offenses, and the identification of criminals between member states. It also contains dossiers about the protection of workers, businesses, refugees, internal markets,

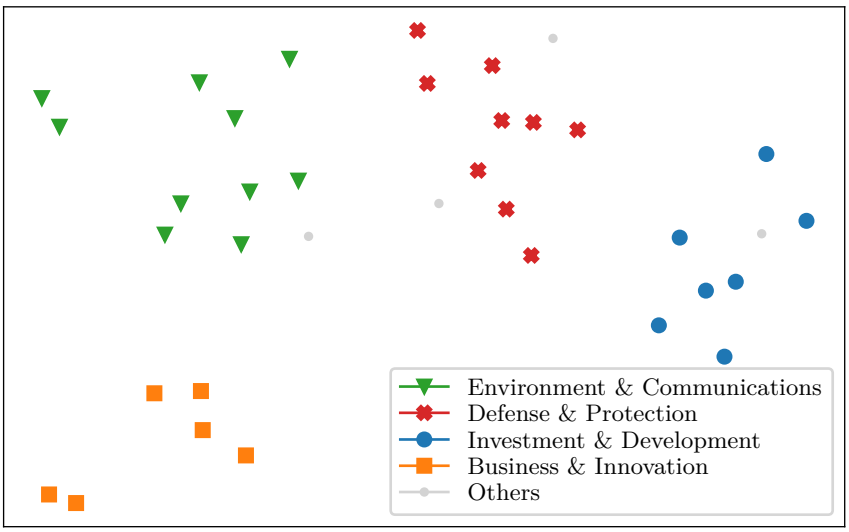


Figure 3.7 – Visualization with t-SNE of the top-10 and bottom-10 dossiers on the first two principal components in EP8. There are four clusters: Environment and Communications, Defense and Protection, Investment and Development, and Business and Innovation.

and cultural goods. We interpret this cluster as *defense and protection* (red crosses). The cluster at the top-right contains dossiers about vast investment and development programmes, finance, and the development of internal markets. We interpret this cluster as *investment and development* (blue dots). Finally, the cluster at the bottom-left contains dossiers about economic competitiveness and innovation, as well as frameworks for business development and the funding of start-up companies. We interpret this cluster as *business and innovation* (orange squares).

3.6.9 Solving the Cold-Start Problem

We explore how to solve the cold-start problem by defining a second predictive problem: Given a dossier i *for which we have never seen an edit*, and given a conflict $\mathcal{C} = \{a, b, \ldots\}$, we want to predict which of the edits or the status quo wins. We order the dossiers by the date a committee received a proposal, and we use the dossiers that contain the first 80% of the conflicts as a training set. We use the next 10% as validation set, and we keep the last 10% aside as test set. We ensure that no edits in the training set leak into the validation and test sets. This scenario is more realistic because we make predictions about new dossiers that the model has never observed before.

We report, in Table 3.6, the results for WoW(*Explicit*), WoW(*Text*), and WoW(XT), together with the baselines. The latent features cannot be used for this task, as the dossier embeddings $\boldsymbol{y}_i$ are unavailable for new dossiers. For our models, the difficulty parameter d_i is set to the average difficulty learned in the training set. The random predictor, which learns the prior probability of the status quo winning for each conflict

Table 3.6 – Average cross entropy of the baselines and our model on predicting new, unseen dossiers.

Type	Model	Avg. cross entropy
Baseline	Naive	0.947
	Random	**0.800**
	WoW	0.873
Ours	WoW(*Explicit*)	0.784
	WoW(*Text*)	0.839
	WoW(XT)	**0.759**

size, performs the best out of all the baselines, and it outperforms WoW(*Text*). Our approach outperforms only the random predictor when including explicit features. This suggests that the dossier features help us make more accurate predictions by learning parameter values for the type of dossier, its legal act, and its committee in charge. In this case, adding text features further boosts the performance.

The overall performance, however, is mixed: The improvement of WoW(XT) over the random predictor is rather small. One possible explanation is that the legislative process might be non-stationary. Hence, our model overfits on the training set, which is very different from the test set. The task is also unfair to our model, as in a real setting, predictions would be made for the next dossier only. In the current setting, we make predictions for all future dossiers. We keep further investigations of this aspect for future work.

3.7 Related Work

Amendment analysis in the European Parliament has been studied by the political science community on datasets of small size [100, 183, 101, 6]. The effect of the rapporteur on the success of an amendment has been studied in previous legislature periods and in specific committees [57, 90]. Predicting edits on collaborative corpora of documents has been studied in the context of peer-production systems, such as Wikipedia [48, 2, 163] and the Linux kernel [93, 201]. A whole body of literature covers the conflicts between two Wikipedia edits [173, 202] and the quantification of controversy of Wikipedia articles [166, 165]. The notion of conflict is, however, different in our setting, where multiple edits can be in conflict at the same time: The task of predicting which edit will be accepted out of all the conflicting edits is more complex, and classic approaches cannot be used. In this work, we take a peer-production viewpoint on the law-making process and propose a model of the acceptance of the legislative edits. Our approach generalizes to any peer-production system in which (meta) features of the users and items can be extracted and in which edits can be in conflict with one another.

We use the text of the edits and dossiers as features for classification. Text classification is a well-studied problem in natural language processing. A simple baseline is to apply linear classifiers to term-frequency inverse document-frequency (TF-IDF) vectors [94]. However, these models do not capture the synonymy relation between words, hence suffer from poor generalization. Models based on neural networks show better performance on this task [207]. They tend, however, to require larger datasets, and the features they learn are harder to interpret. The fastText model [95] bridges the gap between the two: It learns embeddings from linear models. We adapt this approach to our problem of edit classification, as edits are inhomogeneous pieces of text. Edit modelling has been studied using neural models[205, 73] that suffer from the aforementioned issues of dataset size and interpretability. In the War of Words models, we combine text features and non-text features to take into account the dynamics of the legislative process. Legal texts also have features and structures that set them apart from other domains. For example, the word "should" has a strong legal significance, whereas it is commonly removed as a stop word.

Our model draws inspiration from probabilistic models of choice, described in Section 1.2. First, it borrows from the logit model to model the competitive dynamics between MEPs. These approaches learn a real-valued score for individuals and model the probability that one individual wins over another as a function of the difference of their scores. Second, it borrows from the Rasch model to model the competitive dynamics between MEPs and the status quo. These approaches learn a real-valued strength for each individual and a real-valued difficulty for each item, and they model the probability that an individual wins over the item as a function of the difference of the strength and the difficulty. Our model unifies both approaches by learning a strength for each MEP and a difficulty for each dossier, considering (i) conflicts between MEPs and (ii) conflicts between MEPs and the status quo.

3.8 Summary

In this chapter, we have introduced a new dataset of legislative edits and a model of edit outcomes. Our dataset provides rich information on a long-term, dynamical process of interactions between parliamentarians. Our proposed model learns a skill parameter for MEPs who propose edits and an inertia parameter for the law proposals that resist to change. Our model also incorporates (a) explicit features of the edits, of the MEPs, and of the dossiers, (b) latent features of the MEPs and dossiers, and (c) text features of the edits and dossiers. Each of the three classes of additional features improve the performance significantly, and the best performance is achieved by combining all features. We interpreted the values of the learned parameters to gain insights into the legislative process. We provided interpretation of all explicit features to characterize what makes the success of an edit more likely. We have shown that the latent features capture the representation of MEPs and dossiers in an ideological space. We have analyzed the words

and bigrams in different parts of an edit and a dossier in terms of their influence on the acceptance probability. We have also analyzed the performance of our model on subsets of the test set based on conflict size, and we have shown that our best model can leverage the features of the data to make more accurate predictions on conflicts of higher size than other baselines. Finally, we have described how to use our model for predicting edits made on new, unseen dossiers.

Applications and Broader Impact We believe that approaches such as ours are helpful to political scientists, journalists and transparency observers, and to the general public: First, it could be useful in validating theoretical hypotheses using large-scale datasets and advanced computational methods. Second, it could help uncover lesser-known facts, such as controversial dossiers that slipped under the radar. Finally, the greater transparency that results from these insights can enhance trust in public institutions and strengthen democratic processes.

Perspective First, we currently use pre-trained word embeddings and embeddings trained on an ad-hoc binary classification task. We plan to explore how to learn text embeddings in an end-to-end manner using the conflictive structure of the War of Words model. Second, as shown in Section 3.6.9, our model has only limited predictive power on edits made on future dossiers. We plan to further explore how to exploit the temporality of the data and how to develop a dynamical model able to take into account the non-stationarity of the law-making process. Finally, the current setting of the predictive task assumes that conflicts are independent of each other; because an edit can be involved in multiple conflicts, they are not always independent. We plan to develop more advanced models by leveraging these correlations between conflicts. For example, we plan to explore how to include latent features and text features to the mixed logit model [80].

4 Voting Processes

In this chapter[1], 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. While our approach does not use discrete-choice models directly, the problem we tackle is related to one of the most fundamental choice processes: voting. Our 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[2]. We also explore the regional representations in terms of ideological and cultural patterns. Finally, we deploy an online Web platform[3] to provide real-time vote predictions in Switzerland and a data visualization tool to explore voting behavior.

4.1 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 chapter, 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,

[1] This chapter is based on Immer et al. [91].
[2] Data and code publicly available on https://github.com/indy-lab/submatrix-factorization.
[3] The platform is accessible on https://www.predikon.ch.

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 [177, 178]. 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 4.3, our algorithm can be applied to other countries and in other settings.

In Section 4.2, 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 4.3.

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 4.4.

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.

4.2 Methodology

4.2.1 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, $\theta \coloneqq \boldsymbol{x}^\intercal \boldsymbol{w} \in \mathbf{R}$ be the natural parameter, and $A(\theta) \in \mathbf{R}$ be the log-partition function. Then, the likelihood of a GLM is

$$p(y|\boldsymbol{w}, \boldsymbol{x}) = h(y) \exp\{y\theta - A(\theta)\}. \tag{4.1}$$

Point-wise predictions are obtained from the mean parameter

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

where the invertible function $g : \mathcal{D} \to \mathbf{R}$ 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 (4.1) 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 [18]. In Table 4.1, 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 [134, Chapter 9] for a detailed introduction to GLMs.

Table 4.1 – List of Generalized Linear Models. The softmax function is denoted by $\mathcal{S}$.

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

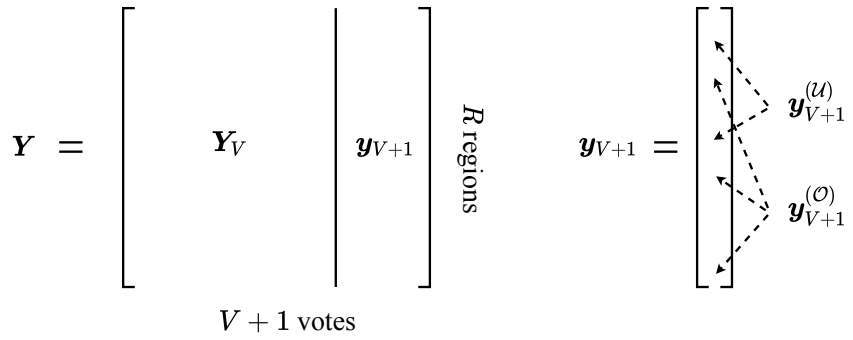


Figure 4.1 – Decomposition of the vote matrix $\boldsymbol{Y}$ into the fully observed *sub-matrix* $\boldsymbol{Y}_V$ and the new vote $\boldsymbol{y}_{V+1}$, whose results arrive sequentially. The $(V+1)$ votes are chronologically ordered and the R regions are arbitrarily ordered.

4.2.2 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[4] in $\boldsymbol{Y}$, which we denote by $\boldsymbol{y}_{V+1}$. Let $\boldsymbol{Y}_V \in \mathbf{R}^{R \times V}$ be the *sub-matrix* of all observed, historical results up to vote V. Denoting the set of consecutive integers by $[R] \coloneqq \{1, 2, \ldots, R\}$, 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 $\boldsymbol{y}_{V+1}^{(\mathcal{O})}$ and $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$ denote the observed and unobserved entries of $\boldsymbol{y}_{V+1}$, respectively. Our task is to predict the missing entries $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$ from $\boldsymbol{Y}_V$ and $\boldsymbol{y}_{V+1}^{(\mathcal{O})}$ only. Figure 4.1 depicts the structure of the matrix $\boldsymbol{Y}$.

To predict the missing entries of $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$, Etter et al. [55] use standard matrix factorization with alternating least-squares (ALS) to minimize the non-convex loss based on the

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

Frobenius norm

$$\min_{\boldsymbol{A},\boldsymbol{B}} \left\| \boldsymbol{Y}^{(\mathcal{O})} - \left(\boldsymbol{A}\boldsymbol{B}^T\right)^{(\mathcal{O})} \right\|_F, \tag{4.2}$$

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 [10, 99]. According to the Eckart-Young-Mirsky Theorem [52], the optimal solution to Equation (4.2) 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[55].

4.2.3 Algorithm

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

$$\boldsymbol{Y}_V \approx \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal, \tag{4.3}$$

where the diagonal matrix $\boldsymbol{\Sigma} \in \mathbf{R}^{D\times D}$ 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

$$\boldsymbol{X} = \boldsymbol{Y}_V\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}, \tag{4.4}$$

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 4.3. These two steps are performed offline, *i.e.*, they are performed once.

Third, we use the observed results of a new vote $\boldsymbol{y}_{V+1}^{(\mathcal{O})}$ 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 (4.1), with regularization parameter $\lambda \in \mathbf{R}$,

$$\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1}^{(\mathcal{O})}) = -\sum_r \log p(y_{r,V+1}^{(\mathcal{O})} | \boldsymbol{w}, \boldsymbol{x}_r) + \lambda\|\boldsymbol{w}\|_2^2, \tag{4.5}$$

where $y_{r,V+1}^{(\mathcal{O})} \in \mathbf{R}$ 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

$$\boldsymbol{y}_{V+1}^{(\mathcal{U})} := g^{-1}\left(\boldsymbol{X}^{(\mathcal{U})}\boldsymbol{w}_*\right), \tag{4.6}$$

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 $\boldsymbol{y}_{V+1}^{(\mathcal{O})}$ and $\boldsymbol{y}_{V+1}^{(\mathcal{U})}$, weighted by the population of each region r. We summarize these steps in Algorithm 4.1.

Algorithm 4.1 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 $\boldsymbol{Y}_V \approx \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$. ▷ Equation (4.3)
2: Project $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}$. ▷ Equation (4.4)
3: Optimize $\boldsymbol{w}_* = \arg\min_{\boldsymbol{w}} -\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1}^{(\mathcal{O})})$. ▷ Equation (4.5)
4: Predict $\boldsymbol{y}_{V+1}^{(\mathcal{U})} = g^{-1}\left(\boldsymbol{X}^{(\mathcal{U})}\boldsymbol{w}_*\right)$. ▷ Equation (4.6)

To predict the outcomes of referenda and elections, we use the GLMs described in Table 4.1. 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 4.1) in closed form with least-squares

$$\boldsymbol{w}_* = \left(\boldsymbol{X}^{(\mathcal{O})\intercal}\boldsymbol{X}^{(\mathcal{O})} + \lambda \boldsymbol{I}_D\right)^{-1} \boldsymbol{X}^{(\mathcal{O})\intercal}\boldsymbol{y}_{V+1}^{(\mathcal{O})}, \tag{4.7}$$

where $\boldsymbol{X}^{(\mathcal{O})} \in \mathbf{R}^{|\mathcal{O}|\times D}$ are the representations of the observed regions, $\boldsymbol{y}_{V+1}^{(\mathcal{O})} \in \mathbf{R}^{|\mathcal{O}|}$ 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 $|\mathcal{O}|$ observations when new observations arrive.

Although this algorithm is intuitive, considering the particular structure shown in Figure 4.1, its general performance is not obvious. In standard matrix factorization, defined in Equation (4.2), both $\boldsymbol{A}$ and $\boldsymbol{B}$ are learned together. Our algorithm fixes $\boldsymbol{A}$ to be equal to $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}$, 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 $O(n(|\mathcal{O}|D + D^3))$ for n iterations of Newton's method. With a univariate Gaussian likelihood, computing the least-squares solution has asymptotic complexity $O(|\mathcal{O}|D^2 + D^3)$, which is dominated by the $|\mathcal{O}|D^2$ term, as typically $D < |\mathcal{O}|$.

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).

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 (4.2), 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 (4.3) and Equation (4.4), our algorithm learns the latent features of the regions $\boldsymbol{a}_r = (\boldsymbol{U}\boldsymbol{\Sigma})_r = \boldsymbol{x}_r$ from historical data. In Equation (4.5), 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 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 (4.5) by

$$-\ell_p(\boldsymbol{w}; \boldsymbol{X}, \boldsymbol{y}_{V+1}^{(\mathcal{O})}) = \sum_r N_r \log p(y_{r,V+1}^{(\mathcal{O})} | \boldsymbol{w}, \boldsymbol{x}_r) + \lambda\|\boldsymbol{w}\|_2^2, \tag{4.8}$$

Table 4.2 – 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

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 (4.5) as *unweighted*. A similar argument can be trivially made for elections by letting $P_{rv}^{(i)} \sim \text{Categorical}(\boldsymbol{p}_{rv}, K)$ be a random variable describing the vote cast by voter i in region r on vote v for K political parties.

4.2.5 Limitations

By design, our approach suffers from the cold-start problem of collaborative filtering [99]. We can make predictions only when at least one past observation is available, *i.e.*, when $|\mathcal{O}| = 1$. To bypass this problem, Etter et al. [55] 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 4.4.

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 [7] to capture the dynamics of the voting process.

4.3 Experimental Results

We evaluate our algorithm on the four datasets[5] described in Table 4.2. 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.

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

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 (4.8). 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 (4.5).

4.3.1 Evaluation

For each dataset, we find the best hyperparameters using the training set only. 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 $\boldsymbol{y} \coloneqq \boldsymbol{y}_{V+1} \in \mathbf{R}^R$ be a prediction. The true national outcome $y^* \in \mathbf{R}$ is defined as

$$y^* \coloneqq \frac{1}{N} \sum_{r \in [R]} N_r y_r^*,$$

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

$$y \coloneqq \frac{1}{N} \left(\sum_{r \in \mathcal{U}} N_r y_r + \sum_{r \in \mathcal{O}} N_r y_r^* \right),$$

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

$$\begin{aligned} \mathrm{MAE}(y, y^*) &= |y - y^*|, \\ \mathrm{Acc}(y, y^*) &= \mathbf{1}_{\{y \geq 0.5 \text{ and } y^* \geq 0.5\}} + \mathbf{1}_{\{y < 0.5 \text{ and } y^* < 0.5\}}, \end{aligned} \tag{4.9}$$

where $\mathbf{1}_{\{.\}}$ 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 (4.9) 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) [46]. 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

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

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 $\mathcal{O}$ 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 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. [55] and as formulated in Equation (4.2). For the multiple outcome task, we restrict our comparison to weighted averaging.

4.3.2 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 4.4, we analyze in depth the last four votes (two votes on two dates) for which we have real, sequential data. The best combination for the Bernoulli likelihood is $\lambda = 0.01$ and $D = 25$.

In Figure 4.2, 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 4.4, 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

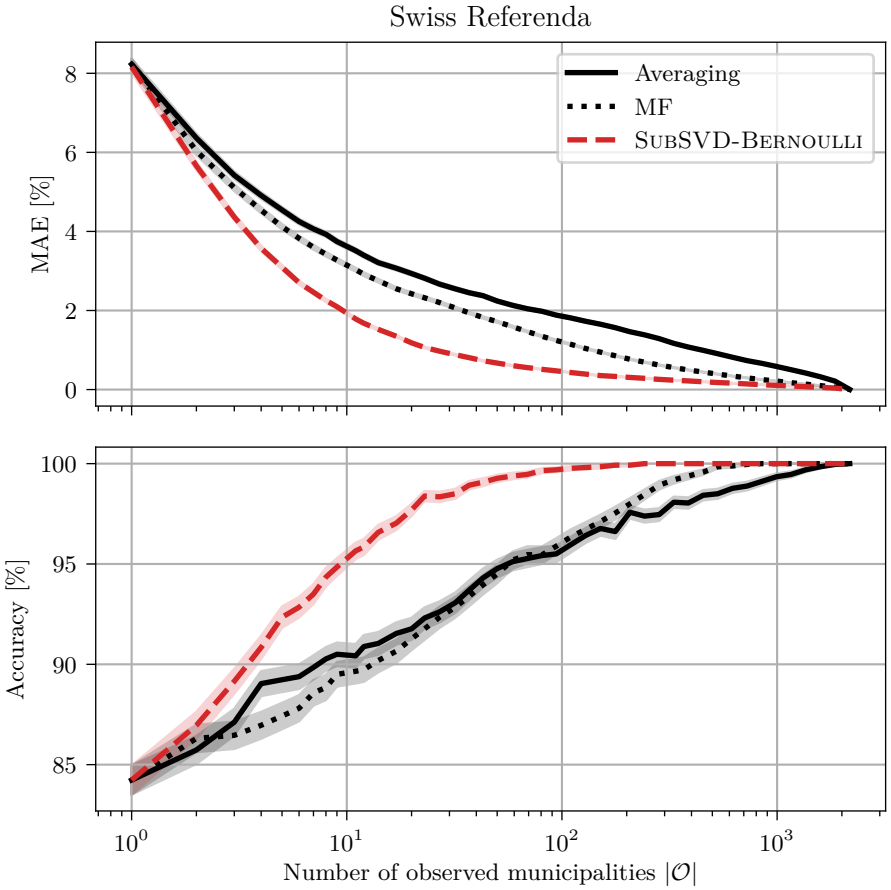


Figure 4.2 – MAE (top) and accuracy (bottom) averaged over 26 Swiss referenda and 100 reveal orders each.

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.

We explore the patterns in the feature matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}$ obtained from (4.4). In Figure 4.3, 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. [54], 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 4.4 a projection of the result matrix $\boldsymbol{Y}$ by using t-SNE [115]. 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

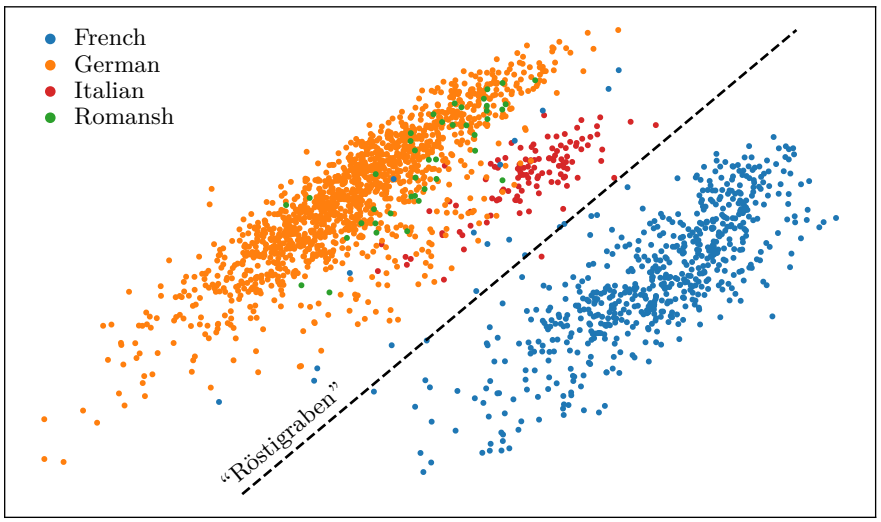


Figure 4.3 – Projection of Swiss municipalities on the first two singular vectors of referendum matrix $\boldsymbol{Y}$. Municipalities are colored according to their language.

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.

4.3.3 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 [132]. In the spirit of Nate Silver's *FiveThirtyEight* [167], 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 work. 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 best combination for the Bernoulli likelihood is $\lambda = 0.01$ and $D = 7$.

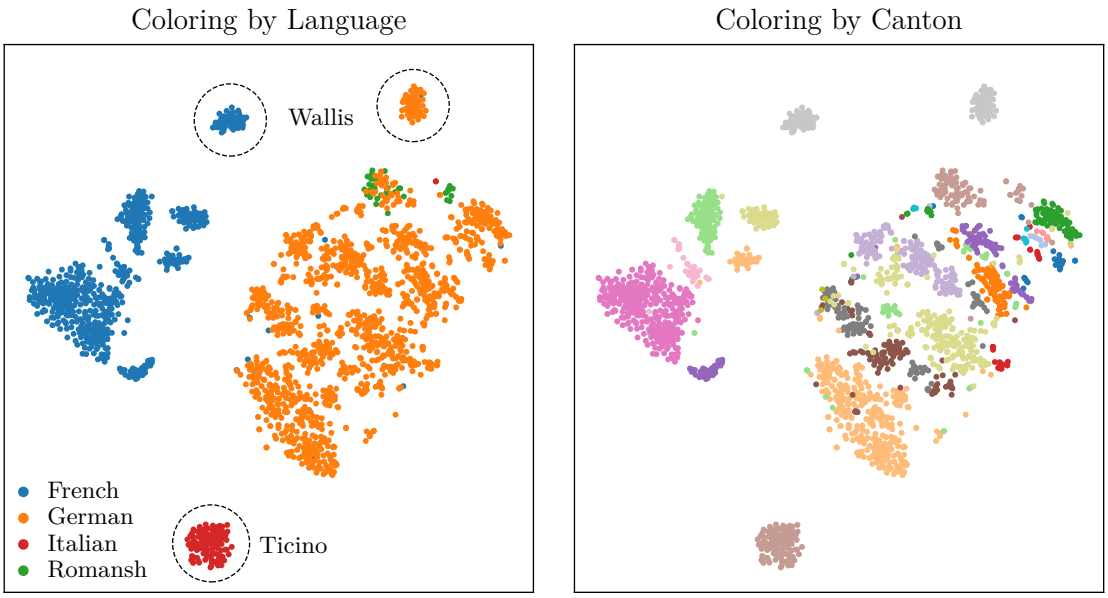


Figure 4.4 – Projection of Swiss municipalities from referendum matrix $\boldsymbol{Y}$ with t-SNE. (Left) Municipalities are colored according to their language. (Right) Municipalities are colored according to their canton (26 cantons). The bilingual canton of Wallis is split into two clusters. The only Italian-speaking canton of Ticino is isolated from the other clusters.

In Figure 4.5, 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.

4.3.4 German Legislative Election

German legislative elections take place every four years. We obtain two datasets [140] 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[6] 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

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

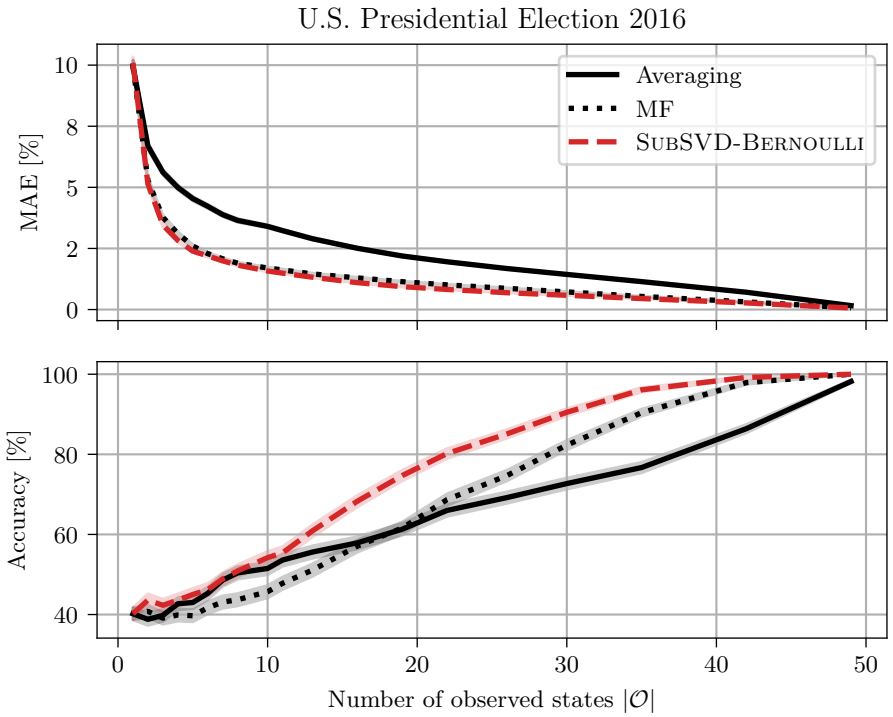


Figure 4.5 – MAE (top) and accuracy (bottom) of the popular vote of the U.S. presidential election in 2016.

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 4.6, 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 4.6, 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 4.6, 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 4.3.2, we explore the representations of the regions contained in the feature matrix $\boldsymbol{X}$ for Germany. In Figure 4.7, we plot the first two columns of $\boldsymbol{X}$, *i.e.*, a projection of the districts on the first two singular vectors of the

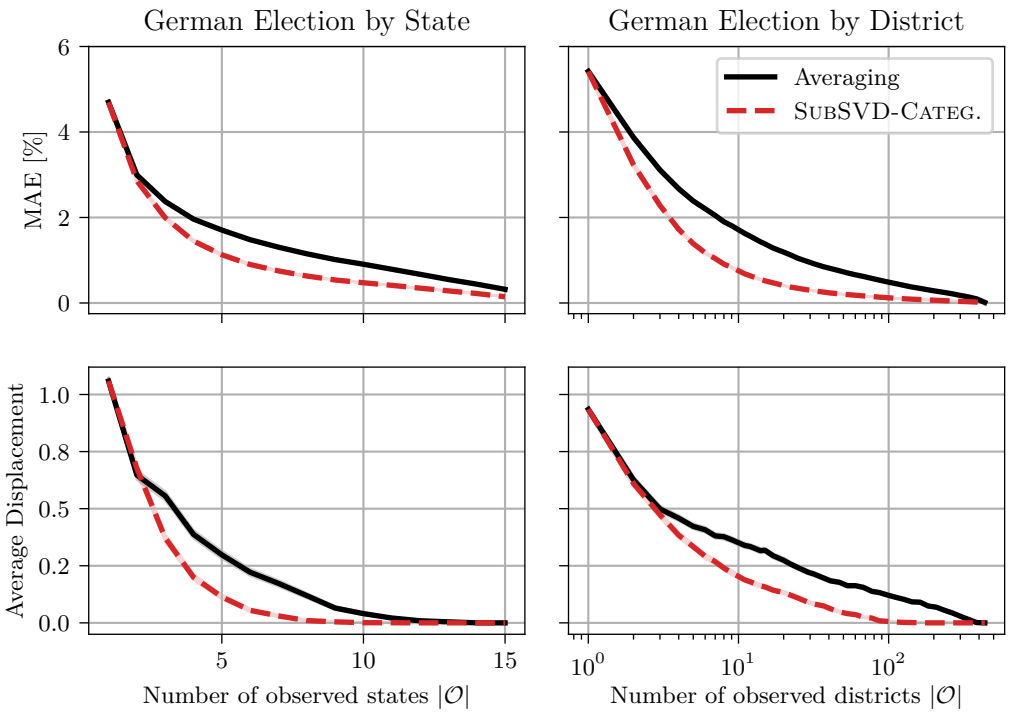


Figure 4.6 – MAE (top) and average displacement (bottom) of German legislative elections at state level in 2009 (left) and district level in 2005 (right).

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.[7] 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.

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

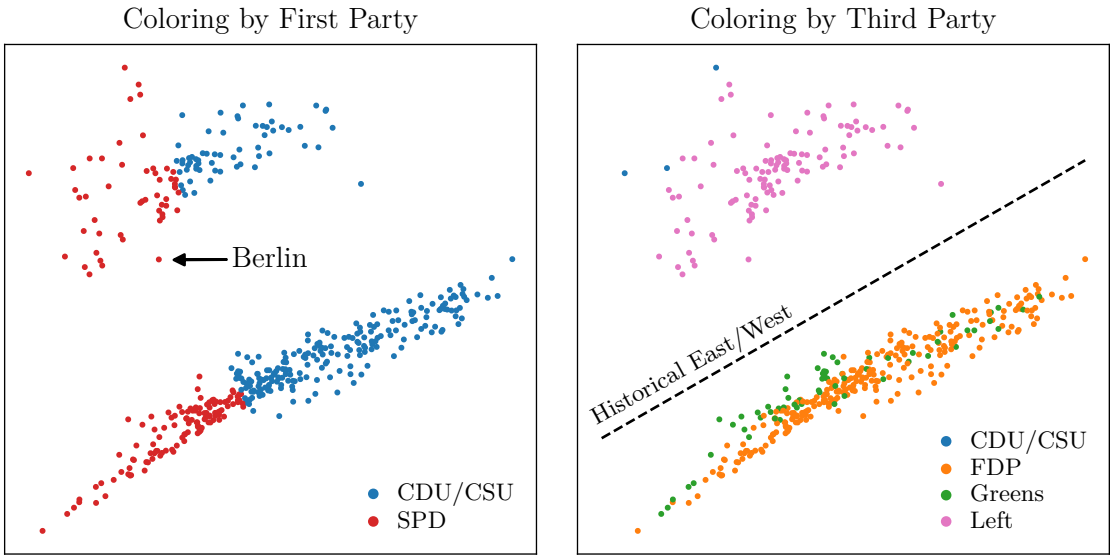


Figure 4.7 – Projection of German district on the first two singular vectors of election matrix $\boldsymbol{Y}$. (Left) Districts are colored according to the first party elected in each of them. (Right) Districts are colored according to the third party elected. This coloring reveals the historical East/West separation.

4.4 Deployed System

We deploy a Web platform, called Predikon[8], to provide real-time predictions for Swiss referenda (see Appendix A.1. 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.

4.4.1 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 [174]. 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 4.2 for Switzerland, which contains $R = 2196$

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

Table 4.3 – 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
Feb 9, 2020	More Affordable Housing	42.97	41.57	1.40
	Ban of Sexual Discrimination	63.03	62.95	0.08
Sep 27, 2020	Moderate Immigration	38.35	38.53	0.18
	Hunting Act	48.07	47.54	0.53
	Tax Deduction of Childcare Expenses	36.77	35.58	1.19
	Paternity Leave	60.27	59.33	0.94
	New Fighter Aircrafts	50.16	51.29	1.13
Nov 29, 2020	Responsible Businesses	50.73	50.13	0.60
	Ban on Financing War Material	42.55	41.91	0.64
Mar 7, 2021	Ban on Full Face Coverings	51.21	50.80	0.41
	e-ID Act	35.64	38.03	2.39
	Trade Agreement with Indonesia	51.65	51.54	0.11

municipalities. We predict the outcome of twelve items between February 9, 2020, and March 7, 2021. We summarize these items in Table 4.3. On average, the turnout is 53.3% and 2.8 million valid ballots are counted for each referendum.

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 $\boldsymbol{Y}_V$. For example, for February 9, 2020, we train the model using $V = 328$ votes and we predict the results of the two referenda on that date. 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 minute[9]. 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 $\boldsymbol{y}^{(\mathcal{U})}_{V+1} = \sigma(X^{(\mathcal{U})}\boldsymbol{w}_*)$. Similar to Equation (4.6), 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

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

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 $y^*_{\text{nat}} \in [0, 1]$.

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

4.4.2 Real-Time Predictions

In Figure 4.8, we show eight examples of the evolution of our predictions (red line), together with the weighted averaging (black line), as a function of the progress of the ballot counting. The ballot counting starts at 12pm and ends later in the afternoon, after all municipalities reported their results. Looking at the trajectory of the weighted averaging, the jumps occurring at several steps correspond to the publication of the results of whole cantons, such as Wallis, and of large municipalities, such as the cities of Basel, Genera, Bern or Zurich.

Our first predictions are made at 12:01pm, using the results of 355 municipalities on average (16.2% of all municipalities) representing 6.5% of the total population. For the twelve referenda, these predictions reach a mean absolute error of 0.8% to the true outcome. The largest error is made on the "e-ID Act", with a MAE of 2.39%. This could be due to the lack of historical votes related to digitalization in Switzerland leading to a vote embedding lying in an empty region of the ideological space. The weighted average for the current count varies up to a difference of 7.3%, whereas our predictions are qualitatively stable over time. To provide a robust estimation of the final outcome, our algorithm takes advantage of the correlation across municipalities and votes. Furthermore, our earliest predictions were always on the correct side of the 50% threshold, *i.e.*, it reached an accuracy of 100% at predicting the acceptance or rejection of a referenda.

4.5 Related Work

We base the present paper on the work of Etter et al. [54, 55]. Their approach consists in combining 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 (4.4) in our algorithm and illustrated in Figure 4.3: 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.

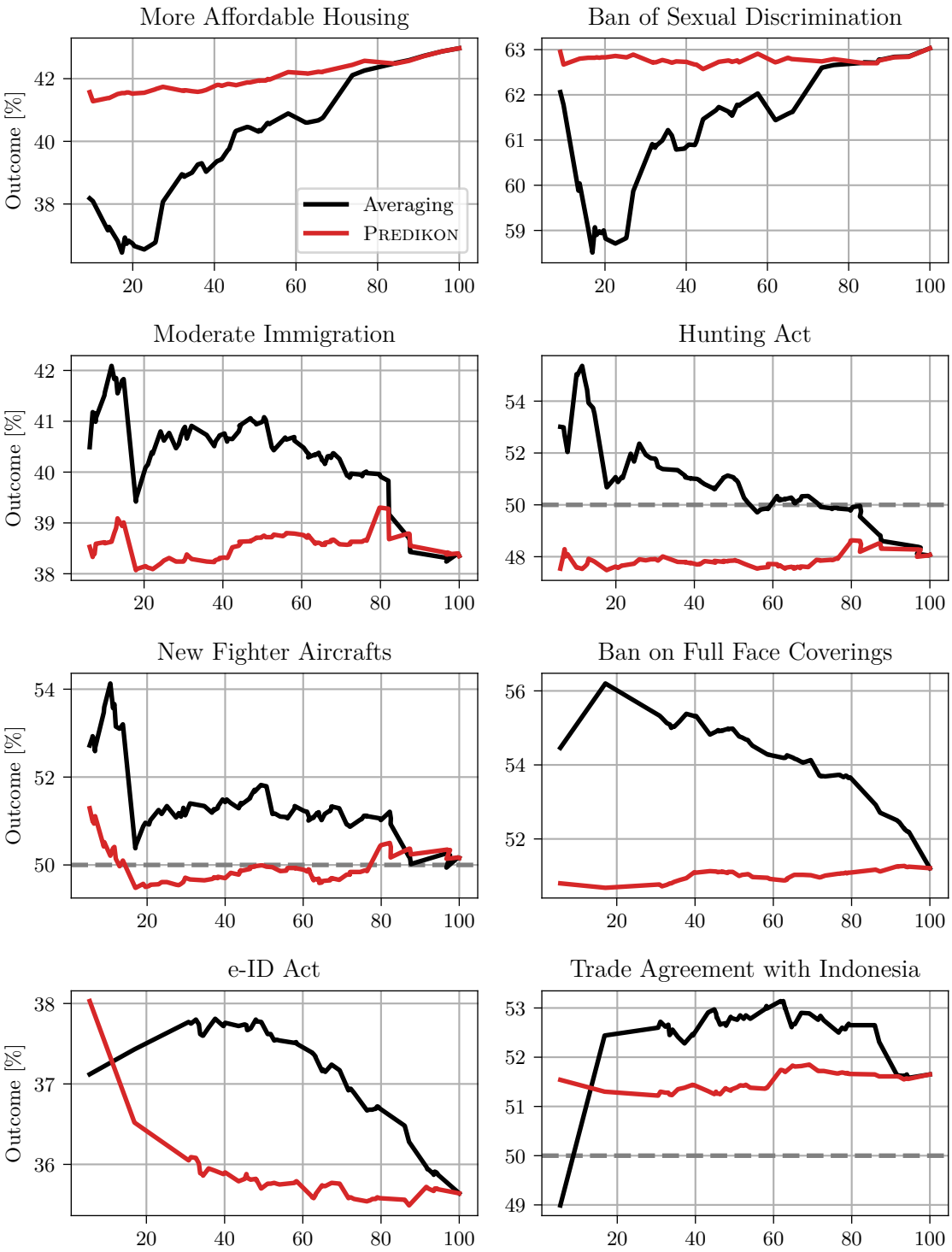


Figure 4.8 – Examples of evolution of predictions (red) and weighted averaging (black) on real, sequential data for the referenda between February 9, 2020, and March 7, 2021.

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 [182, 78]. 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. [12]. Incremental SVD revisions have been studied in the context of computer vision [20] and recommender systems [21]. 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 (4.7).

A whole body of work in the political science community exists on election forecasting [113], *i.e.*, predicting the outcome of an election before it happens. The seminal work of Bean [8], 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 [184], France [9], the U.K. [60], and the U.S. [158, 97]. 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 [16], and as he predicted Barack Obama's victory in the presidential election from polling data [167]. In the computer science community, algorithms for election forecasting have also been developed using social media data in Denmark [102], Finland [189], the U.S. [30, 150], and the developing world [51]. 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.

4.6 Summary

In this chapter, 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.

Perspective 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 [134]. Second, our algorithm is capable of making predictions only with at least one observed regional result. In the spirit of Etter et al. [55], 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.

5 Carbon Footprint Perception

In this chapter[1], we propose a statistical model to understand people's perception of their carbon footprint. Our model is inspired by the probit model of Thurstone [180] that we describe in Section 1.2. Driven by the observation that few people think of CO_2 impact in absolute terms, we design a system to probe people's perception from simple pairwise comparisons of the relative carbon footprint of their actions. The formulation of the model enables us to take an active-learning approach to selecting the pairs of actions that are maximally informative about the model parameters. We define a set of 18 actions and collect a dataset of 2183 comparisons from 176 users on a university campus by developing a Web platform[2]. The early results reveal promising directions to improve climate communication and enhance climate mitigation.

5.1 Introduction

To put the focus on actions that have high potential for emission reduction, we must first understand whether people have an accurate perception of the carbon footprint of these actions. If they do not, their efforts might be wasted. As an example, recent work by Wynes and Nicholas [200] shows that Canadian high-school textbooks encourage daily actions that yield negligible emission reduction. Actions with a higher potential of emission reduction are poorly documented. In this work, we model how people perceive the carbon footprint of their actions, which could guide educators and policy-makers.

In their daily life, consumers repeatedly face multiple options with varying environmental effects. Except for a handful of experts, no one is able to estimate the absolute quantity of CO_2 emitted by their actions of say, flying from Paris to London. Most people, however, are aware that taking the train for the same trip would release less CO_2. Hence, in the spirit of Thurstone [180] and Salganik and Levy [162] (among many others), we posit

[1] This chapter is based on Kristof et al. [103].
[2] The platform is accessible on http://www.climpact.ch.

that the perception of a population can be probed by simple pairwise comparisons. By doing so, we shift the complexity from the probing system to the model: Instead of asking difficult questions about each action and simply averaging the answers, we ask simple questions in the form of comparisons and design a non-trivial model to estimate the perception. *In fine*, human behaviour boils down to making choices: For example, we choose between eating local food and eating imported food; we do not choose between eating or not eating. Our awareness of relative emissions between actions (of the same purpose) is often sufficient to improve our carbon footprint.

Our contributions are as follows. First, we cast the problem of inferring a population's global perception from pairwise comparisons as a linear regression. Second, we adapt a well-known active-learning method to maximize the information gained from each comparison. We describe the model and the active-learning algorithm in Section 5.2. We design an interactive platform to collect real data for an experiment on our university campus, and we show early results in Section 5.3. Our approach could help climate scientists, sociologists, journalists, governments, and individuals improve climate communication and enhance climate mitigation.

5.2 Models

Let $\mathcal{A}$ be a set of M actions. For instance, "flying from London to New York" or "eating meat for a year" are both actions in $\mathcal{A}$. Let (i, j, y) be a triplet encoding that action $i \in \mathcal{A}$ has an *impact ratio* of $y \in \mathbf{R}_{>0}$ over action $j \in \mathcal{A}$. Said otherwise, if $y > 1$, action i has a carbon footprint y times *greater* than action j, and if $y < 1$, action i has a carbon footprint $1/y$ times *smaller* than action j.

Given some parameters $w_i, w_j \in \mathbf{R}$ representing the perceived (log-)carbon footprint in CO_2-equivalent of action i and action j, we posit

$$y = \frac{\exp w_i}{\exp w_j}.$$

We gather the parameters in a vector $\boldsymbol{w} \in \mathbf{R}^M$. Assuming a centered Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, $\sigma_n^2 \in \mathbf{R}$, we model the (log-)impact ratio

$$\log y = w_i - w_j + \epsilon = \boldsymbol{x}^\intercal \boldsymbol{w} + \epsilon, \tag{5.1}$$

where the comparison vector $\boldsymbol{x} \in \mathbf{R}^M$ is zero everywhere except in entry i where it is $+1$ and in entry j where it is -1. Vector $\boldsymbol{x}$ "selects" the pair of actions to compare. For a dataset $\mathcal{D} = \{(i_n, j_n, y_n) : n = 1, \ldots, N\}$ of N independent triplets and since

$\log y \sim \mathcal{N}(\boldsymbol{x}^\intercal \boldsymbol{w}, \sigma_n^2)$, the likelihood of the model is

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i^\intercal \boldsymbol{w}, \sigma_n^2) = \mathcal{N}(\boldsymbol{X}\boldsymbol{w}, \sigma_n^2 \boldsymbol{I}),$$

where $\boldsymbol{y} \in \mathbf{R}^N$ is the vector of observed (log-)impact ratios, and $\boldsymbol{X} \in \mathbf{R}^{N \times M}$ is a matrix of N comparison vectors.

We assume a Gaussian prior for the weight parameters $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_p)$, where $\boldsymbol{\mu} \in \mathbf{R}^M$ is the prior mean and $\boldsymbol{\Sigma}_p \in \mathbf{R}^{M \times M}$ is the prior covariance matrix. To obtain the global perceived carbon footprint of each action in $\mathcal{A}$ and to enable active learning, we compute the posterior distribution of the weight parameters given the data,

$$\begin{aligned} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{y}) &= \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})} \\ &= \mathcal{N}\left(\overline{\boldsymbol{w}} = \boldsymbol{\Sigma}\left(\sigma_n^{-2}\boldsymbol{X}^\intercal \boldsymbol{y} + \boldsymbol{\Sigma}_p^{-1}\boldsymbol{\mu}\right), \boldsymbol{\Sigma} = \left(\sigma_n^{-2}\boldsymbol{X}^\intercal \boldsymbol{X} + \boldsymbol{\Sigma}_p^{-1}\right)^{-1}\right). \end{aligned} \tag{5.2}$$

The noise variance σ_n^2, the prior mean $\boldsymbol{\mu}$, and the prior covariance matrix $\boldsymbol{\Sigma}_p$ are hyperparameters to be tuned. The global perceived carbon footprint is given by the posterior mean as $\exp \overline{\boldsymbol{w}}$. We use the posterior covariance matrix $\boldsymbol{\Sigma}$ to select the next pair of actions, as described in the following section.

Active Learning We collect the triplets in $\mathcal{D}$ from multiple users who take a quiz. During one session of the quiz, a user sequentially answers comparison questions and decides when to stop to see their overall results. Active learning enables us to maximize the information extracted from a session.

Let $\boldsymbol{\Sigma}_N$ and $\boldsymbol{\Sigma}_{N+1}$ be the covariance matrices of the posterior distribution in Equation (5.2) when N and $N+1$ comparisons have been respectively collected. Let $\boldsymbol{x}$ be the new $(N+1)$-th comparison vector, and recall that the entropy of a multivariate Gaussian distribution is given by

$$S = \frac{M}{2}(1 + \log 2\pi) + \frac{1}{2}\log\det \boldsymbol{\Sigma}. \tag{5.3}$$

As proposed by MacKay [116], we want to select the pair of actions to compare that is maximally informative about the values that the model parameters $\boldsymbol{w}$ should take [32, 87].

For our linear Gaussian model, this is obtained by maximizing the total information gain

$$\begin{aligned}\Delta S &= S_N - S_{N+1} \\ &= \frac{1}{2}\log\frac{\det \boldsymbol{\Sigma}_{N+1}^{-1}}{\det \boldsymbol{\Sigma}_N^{-1}} \\ &= \frac{1}{2}\log\frac{\det[\boldsymbol{\Sigma}_N^{-1} + \sigma_n^{-2}\boldsymbol{x}\boldsymbol{x}^\intercal]}{\det \boldsymbol{\Sigma}_N^{-1}} \qquad (5.4) \\ &= \frac{1}{2}\log\frac{(\det \boldsymbol{\Sigma}_N^{-1})(1 + \sigma_n^{-2}\boldsymbol{x}^\intercal\boldsymbol{\Sigma}_N\boldsymbol{x})}{\det \boldsymbol{\Sigma}_N^{-1}} \qquad (5.5) \\ &= \frac{1}{2}\log(1 + \sigma_n^{-2}\boldsymbol{x}^\intercal\boldsymbol{\Sigma}_N\boldsymbol{x}).\end{aligned}$$

We obtain (5.4) by observing that $\boldsymbol{\Sigma}_N^{-1} + \sigma_n^{-2}\boldsymbol{x}\boldsymbol{x}^\intercal = \sigma_n^{-2}\boldsymbol{X}^\intercal\boldsymbol{X} + \boldsymbol{\Sigma}_p^{-1} + \sigma_n^{-2}\boldsymbol{x}\boldsymbol{x}^\intercal = \boldsymbol{\Sigma}_{N+1}^{-1}$. We obtain (5.5) by the matrix determinant lemma.

Hence, to maximize ΔS, we maximize $\boldsymbol{x}^\intercal\boldsymbol{\Sigma}_N\boldsymbol{x}$ for all possible $\boldsymbol{x}$ in our dataset. Recall that comparison vectors $\boldsymbol{x}$ are zero everywhere except in entry i (+1) and in entry j (-1). By denoting $\boldsymbol{\Sigma}_N = [\sigma_{ij}^2]_{i,j=1}^M$, we seek, therefore, to find the pair of actions

$$(i^\star, j^\star) = \arg\max_{i,j}\left\{\sigma_{ii}^2 + \sigma_{jj}^2 - 2\sigma_{ij}^2\right\}.$$

The prior covariance matrix $\boldsymbol{\Sigma}_p$ could capture the prior knowledge about the typical user perception of relative carbon footprint. In future work, we intend to further reduce the number of questions asked during one session by a judicious choice of $\boldsymbol{\Sigma}_p$. In our experiments so far, we simply initialize it to a spherical covariance, as explained in the next section.

5.3 Experimental Results

Starting with no information at all, we arbitrarily set the prior noise $\sigma_n^2 = 1$ and the prior covariance matrix to a spherical covariance $\boldsymbol{\Sigma}_p = \sigma_p^2\boldsymbol{I}$, with $\sigma_p^2 = 10$. Our results are qualitatively robust to a large range of values for σ_p^2. In order to compare the perceived carbon footprint $\exp\overline{\boldsymbol{w}}$ with its true value $\exp\boldsymbol{v}$, we set the prior mean to $\boldsymbol{\mu} = c\mathbf{1}$, where $c = \frac{1}{M}\sum_{i=1}^M v_i$ is the mean of the (log-)true values. This guarantees that the perceived carbon footprint estimated from the model parameters have the same scale as the true values.

We compile a set $\mathcal{A}$ of $M = 18$ individual actions about transportation, food, and household (the full list of actions is provided in Appendix A.2.2). We deploy an online quiz[3] to collect pairwise comparisons of actions from real users on a university campus

[3] Accessible at `http://www.climpact.ch`

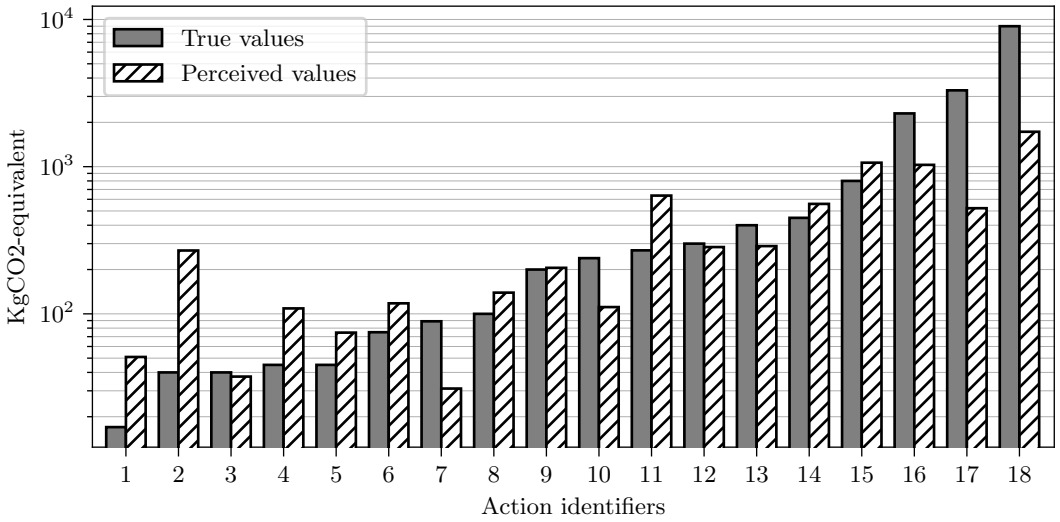


Figure 5.1 – Global perceived carbon footprint of 18 actions in kgCO_2-equivalent and their true values (log scale). The list of actions is provided in Appendix A.2.2.

(see Appendix A.2.1) We collect $N = 2183$ triplets from 176 users, mostly students between 16 and 25 years old. We show in Figure 5.1 the true carbon footprint, together with the global perception of the population, *i.e.*, the values $\exp \overline{w}_i$ for each action $i \in \mathcal{A}$.

The users in our population have a globally accurate perception. Among the actions showing the most discrepancy, the carbon footprint of short-haul flights is *overestimated* (Action 11), whereas the carbon footprint of long-haul flights (16) is highly *underestimated* (the scale is logarithmic). Similarly, the carbon footprint of first-class flights (18) is also *underestimated*. The users tend to *overestimate* the carbon footprint of more ecological transports, such as the train, the bus, and car-sharing (1, 4, and 6). The users have an accurate perception of actions related to diet (8, 14, and 15) and of actions related to domestic lighting (3 and 10). They *overestimate*, however, the carbon footprint of a dryer (2). Finally, they highly *underestimate* the carbon footprint of oil heating (17). Switzerland, where the users live, is one of the European countries whose consumption of oil for heating houses is the highest. There is, therefore, a high potential for raising awareness around this issue.

5.4 Summary

In this chapter, we proposed a statistical model for understanding people's global perception of their carbon footprint. The Bayesian formulation of the model enables us to take an active-learning approach to selecting the pairs of actions that maximize the gain of information. We deployed an online platform to collect real data from users.

The estimated perception of the users gives us insight into this population and reveals interesting directions for improving climate communication. In particular, we observed that the CO_2 emissions of actions with low carbon footprint tend to be *overestimated*, and actions with high carbon footprint tend to be *underestimated*.

Perspective Our model learns the overall perception of the whole population. This estimation lead to coarse results that may bias interpretation. We plan to enrich our model by replacing the global perception parameters $\boldsymbol{w}$ with parameters that depend on features of the users and of the actions. For example, the political views, income level, and region of residence of users might affect their perception. We also plan to collaborate with domain experts to further analyze people's estimated perception of their carbon footprint and to translate the conclusions of the results into concrete actions.

6 Dynamic Choices

Inspired by applications in sports where the skill of players or teams competing against each other varies over time, we propose in this chapter[1] a probabilistic model of pairwise-comparison outcomes that capture a wide range of time dynamics. We achieve this by replacing the static parameters of a class of popular pairwise-comparison models by continuous-time Gaussian processes; the covariance function of these processes enables expressive dynamics. We develop an efficient inference algorithm that computes an approximate Bayesian posterior distribution. Despite the flexibility of our model, our inference algorithm requires only a few linear-time iterations over the data and can take advantage of modern multiprocessor computer architectures. We apply our model to several historical databases[2] of sports outcomes and find that our approach (*a*) outperforms competing approaches in terms of predictive performance, (*b*) scales to millions of observations, and (*c*) generates compelling visualizations that help in understanding and interpreting the data. Finally, we deploy our algorithm on a Web platform[3] to predict the outcome of football matches in European leagues and international competitions.

6.1 Introduction

In many competitive sports and games (such as tennis, basketball, chess and electronic sports), the most useful definition of a competitor's skill is the propensity of that competitor to win against an opponent. It is often difficult to measure this skill *explicitly*: take basketball for example, a team's skill depends on the abilities of its players in terms of shooting accuracy, physical fitness, mental preparation, but also on the team's cohesion and coordination, on its strategy, on the enthusiasm of its fans, and a number of other intangible factors. However, it is easy to observe this skill *implicitly* through the outcomes of matches.

[1] This chapter is based on Maystre et al. [124].
[2] Data and code publicly available on https://github.com/lucasmaystre/kickscore-kdd19.
[3] The platform is accessible on https://kickoff.ai.

In this setting, probabilistic models of pairwise-comparison outcomes provide an elegant and practical approach to quantifying skill and to predicting future match outcomes given past data. These models, pioneered by Zermelo [206] in the context of chess (and by Thurstone [179] in the context of psychophysics), have been studied for almost a century. They posit that each competitor i (i.e., a team or player) is characterized by a latent score $s_i \in \mathbf{R}$ and that the outcome probabilities of a match between i and j are a function of the difference $s_i - s_j$ between their scores. By estimating the scores $\{s_i\}$ from data, we obtain an interpretable proxy for skill that is predictive of future match outcomes. If a competitor's skill is expected to remain stable over time, these models are very effective. But what if it varies over time?

A number of methods have been proposed to adapt comparison models to the case where scores change over time. Perhaps the best known such method is the Elo rating system [53], used by the World Chess Federation for their official rankings. In this case, the time dynamics are captured essentially as a by-product of the learning rule (c.f. Section 6.5). Other approaches attempt to model these dynamics explicitly [56, 66, 42, 41]. These methods greatly improve upon the static case when considering historical data, but they all assume the simplest model of time dynamics (that is, Brownian motion). Hence, they fail to capture more nuanced patterns such as variations at different timescales, linear trends, regression to the mean, discontinuities, and more.

In this work, we propose a new model of pairwise-comparison outcomes with expressive time-dynamics: it generalizes and extends previous approaches. We achieve this by treating the score of an opponent i as a time-varying Gaussian process $s_i(t)$ that can be endowed with flexible priors [153]. We also present an algorithm that, in spite of this increased flexibility, performs approximate Bayesian inference over the score processes in linear time in the number of observations so that our approach scales seamlessly to datasets with millions of observations. This inference algorithm addresses several shortcomings of previous methods: it can be parallelized effortlessly and accommodates different variational objectives. The highlights of our method are as follows.

Flexible Dynamics As scores are modeled by continuous-time Gaussian processes, complex (yet interpretable) dynamics can be expressed by composing covariance functions.

Generality The score of an opponent for a given match is expressed as a (sparse) linear combination of features. This enables, *e.g.*, the representation of a home advantage or any other contextual effect. Furthermore, the model encompasses a variety of observation likelihoods beyond win / lose, based, *e.g.*, on the number of points a competitor scores.

Bayesian Inference Our inference algorithm returns a posterior *distribution* over score processes. This leads to better predictive performance and enables a principled way

to learn the dynamics (and any other model hyperparameters) by optimizing the log-marginal likelihood of the data.

Ease of Intepretation By plotting the score processes $\{s_i(t)\}$ over time, it is easy to visualize the probability of any comparison outcome under the model. As the time dynamics are described through the composition of simple covariance functions, their interpretation is straightforward as well.

Concretely, our contributions are threefold. First, we develop a probabilistic model of pairwise-comparison outcomes with flexible time-dynamics (Section 6.2). The model covers a wide range of use cases, as it enables (a) opponents to be represented by a sparse linear combination of features, and (b) observations to follow various likelihood functions. In fact, it unifies and extends a large body of prior work. Second, we derive an efficient algorithm for approximate Bayesian inference (Section 6.3). This algorithm adapts to two different variational objectives; in conjunction with the "reverse-KL" objective, it provably converges to the optimal posterior approximation. It can be parallelized easily, and the most computationally intensive step can be offloaded to optimized off-the-shelf numerical software. Third, we apply our method on several sports datasets and show that it achieves state-of-the-art predictive performance (Section 6.4). Our results highlight that different sports are best modeled with different time-dynamics. We also demonstrate how domain-specific and contextual information can improve performance even further; in particular, we show that our model outperforms competing ones even when there are strong intransitivities in the data.

In addition to prediction tasks, our model can also be used to generate compelling visualizations of the temporal evolution of skills. All in all, we believe that our method will be useful to data-mining practitioners interested in understanding comparison time-series and in building predictive systems for games and sports. Our algorithm is deployed on the Kickoff.ai[4] platform to provide predictions of football matches is European leagues (see Appendix A.3).

A Note on Extensions In this chapter, we focus on *pairwise* comparisons for conciseness. However, the model and inference algorithm could be extended to multiway comparisons or partial rankings over small sets of opponents without any major conceptual change, similarly to Herbrich et al. [82]. Furthermore, and even though we develop our model in the context of sports, it is relevant to all applications of ranking from comparisons, *e.g.*, to those where comparison outcomes reflect human preferences or opinions [179, 126, 162].

[4] https://kickoff.ai

6.2 Model

In this section, we formally introduce our probabilistic model, called *Kickscore*. For clarity, we take a clean-slate approach and develop the model from scratch. We discuss in more detail how it relates to prior work in Section 6.5.

The basic building blocks of Kickscore are *features*[5]. Let M be the number of features; each feature $m \in [M]$ is characterized by a latent, continuous-time Gaussian process

$$s_m(t) \sim \mathrm{GP}[0, k_m(t, t')]. \tag{6.1}$$

We call $s_m(t)$ the *score process* of m, or simply its *score*. The *covariance function* of the process, $k_m(t, t') \coloneqq \mathbf{E}\left[s_m(t)s_m(t')\right]$, is used to encode time dynamics. A brief introduction to Gaussian processes as well as a discussion of useful covariance functions is given in Section 6.2.1. The M scores $s_1(t), \ldots, s_M(t)$ are assumed to be (a priori) jointly independent, and we collect them into the *score vector*

$$\boldsymbol{s}(t) = \begin{bmatrix} s_1(t) & \cdots & s_M(t) \end{bmatrix}^\intercal.$$

For a given match, each opponent i is described by a sparse linear combination of the features, with coefficients $\boldsymbol{x}_i \in \mathbf{R}^M$. That is, the score of an opponent i at time t^* is given by

$$s_i \coloneqq \boldsymbol{x}_i^\intercal \boldsymbol{s}(t^*). \tag{6.2}$$

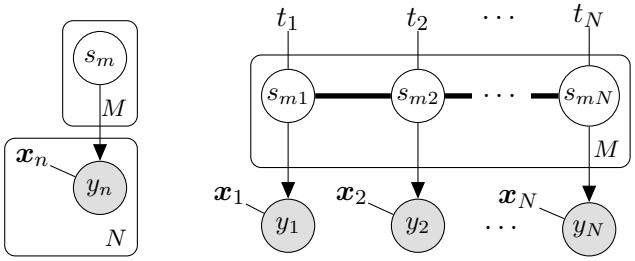
In the case of a one-to-one mapping between competitors and features, $\boldsymbol{x}_i$ is simply the one-hot encoding of opponent i. More complex setups are possible: For example, in the case of team sports and if the player lineup is available for each match, it could also be used to encode the players taking part in the match [123]. Note that $\boldsymbol{x}_i$ can also depend contextually on the match. For instance, it can be used to encode the fact that a team plays at home [4].

Each observation consists of a tuple $(\boldsymbol{x}_i, \boldsymbol{x}_j, t^*, y)$, where $\boldsymbol{x}_i, \boldsymbol{x}_j$ are the opponents' feature vectors, $t^* \in \mathbf{R}$ is the time, and $y \in \mathcal{Y}$ is the match outcome. We posit that this outcome is a random variable that depends on the opponents through their latent score difference:

$$y \mid \boldsymbol{x}_i, \boldsymbol{x}_j, t^* \sim p(y \mid s_i - s_j),$$

where p is a known probability density (or mass) function and s_i, s_j are given by (6.2). The idea of modeling outcome probabilities through score differences dates back to Thurstone

[5] In the simplest case, there is a one-to-one mapping between competitors (e.g., teams) and features, but decoupling them offers increased modeling power.



(a) Static model (b) Our dynamic model

Figure 6.1 – Graphical representation of a static model (left) and of the dynamic model presented in this chapter (right). The observed variables are shaded. For conciseness, we let $\boldsymbol{x}_n \coloneqq \boldsymbol{x}_{n,i} - \boldsymbol{x}_{n,j}$. Right: the latent score variables are mutually dependent across time, as indicated by the thick line.

Table 6.1 – Examples of observation likelihoods. The score difference is denoted by $d \coloneqq s_i - s_j$ and the Gaussian cumulative density function is denoted by Φ.

Name	$\mathcal{Y}$	$p(y \mid d)$	References
Probit	$\{\pm 1\}$	$\Phi(yd)$	[179, 82]
Logit	$\{\pm 1\}$	$[1 + \exp(-yd)]^{-1}$	[206, 19]
Ordinal probit	$\{\pm 1, 0\}$	$\Phi(yd - \alpha), \ldots$	[64]
Poisson-exp	$\mathbf{N}_{\geq 0}$	$\exp(yd - e^d)/y!$	[117]
Gaussian	$\mathbf{R}$	$\propto \exp[(y-d)^2/(2\sigma^2)]$	[72]

[179] and Zermelo [206]. The likelihood p is chosen such that positive values of $s_i - s_j$ lead to successful outcomes for opponent i and vice-versa.

A graphical representation of the model is provided in Figure 6.1. For perspective, we also include the representation of a static model, such as that of Thurstone [179]. Our model can be interpreted as "conditionally parametric": conditioned on a particular time, it falls back to a (static) pairwise-comparison model parametrized by real-valued scores.

Observation Models Choosing an appropriate likelihood function $p(y \mid s_i - s_j)$ is an important modeling decision and depends on the information contained in the outcome y. The most widely applicable likelihoods require only *ordinal* observations, *i.e.*, whether a match resulted in a win or a loss (or a tie, if applicable). In some cases, we might additionally observe points (e.g., in association football, the number of goals scored by each team). To make use of this extra information, we can model (*a*) the number of points of opponent i with a Poisson distribution whose rate is a function of $s_i - s_j$, or (*b*) the points difference with a Gaussian distribution centered at $s_i - s_j$. A non-exhaustive list of likelihoods is given in Table 6.1.

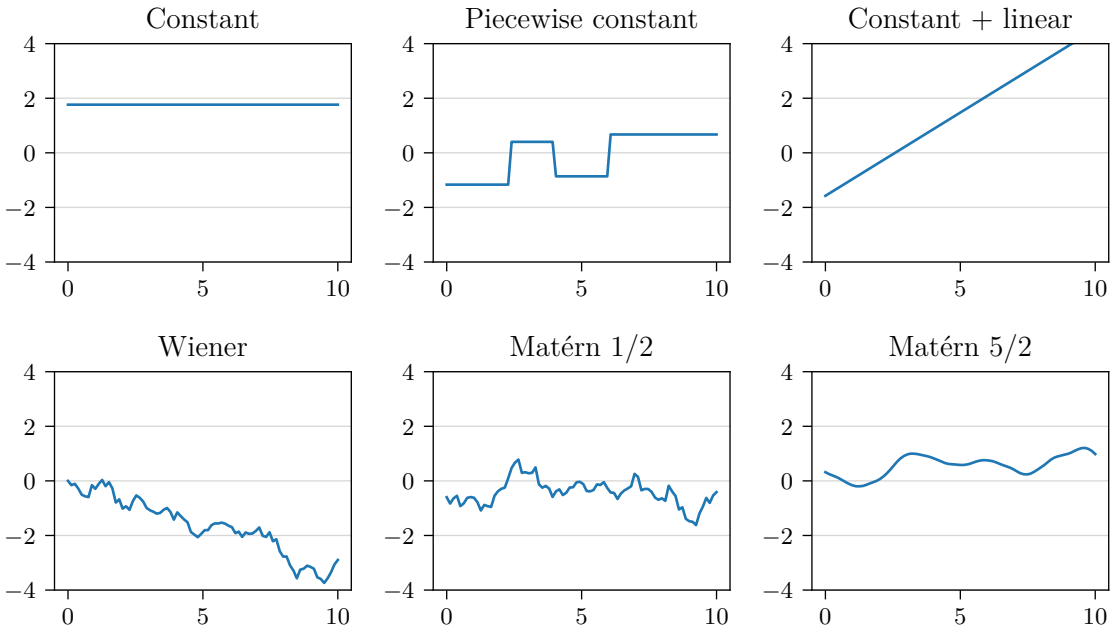


Figure 6.2 – Random realizations of a zero-mean Gaussian process with six different covariance functions.

6.2.1 Covariance Functions

A Gaussian process $s(t) \sim \mathrm{GP}[0, k(t, t')]$ can be thought of as an infinite collection of random variables indexed by time, such that the joint distribution of any finite vector of N samples $\boldsymbol{s} = [s(t_1) \cdots s(t_N)]$ is given by $\boldsymbol{s} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{K})$, where $\boldsymbol{K} = [k(t_i, t_j)]$. That is, $\boldsymbol{s}$ is jointly Gaussian with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{K}$. We refer the reader to Rasmussen and Williams [153] for an excellent introduction to Gaussian processes.

Hence, by specifying the covariance function appropriately, we can express prior expectations about the time dynamics of a feature's score, such as smooth or non-smooth variations at different timescales, regression to the mean, discontinuities, linear trends and more. Here, we describe a few functions that we find useful in the context of modeling temporal variations. Figure 6.2 illustrates these functions through random realizations of the corresponding Gaussian processes.

Constant This covariance captures processes that remain constant over time. It is useful in composite covariances to model a constant offset (i.e., a mean score value).

Piecewise Constant Given a partition of $\mathbf{R}$ into disjoint intervals, this covariance is constant inside a partition and zero between partitions. It can, for instance, capture discontinuities across seasons in professional sports leagues.

Wiener This covariance reflects Brownian motion dynamics (c.f. Section 6.5). It is non-stationary: the corresponding process drifts away from 0 as t grows.

Matérn This family of stationary covariance functions can represent smooth and non-smooth variations at various timescales. It is parametrized by a variance, a characteristic timescale and a smoothness parameter ν. When $\nu = 1/2$, it corresponds to a mean-reverting version of Brownian motion.

Linear This covariance captures linear dynamics.

Finally, note that composite functions can be created by adding or multiplying covariance functions together. For example, let k_a and k_b be constant and Matérn covariance functions, respectively. Then, the composite covariance $k(t, t') \coloneqq k_a(t, t') + k_b(t, t')$ captures dynamics that fluctuate around a (non-zero) mean value. Duvenaud [50, Section 2.3] provides a good introduction to building expressive covariance functions by composing simple ones.

6.3 Inference Algorithm

In this section, we derive an efficient inference algorithm for our model. For brevity, we focus on explaining the main ideas behind the algorithm. A reference software implementation, available online at `https://github.com/lucasmaystre/kickscore`, complements the description provided here.

We begin by introducing some notation. Let $\mathcal{D} = \{(\boldsymbol{x}_n, t_n, y_n) : n \in [N]\}$ be a dataset of N independent observations, where for conciseness we fold the two opponents $\boldsymbol{x}_{n,i}$ and $\boldsymbol{x}_{n,j}$ into $\boldsymbol{x}_n \coloneqq \boldsymbol{x}_{n,i} - \boldsymbol{x}_{n,j}$, for each observation[6]. Let $\mathcal{D}_m \subseteq [N]$ be the subset of observations involving feature m, *i.e.*, those observations for which $x_{nm} \neq 0$, and let $N_m = |\mathcal{D}_m|$. Finally, denote by $\boldsymbol{s}_m \in \mathbf{R}^{N_m}$ the samples of the latent score process at times corresponding to the observations in $\mathcal{D}_m$. The joint prior distribution of these samples is $p(\boldsymbol{s}_m) = \mathrm{N}(\mathbf{0}, \boldsymbol{K}_m)$, where $\boldsymbol{K}_m$ is formed by evaluating the covariance function $k_m(t, t')$ at the relevant times.

We take a Bayesian approach and seek to compute the posterior distribution

$$p(\boldsymbol{s}_1, \dots, \boldsymbol{s}_M \mid \mathcal{D}) \propto \prod_{m=1}^{M} p(\boldsymbol{s}_m) \prod_{n=1}^{N} p[y_n \mid \boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)]. \tag{6.3}$$

As the scores are coupled through the observations, the posterior no longer factorizes over $\{\boldsymbol{s}_m\}$. Furthermore, computing the posterior is intractable if the likelihood is non-Gaussian.

To overcome these challenges, we consider a mean-field variational approximation [191]. In particular, we assume that the posterior can be well-approximated by a multivariate

[6] This enables us to write the score difference more compactly. Given an observation at time t^* and letting $\boldsymbol{x} \coloneqq \boldsymbol{x}_i - \boldsymbol{x}_j$, we have $s_i - s_j = \boldsymbol{x}_i^\intercal \boldsymbol{s}(t^*) - \boldsymbol{x}_j^\intercal \boldsymbol{s}(t^*) = \boldsymbol{x}^\intercal \boldsymbol{s}(t^*)$.

Gaussian distribution that factorizes over the features:

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M \mid \mathcal{D}) \approx q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \coloneqq \prod_{m=1}^{M} \mathrm{N}(\boldsymbol{s}_m \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m). \tag{6.4}$$

Computing this approximate posterior amounts to finding the variational parameters $\{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ that best approximate the true posterior. More formally, the inference problem reduces to the optimization problem

$$\min_{\{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}} \mathrm{div}\left[p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M \mid \mathcal{D}) \parallel q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M)\right], \tag{6.5}$$

for some divergence measure $\mathrm{div}(p\|q) \geq 0$. We will consider two different such measures in Section 6.3.1.

A different viewpoint on the approximate posterior is as follows. For both of the variational objectives that we consider, it is possible to rewrite the optimal distribution $q(\boldsymbol{s}_m)$ as

$$q(\boldsymbol{s}_m) \propto p(\boldsymbol{s}_m) \prod_{n \in \mathcal{D}_m} \mathrm{N}(s_{mn} \mid \tilde{\mu}_{mn}, \tilde{\sigma}^2_{mn}).$$

Letting $\mathcal{X}_n \subseteq [M]$ be the subset of features such that $x_{nm} \neq 0$, we can now reinterpret the variational approximation as transforming every observation $(\boldsymbol{x}_n, t_n, y_n)$ into several independent *pseudo-observations* with Gaussian likelihood, one for each feature $m \in \mathcal{X}_n$. Instead of optimizing directly $\{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ in (6.5), we can alternatively choose to optimize the parameters $\{\tilde{\mu}_{mn}, \tilde{\sigma}^2_{mn}\}$. For any feature m, given the pseudo-observations' parameters $\tilde{\boldsymbol{\mu}}_m$ and $\tilde{\boldsymbol{\sigma}}^2_m$, computing $q(\boldsymbol{s}_m)$ becomes tractable (c.f. Section 6.3.2).

An outline of our iterative inference procedure is given in Algorithm 6.1. Every iteration consists of two steps:

1. updating the pseudo-observations' parameters given the true observations and the current approximate posterior (lines 4–7), and

2. recomputing the approximate posterior given the current pseudo-observation (lines 8 and 9).

Convergence is declared when the difference between two successive iterates of $\{\tilde{\mu}_{mn}\}$ and $\{\tilde{\sigma}^2_{mn}\}$ falls below a threshold. Note that, as a by-product of the computations performed by the algorithm, we can also estimate the log-marginal likelihood of the data, $\log p(\mathcal{D})$.

Running Time In Section 6.3.1, we show that Derivatives and UpdateParams run in constant time. In Section 6.3.2, we show that UpdatePosterior runs in time $O(N_m)$. Therefore, if we assume that the vectors $\{\boldsymbol{x}_n\}$ are sparse, the total running time

Algorithm 6.1 Model inference.

Require: $\mathcal{D} = \{(\boldsymbol{x}_n, t_n, y_n) : n \in [N]\}$
1: $\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\sigma}}_m^2 \leftarrow \mathbf{0}, \boldsymbol{\infty} \quad \forall m$
2: $q(\boldsymbol{s}_m) \leftarrow p(\boldsymbol{s}_m) \quad \forall m$
3: **repeat**
4: **for** $n = 1, \ldots, N$ **do**
5: $\boldsymbol{\delta} \leftarrow \textsc{Derivatives}(\boldsymbol{x}_n, y_n)$
6: **for** $m \in \mathcal{X}_n$ **do**
7: $\tilde{\mu}_{mn}, \tilde{\sigma}_{mn}^2 \leftarrow \textsc{UpdateParams}(x_{nm}, \boldsymbol{\delta})$
8: **for** $m = 1, \ldots, M$ **do**
9: $q(\boldsymbol{s}_m) \leftarrow \textsc{UpdatePosterior}(\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\sigma}}_m^2)$
10: **until** convergence

per iteration of Algorithm 6.1 is $O(N)$. Furthermore, each of the two outer *for* loops (lines 4 and 8) can be parallelized easily, leading in most cases to a linear acceleration with the number of available processors.

6.3.1 Updating the Pseudo-Observations

The exact computations performed during the first step of the inference algorithm—updating the pseudo-observations—depend on the specific variational method used. We consider two: expectation propagation [131], and reverse-KL variational inference [15]. The ability of Algorithm 6.1 to seamlessly adapt to either of the two methods is valuable, as it enables practitioners to use the most advantageous method for a given likelihood function.

Expectation Propagation

We begin by defining two distributions. The *cavity* distribution q_{-n} is the approximate posterior without the pseudo-observations associated with the nth datum, that is,

$$q_{-n}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \propto \frac{q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M)}{\prod_{m \in \mathcal{X}_n} \mathrm{N}(s_{mn} \mid \tilde{\mu}_{mn}, \tilde{\sigma}_{mn}^2)}.$$

The *hybrid* distribution $\hat{q}_n$ is given by the cavity distribution multiplied by the nth likelihood factor, *i.e.*,

$$\hat{q}_n(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \propto q_{-n}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) p[y_n \mid \boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)].$$

Informally, the hybrid distribution $\hat{q}_n$ is "closer" to the true distribution than q.

Expectation propagation (EP) works as follows. At each iteration and for each n, we update the parameters $\{\tilde{\mu}_{mn}, \tilde{\sigma}_{mn} : m \in \mathcal{X}_n\}$ such that $\mathrm{KL}(\hat{q}_n \| q)$ is minimized. To this

end, the function Derivatives (on line 5 of Algorithm 6.1) computes the first and second derivatives of the log-partition function

$$\log \mathbf{E}_{q_{-n}} \{p[y_n \mid \boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)]\} \tag{6.6}$$

with respect to $\mu_{-n} \coloneqq \mathbf{E}_{q_{-n}}[\boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)]$. These computations can be done in closed form for the widely-used probit likelihood, and they involve one-dimensional numerical integration for most other likelihoods. EP has been reported to result in more accurate posterior approximations on certain classification tasks [137].

Reverse KL Divergence

This method (often referred to simply as *variational inference* in the literature) seeks to minimize $\mathrm{KL}(q\|p)$, *i.e.*, the KL divergence from the approximate posterior q to the true posterior p.

To optimize this objective, we adopt the approach of Khan and Lin [98]. In this case, the function Derivatives computes the first and second derivatives of the expected log-likelihood

$$\mathbf{E}_q \{\log p[y_n \mid \boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)]\} \tag{6.7}$$

with respect to $\mu \coloneqq \mathbf{E}_q[\boldsymbol{x}_n^\intercal \boldsymbol{s}(t_n)]$. These computations involve numerically solving two one-dimensional integrals.

In comparison to EP, this method has two advantages. The first is theoretical: If the likelihood $p(y \mid d)$ is log-concave in d, then the variational objective has a unique global minimum, and we can guarantee that Algorithm 6.1 converges to this minimum [98]. The second is numerical: Excepted for the probit likelihood, computing (6.7) is numerically more stable than computing (6.6).

6.3.2 Updating the Approximate Posterior

The second step of Algorithm 6.1 (lines 8 and 9) solves the following problem, for every feature m. Given Gaussian pseudo-observations $\{\tilde{\mu}_{mn}, \tilde{\sigma}_{mn} : n \in \mathcal{D}_m\}$ and a Gaussian prior $p(\boldsymbol{s}_m) = \mathrm{N}(\mathbf{0}, \boldsymbol{K}_m)$, compute the posterior

$$q(\boldsymbol{s}_m) \propto p(\boldsymbol{s}_m) \prod_{n \in \mathcal{D}_m} \mathrm{N}(s_{mn} \mid \tilde{\mu}_{mn}, \tilde{\sigma}_{mn}^2).$$

This computation can be done independently and in parallel for each feature $m \in [M]$.

A naive approach is to use the self-conjugacy properties of the Gaussian distribution directly. Collecting the parameters of the pseudo-observations into a vector $\tilde{\boldsymbol{\mu}}_m$ and a

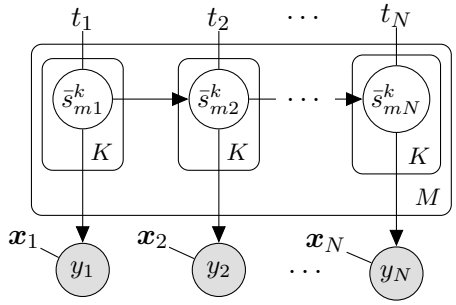


Figure 6.3 – State-space reformulation of our model. With respect to the representation in Figure 6.1b, the number of latent variables has increased, but they now form a Markov chain.

diagonal matrix $\tilde{\boldsymbol{\Sigma}}_m$, the parameters of the posterior $q(\boldsymbol{s}_m)$ are given by

$$\boldsymbol{\Sigma}_m = (\boldsymbol{K}_m^{-1} + \tilde{\boldsymbol{\Sigma}}_m^{-1})^{-1}, \qquad \boldsymbol{\mu}_m = \boldsymbol{\Sigma}_m \tilde{\boldsymbol{\Sigma}}_m^{-1} \tilde{\boldsymbol{\mu}}_m. \tag{6.8}$$

Unfortunately, this computation runs in time $O(N_m^3)$, a cost that becomes prohibitive if some features appear in many observations.

Instead, we use an alternative approach that exploits a link between temporal Gaussian processes and state-space models [76, 154]. Without loss of generality, we now assume that the N observations are ordered chronologically, and, for conciseness, we drop the feature's index and consider a single process $s(t)$. The key idea is to augment $s(t)$ into a K-dimensional vector-valued Gauss-Markov process $\bar{\boldsymbol{s}}(t)$, such that

$$\bar{\boldsymbol{s}}(t_{n+1}) = \boldsymbol{A}_n \bar{\boldsymbol{s}}(t_n) + \boldsymbol{\varepsilon}_n, \qquad \boldsymbol{\varepsilon}_n \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{Q}_n)$$

where $K \in \mathbf{N}_{>0}$ and $\boldsymbol{A}_n, \boldsymbol{Q}_n \in \mathbf{R}^{K \times K}$ depend on the time interval $|t_{n+1} - t_n|$ and on the covariance function $k(t, t')$ of the original process $s(t)$. The original (scalar-valued) and the augmented (vector-valued) processes are related through the equation

$$s(t) = \boldsymbol{h}^\intercal \bar{\boldsymbol{s}}(t),$$

where $\boldsymbol{h} \in \mathbf{R}^K$ is called the *measurement vector*.

Figure 6.3 illustrates our model from a state-space viewpoint. It is important to note that the mutual time dependencies of Figure 6.1b have been replaced by Markovian dependencies. In this state-space formulation, posterior inference can be done in time $O(K^3 N)$ by using the Rauch–Tung–Striebel smoother [175].

From Covariance Functions to State-Space Models A method for converting a process $s(t) \sim \mathrm{GP}[\boldsymbol{0}, k(t, t')]$ into an equivalent Gauss-Markov process $\bar{\boldsymbol{s}}(t)$ by explicit construction of $\boldsymbol{h}$, $\{\boldsymbol{A}_n\}$ and $\{\boldsymbol{Q}_n\}$ is given in Solin [168]. All the covariance functions

described in Section 6.2.1 lead to exact state-space reformulations of order $K \leq 3$. The composition of covariance functions through addition or multiplication can also be treated exactly and automatically. Some other covariance functions, such as the squared-exponential function or periodic functions [153], cannot be transformed exactly but can be approximated effectively and to arbitrary accuracy [76, 169].

Finally, we stress that the state-space viewpoint is useful because it leads to a faster inference procedure; but defining the time dynamics of the score processes in terms of covariance functions is much more intuitive.

6.3.3 Predicting at a New Time

Given the approximate posterior $q(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M)$, the probability of observing outcome y at a new time t^* given the feature vector $\boldsymbol{x}$ is given by

$$p(y \mid \boldsymbol{x}, t^*) = \int_{\mathbf{R}} p(y \mid z) p(z) dz,$$

where $z = \boldsymbol{x}^\intercal \boldsymbol{s}(t^*)$ and the distribution of $s_m(t^*)$ is derived from the posterior $q(\boldsymbol{s}_m)$. By using the state-space formulation of the model, the prediction can be done in constant time [160].

6.4 Experimental Results

In this section, we evaluate our model and inference algorithm on real data. Our experiments cover three aspects. First, in Section 6.4.1, we compare the predictive performance of our model against competing approaches, focusing on the impact of flexible time-dynamics. Second, in Section 6.4.2, we show that by carefully choosing features and observation likelihoods, predictive performance can be improved significantly. Finally, in Section 6.4.3, we study various facets of our inference algorithm. We measure the impact of the mean-field assumption and of the choice of variational objective, and we demonstrate the scalability of the algorithm.

Datasets We consider six datasets of pairwise-comparison outcomes of various sports and games. Four of them contain timestamped outcomes; they relate to tennis, basketball, association football and chess. Due to the large size of the chess dataset[7], we also consider a subset of the data spanning 30 years. The two remaining datasets contain match outcomes of the StarCraft computer game and do not have timestamps. Table 6.2

[7] This dataset consists of all the match outcomes contained in *ChessBase Big Database 2018*, available at `https://shop.chessbase.com/en/products/big_database_2018`.

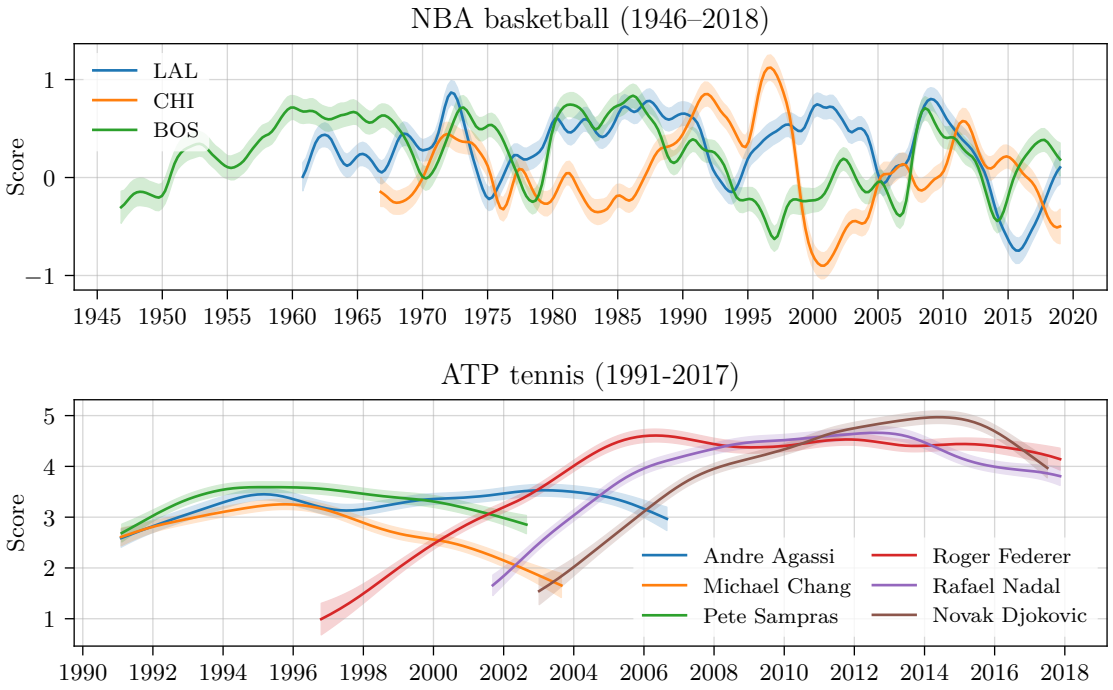


Figure 6.4 – Temporal evolution of the score processes ($\mu \pm \sigma$) corresponding to selected basketball teams (top) and tennis players (bottom). The basketball teams are the Los Angeles Lakers (LAL), the Chicago Bulls (CHI) and the Boston Celtics (BOS).

provides summary statistics for all the datasets. Except for chess, all data are publicly available online[8].

Performance Metrics Let $(\boldsymbol{x}, t^*, y)$ be an observation. We measure performance by using the logarithmic loss: $-\log p(y \mid \boldsymbol{x}, t^*)$ and the accuracy: $\mathbf{1}_{\{y=\arg\max_{y'} p(y'|\boldsymbol{x},t^*)\}}$. We report their average values on the test set.

Methodology Unless specified otherwise, we partition every dataset into a training set containing the first 70% of the observations and a test set containing the remaining 30%, in chronological order. The various hyperparameters (such as covariance functions and their parameters, learning rates, etc.) are selected based on the training data only, by maximizing the log-marginal likelihood of Bayesian models and by minimizing the average leave-one-out log loss otherwise. In order to predict the outcome of an observation at time t^*, we use *all* the data (in both training and test sets) up to the day preceding t^*. This closely mimics the setting where a predictor must guess the outcome of an event in the near future based on all past data. Unless specified otherwise, we use Algorithm 6.1

[8]Tennis: `https://github.com/JeffSackmann/tennis_atp`, basketball: `https://projects.fivethirtyeight.com/nba-model/nba_elo.csv`, football: `https://int.soccerway.com/`, StarCraft: `https://github.com/csinpi/blade_chest`.

Table 6.2 – Summary statistics of the sports datasets.

Name	Ties	M	N	Time span
ATP tennis	No	20 046	618 934	1991–2017
NBA basketball	No	102	67 642	1946–2018
World football	Yes	235	19 158	1908–2018
ChessBase small	Yes	19 788	306 764	1950–1980
ChessBase full	Yes	343 668	7 169 202	1475–2017
StarCraft WoL	No	4381	61 657	—
StarCraft HotS	No	2287	28 582	—

with the EP variational objective, and we declare convergence when the improvement in log-marginal likelihood falls below 10^{-3}. Typically, the algorithm converges in less than a hundred iterations.

6.4.1 Flexible Time-Dynamics

In this experiment, we compare the predictive performance of our model against competing approaches on four timestamped datasets. In order to better isolate and understand the impact of accurately modeling *time dynamics* on predictive performance, we keep the remaining modeling choices simple: we treat all outcomes as ordinal-valued (i.e., *win*, *loss* and possibly *tie*) with a probit likelihood and use a one-to-one mapping between competitors and features. In Table 6.3, we report results for the following models:

- *Random*. This baseline assigns equal probability to every outcome.
- *Constant*. The model of Section 6.2 with a constant covariance function. This model assumes that the scores do not vary over time.
- *Elo*. The system used by the World Chess Federation [53]. Time dynamics are a by-product of the update rule (c.f. Section 6.5).
- *TrueSkill*. The Bayesian model of Herbrich et al. [82]. Time dynamics are assumed to follow Brownian motion (akin to our Wiener kernel) and inference is done in a single pass over the data.
- *Kickscore*. The model of Section 6.2. We try multiple covariance functions and report the one that maximizes the log-marginal likelihood.

Our model matches or outperforms other approaches in almost all cases, both in terms of log loss and in terms of accuracy. Interestingly, different datasets are best modeled by using different covariance functions, perhaps capturing underlying skill dynamics specific to each sport.

Table 6.3 – Predictive performance of our model and of competing approaches on four datasets, in terms of average log loss and average accuracy. The best result is indicated in bold.

	Random		Constant		Elo		TrueSkill		Kickscore	
Dataset	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Loss	Acc.	Covariance
ATP tennis	0.693	0.500	0.581	0.689	0.563	0.705	0.563	0.705	**0.552**	**0.714**	Affine + Wiener
NBA basketball	0.693	0.500	0.692	0.536	0.634	0.644	0.634	0.644	**0.630**	**0.645**	Constant + Matérn 1/2
World football	1.099	0.333	0.929	**0.558**	0.950	0.551	0.937	0.554	**0.926**	**0.558**	Constant + Matérn 1/2
ChessBase small	1.099	0.333	1.030	**0.478**	1.035	0.447	1.030	0.467	**1.026**	0.474	Constant + Wiener

Table 6.4 – Average predictive log loss of models with different observation likelihoods. The best result is indicated in bold.

Dataset	Probit	Logit	Gaussian	Poisson
NBA basketball	0.630	0.630	**0.627**	0.630
World football	0.926	0.926	0.927	**0.922**

Visualizing and Interpreting Scores Figure 6.4 displays the temporal evolution of the score of selected basketball teams and tennis players. In the basketball case, we can recognize the dominance of the Boston Celtics in the early 1960's and the Chicago Bulls' strong 1995-96 season. In the tennis case, we can see the progression of a new generation of tennis champions at the turn of the 21st century. Plotting scores over time provides an effective way to compactly represent the history of a given sport. Analyzing the optimal hyperparameters is also insightful: the characteristic timescale of the dynamic covariance component is 1.75 and 7.47 years for basketball and tennis, respectively. The score of basketball teams appears to be much more volatile.

6.4.2 Generality of the Model

In this section, we demonstrate how we can take advantage of additional modeling options to further improve predictive performance. In particular, we show that choosing an appropriate likelihood and parametrizing opponents with match-dependent combinations of features can bring substantial gains.

Observation Models

Basketball and football match outcomes actually consist of *points* (respectively, goals) scored by each team during the match. We can make use of this additional information to improve predictions [117]. For each of the basketball and football datasets, we compare the best model obtained in Section 6.4.1 to alternative models. These alternative models keep the same time dynamics but use either

1. a logit likelihood on the ordinal outcome,
2. a Gaussian likelihood on the points difference, or
3. a Poisson-exp likelihood on the points scored by each team.

The results are presented in Table 6.4. The logit likelihood performs similarly to the probit one [170], but likelihoods that take points into account can indeed lead to better predictions.

Table 6.5 – Predictive performance of models with a home or first-mover advantage in comparison to models without.

	Basic		Advantage	
Dataset	Loss	Acc.	Loss	Acc.
World football	0.926	0.558	**0.900**	**0.579**
ChessBase small	1.026	0.480	**1.019**	**0.485**

Match-Dependent Parametrization

For a given match, we can represent opponents by using (non-trivial) linear combinations of features. This enables, *e.g.*, to represent context-specific information that might influence the outcome probabilities. In the case of football, for example, it is well-known that a team playing at home has an advantage. Similarly, in the case of chess, playing White results in a slight advantage. Table 6.5 displays the predictive performance achieved by our model when the score of the home team (respectively, that of the opponent playing White) is modeled by a linear combination of two features: the identity of the team or player and an *advantage* feature. Including this additional feature improves performance significantly, and we conclude that representing opponents in terms of match-dependent combinations of features can be very useful in practice.

Capturing Intransitivity

Score-based models such as ours are sometimes believed to be unable to capture meaningful intransitivities, such as those that arise in the "rock-paper-scissors" game [29]. This is incorrect: if an opponent's score can be modeled by using match-dependent features, we can simply add an *interaction* feature for every pair of opponents. In the next experiment, we model the score difference between two opponents i, j as $d \coloneqq s_i - s_j + s_{ij}$. Informally, the model learns to explain the transitive effects through the usual player scores s_i and s_j and the remaining intransitive effects are captured by the interaction score s_{ij}. We compare this model to the Blade-Chest model of Chen and Joachims [29] on the two StarCraft datasets, known to contain strong intransitivities. The Blade-Chest model is specifically designed to handle intransitivities in comparison data. We also include two baselines, a simple Bradley–Terry model without the interaction features (*logit*) and a non-parametric estimator (*naive*) that estimates probabilities based on match outcomes between each pair—without attempting to capture transitive effects. As shown in Figure 6.5, our model outperforms all other approaches, including the Blade-Chest model.

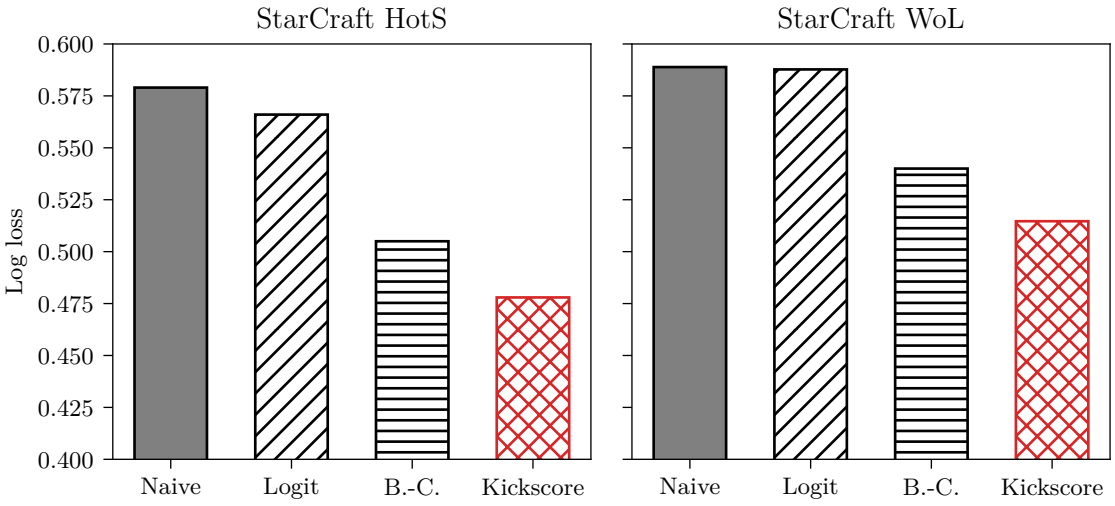


Figure 6.5 – Average log loss of four models (Bradley–Terry, naive, blade-chest, and Kickscore) on the StarCraft datasets.

6.4.3 Inference Algorithm

We turn our attention to the inference algorithm and study the impact of several implementation choices. We start by quantifying the impact of the mean-field assumption (6.4) and of the choice of variational objective on predictive performance. Then, we demonstrate the scalability of the algorithm on the ChessBase dataset and measure the acceleration obtained by parallelizing the algorithm.

Mean-Field Approximation

In order to gain understanding on the impact of the factorization assumption in (6.4), we devise the following experiment. We consider a small subset of the basketball data containing all matches between 2000 and 2005 ($N = 6382$, $M = 32$). We evaluate the predictive performance on each week of the last season by using all the matches prior to the test week as training data. Our model uses a one-to-one mapping between teams and features, a constant + Matérn 1/2 covariance function, and a Gaussian likelihood on the points difference.

We compare the predictive performance resulting from two inference variants, (*a*) mean–field approximate inference, *i.e.*, Algorithm 6.1, and (*b*) *exact* posterior inference[9]. Both approaches lead to an average log loss of 0.634 and an average accuracy of 0.664. Strikingly, both values are equal up to four decimal places, suggesting that the mean-field assumption is benign in practice [13].

[9]This is possible for this particular choice of likelihood thanks to the self-conjugacy of the Gaussian distribution, but at a computational cost $O(N^3)$.

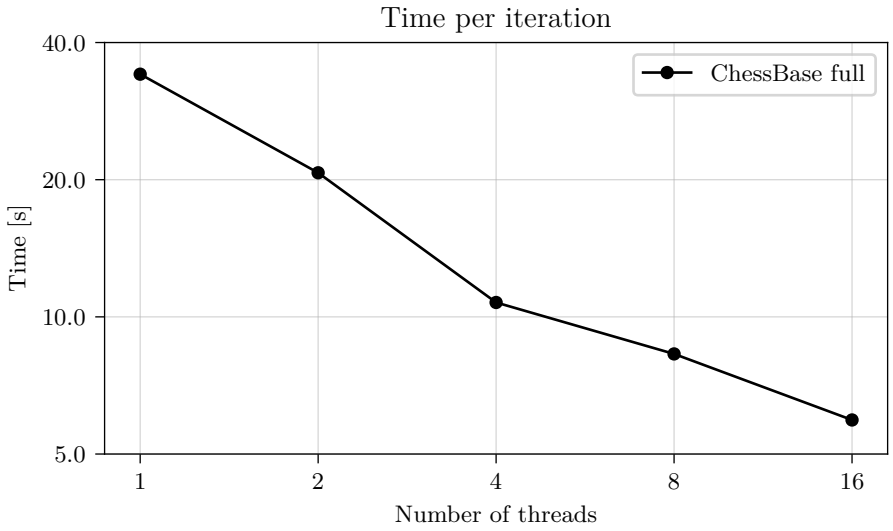


Figure 6.6 – Running time per iteration of a multithreaded implementation of Algorithm 6.1 on the ChessBase full dataset, containing over 7 million observations.

Variational Objective

Next, we study the influence of the variational method. We re-run the experiments of Section 6.4.1, this time by using the reverse-KL objective instead of EP. The predictive performance in terms of average log loss and average accuracy is equal to the EP case (Table 6.3, last three columns) up to three decimal places, for all four datasets. Hence, the variational objective seems to have little practical impact on predictive performance. As such, we recommend using the reverse-KL objective for likelihoods whose log-partition function (6.6) cannot be computed in closed form, as the numerical integration of the expected log-likelihood (6.7) is generally more stable.

Scalability

Finally, we demonstrate the scalability of our inference algorithm by training a model on the full ChessBase dataset, containing over 7 million observations. We implement a multithreaded version of Algorithm 6.1 in the Go programming language[10] and run the inference computation on a machine containing two 12-core Intel Xeon E5-2680 v3 (Haswell generation) processors clocked at 2.5 GHz. Figure 6.6 displays the running time per iteration as function of the number of worker threads. By using 16 threads, we need only slightly over 5 seconds per iteration.

[10]The code is available at `https://github.com/lucasmaystre/gokick`.

6.5 Related Work

As described in Section 1.2, probabilistic models for pairwise comparisons have been studied for almost a century. Thurstone [179] proposed his seminal *law of comparative judgment* in the context of psychology. Almost concurrently, Zermelo [206] developed a method to rank chess players from match outcomes. Both rely on the same idea: objects are characterized by a latent score (*e.g.*, the intrinsic quality of a perceptual variable, or a chess player's skill) and the outcomes of comparisons between objects depend on the difference between the corresponding latent scores. Zermelo's model was later rediscovered by Bradley and Terry [19] and is currently usually referred to as the Bradley–Terry model. Stern [170] provides a unifying framework and shows that, in practice, Thurstone's and Zermelo's models result in similar fits to the data. In the context of sports, some authors suggest going beyond ordinal outcomes and investigate pairwise-comparison models with Gaussian [72], Poisson [117, 72], or Skellam [96] likelihoods.

In many applications of practical interest, comparison outcomes tend to vary over time. In chess, for example, this is due to the skill of players changing over time. The World Chess Federation, which uses a variant of the Bradley–Terry model to rank players, updates player scores after each match by using a stochastic gradient update:

$$s_i \leftarrow s_i + \lambda \frac{\partial}{\partial s_i} \log p(y \mid s_i - s_j),$$

where $\lambda \in \mathbf{R}$ is a learning rate. It is interesting that this simple online update scheme (known as the Elo rating system [53]) enables a basic form of "tracking": the sequence of scores gives an indication of a player's evolution over time. Whereas, in this case, score dynamics occur as a by-product of the learning rule, several attempts have been made to model time dynamics explicitly. Usually, these models assume a variant of Brownian motion:

$$s(t_{n+1}) = s(t_n) + \varepsilon_n, \qquad \varepsilon_n \sim \mathcal{N}(0, \sigma^2 |t_{n+1} - t_n|). \tag{6.9}$$

Glickman [65] and Fahrmeir and Tutz [56] are, to the best of our knowledge, the first to consider such a model. Glickman [66] derives a computationally-efficient Bayesian inference method by using closed-form approximations of intractable integrals. Herbrich et al. [82] and Dangauthier et al. [42] propose a similar method based on Gaussian filtering and expectation propagation, respectively. Coulom [41] proposes a method based on the Laplace approximation. Our model strictly subsumes these approaches; Brownian motion is simply a special case of our model obtained by using the Wiener kernel. One of the key contributions of our work is to show that it is not necessary to restrict the dynamics to Brownian motion in order to get linear-time inference.

Finally, we briefly review literature on the link between Gaussian processes (GPs) with scalar inputs and state-space models (SSMs), as this forms a crucial component of

our fast inference procedure. Excellent introductions to this link can be found in the theses of Saatçi [160] and Solin [168]. The connection is known since the seminal paper of O'Hagan [142], which introduced Gaussian processes as a method to tackle general regression problems. It was recently revisited by Hartikainen and Särkkä [76], who provide formulae for going back-and-forth between GP covariance and state-space forms. Extensions of this link to non-Gaussian likelihood models are discussed in Saatçi [160] and Nickisch et al. [138]. To the best of our knowledge, we are the first to describe how the link between GPs and SSMs can be used in the context of observation models that combine *multiple* processes, by using a mean-field variational approximation.

6.6 Summary

We have presented, Kickscore, a probabilistic model of pairwise comparison outcomes that can capture a wide range of temporal dynamics. This model reaches state-of-the-art predictive performance on several sports datasets, and it enables generating visualizations that help in understanding comparison time-series. To fit our model, we have derived a computationally efficient approximate Bayesian inference algorithm. To the best of our knowledge, our algorithm is the first linear-time Bayesian inference algorithm for dynamic pairwise comparison models that minimizes the reverse-KL divergence.

Perspective One of the strengths of our approach is that it enables to discover the structure of the time dynamics by comparing the log-marginal likelihood of the data under various choices of covariance functions. In the future, we would like to fully automatize this discovery process, in the spirit of the *automatic statistician* [50]. Ideally, given only the comparison data, we should be able to systematically discover the time dynamics that best explain the data, and generate an interpretable description of the corresponding covariance functions.

7 Conclusion

In this thesis, we have answered specific questions about social processes from a machine-learning and data-mining viewpoint. In order to propose predictive models that enable interpretation through the learned parameters, we have built upon the literature on discrete-choice models, which we have combined with latent-factor models, Bayesian statistics, and generalized linear models. We have demonstrated that discrete-choice analysis offers a principled and powerful approach to modeling social processes, as making choices is inherent in human behaviour.

In Chapter 2, we have studied the social dynamics behind group collaborations for the collective creation of content, such as in Wikipedia and the Linux kernel. We have proposed a new discrete-choice model inspired from the Bradley-Terry and Rasch models, which incorporates ideas from collaborative filtering. The model enables us to identify controversial Wikipedia articles and core Linux components that are crucial to the system. We have improved the predictive performance by including latent factors, which in turn have helped us understand how users edit some Wikipedia articles: They are either "experts" in popular culture or in high culture, but not both.

In Chapter 3, we have studied, through the lens of peer-production systems, the law-making process in the European Union. To capture the conflictive structure inherent in this process, we have designed a model inspired from the multinomial logit and Rasch models, which we have enhanced with natural language processing techniques. We have quantified the controversy of laws and proposed intuitive visualizations by representing each law as the conflict graph of its edits. We have also identified features of the edits that correlate with a higher probability of acceptance: For example, inserting short edits, providing a justification for the change, deleting "human rights", and having the parliamentarian in charge of the law sponsor the edit are all factors that contribute to acceptance.

In Chapter 4, we have developed an algorithm that combines matrix factorization and generalized linear models for predicting the popular vote of elections and referenda from

partial, regional results. This algorithm learns representations of votes and regions to capture ideological and cultural voting patterns (e.g., rural/urban, liberal/conservative, etc.). Its predictions are also accurate: In Switzerland, for example, it is able to predict referendum votes with an accuracy of 99% and a mean absolute error of less than 1% using only 5% of the results observed in municipalities. We have deployed our algorithm on a Web platform to make real-time predictions for referenda in Switzerland.

In Chapter 5, we have studied how people perceive the carbon footprint of their day-to-day actions. We have cast this problem as a comparison problem between pairs of actions (*e.g.*, between intercontinental flights and using household appliances) and developed a statistical model of relative comparisons reminiscent of the Thurstone model in psychometrics. The model learns users' perception as the parameters of a Bayesian linear regression, which enables us to derive an active-learning algorithm to select the optimal pairs of actions to probe. Because no suitable data existed for answering these questions, we built a Web interface to collect comparison data from students on our university campus. Our results show that users tend to overestimate actions with low carbon footprint and underestimate actions with high carbon footprint.

Finally, in Chapter 6, we have developed a dynamic choice-model that enables the parameters to vary over time. We achieve this by replacing the static parameters by continuous-time Gaussian processes. We have also developed an efficient inference algorithm that computes an approximate Bayesian posterior distribution in a few linear-time iterations over the data. We have shown experimentally on several datasets of (e-)sports that this model outperforms competing approaches in terms of predictive performance, scales to millions of observations, and generates compelling visualizations of the parameters' dynamics. We have deployed our approach as a real-world application on the Web for predicting football matches in European leagues and international competitions.

Ethical Considerations Studying human behavior and addressing social problems from a computational viewpoint induces a risk of abuse. This is especially true for political processes, as popularized by the infamous scandal of Cambridge Analytica in 2016. After submitting our paper [105] forming the basis of Chapter 3 to the Web Conference 2021, one anonymous reviewer expressed concerns regarding the use of machine learning for making decisions in law making, and whether our findings in Section 3.6 could help adversarial attacks. We answer to such concerns by precising that we do not propose to rely on our models for making decisions, such as whether an edit should be accepted or not. Our goal is to understand the factors correlated with the acceptance of edits, and thereby gain insights into the law-making processes. These correlations do not imply a causal relationship that would benefit potential adversarial attackers. Nevertheless, even if such a relationship were to exist, we prefer that these findings are published in an open research community, where possible countermeasures to such attacks could be thought of

for the public good, rather than them being discovered by a company or an influence group that might use them in an opaque manner to push private interests.

Our vote prediction algorithm of Chapter 4 also triggered private concerns regarding its potential effect on the final outcome. If voters see early predictions of the outcome of an election or referendum, will they decide not to vote at all (if the prediction is in their favor) or, on the contrary, will they encourage more people to do so in order to swing the result (if the prediction goes against their preference)? In the political science literature on election forecasting, this is referred to as the *bandwagon* and the *underdog* effects, respectively,and it is unclear which one prevails. However, a recent paper that studies the effect of probabilistic forecasting in the 2016 U.S. election concludes that "forecasting can fundamentally alter the information environment available to potential voters, with the potential to change the outcome of elections" [194]. For that reason, some countries, such as France, Spain, Italy, and Canada, enforce *election silence*: Polling and political campaigning (including predictions) are forbidden some time prior to the voting day to prevent influencing voting behavior and election outcome. In Switzerland, where no such policy is implemented, voting offices close at 12:00pm on the Sunday of the vote. Our predictions are made during ballot counting, which takes a few hours and starts only when voting has closed everywhere. Hence, our predictions cannot influence the voting behavior, because it is impossible to vote when our predictions are made public.

Perspective on Interpretability and Causality In the most machine-learning fashion, the results of this thesis rely on correlations between features and predicted outputs. For example, in Chapter 3, we uncover features of law edits that correlate well with higher probability of acceptance. While these findings enable us to shed light on the problems we address, they could be made stronger by taking a causal inference perspective. In particular, a rigorous evaluation of potential confounding factors would reinforce our statistical models and conclusions. For example, in Chapter 3, controversy might be a confounding factor: Many edits might be rejected because the laws are controversial, but controversy is also modeled as a parameter of the law proposals. Causal inference and reasoning is rapidly expanding in the machine-learning community. Discovering causal relationships in the problems addressed in this thesis could be used to derive new insights and to develop our methods further.

Perspective on Methodological Uncertainties By definition, computational models are approximate representations of (complex) realities. Human behaviour, in particular, is uncertain by essence, and models of social processes are only as good as the datasets they rely on. In the presence of noisy data, it becomes crucial to quantify uncertainty and propagate it through parameter estimation. This enables a model to provide predictive distributions rather than point-wise predictions. Bayesian inference offers a principled approach to achieve uncertainty quantification and propagation. In this work, we have

relied on such methods only in Chapter 6 and, to some extent, in Chapter 5. We believe that the other chapters, and the SubSVD-GLM algorithm of Chapter 4 in particular, would benefit from Bayesian inference to provide more informative and robust predictions.

The discrete-choice models on which the present work is based, such as the Bradley-Terry, the Thurstone, and the multinomial logit models, are subject to structural uncertainty because they assume that the alternatives in the choice set are independent. Clearly, this strong assumption might limit the depth of some analyses. Resorting to models that encode dependencies between alternatives offers a natural and promising direction to further exploit the structure of the problems at hand. For example, the mixed logit, nested logit, and multinomial probit models all enable this by encoding correlations through the joint distribution of the noise model (see Section 1.2). To the best of our knowledge, combining matrix factorization techniques with these models has not yet been explored and opens up fascinating research directions that could lead to new methodologies in discrete-choice analysis and preference learning.

Perspective on Socio-Environmental Processes This thesis is written at a time when the global political spectrum is more polarized than ever and in a society that faces the grand environmental challenges of climate change and biodiversity loss. Although currently being mostly part of the problem, computer science and machine-learning algorithms can become part of the solution. Impactful policies require ambitious target setting and effective implementation. They require combining (1) top-down processes, *i.e.*, how policy-makers shape laws, and (2) bottom-up processes, *i.e.*, how individuals make choices in their daily life. For (1), machine-learning methods can help to discover influence networks and lobbying activities in political processes. For (2), understanding how people make choices and change their opinions over time gives a starting point to bridge the gap between policy and implementation. Processing large-scale datasets of legal texts, parliamentary speeches, and social media activities with recent methods in language modeling, latent-factor models, and network science offers a promising direction to study the hidden influence processes and to understand people's behaviour in law-making and political participation. Monitoring societal currents and making the results available to the general public can increase transparency into political processes and help shape fair, ethical, and effective policies.

A Call for Interdisciplinary Research Tackling social-science problems from a machine-learning perspective gave us the advantage of agnostic analyses, but sometimes at the expense of some limitation in the analysis. Indeed, collaborating with political scientists, economists, psychologists, sociologists, climate scientists, and ecologists would provide expert knowledge to strengthen our findings. As computer science increasingly percolates through other scientific domains, computational methods are often used only as a means to an end, rather than as a source of innovative approaches and fresh viewpoints.

Interdisciplinary research can act as an effective catalyst for scientific progress, as major breakthroughs often take place by crossing ideas from different fields. For example, the sequencing of the human genome in the 1990s required the collective efforts of physicists, chemists, biologists, and computer scientists. Today, at a time when conspiracy theories disseminate doubt and threaten the acceptance of facts, interdisciplinary research could restore trust in science by reinforcing the credibility of scientific results and enhancing their scope. Only synergistic collaborations with other fields will enable computer science to unleash its true potential for transformative societal good.

A Appendix

A.1 Predikon

To make the research presented in Chapter 4 available to the general public, we developed Predikon [1], a website predicting Swiss referendum votes in real time. Four times a year, Swiss citizens are called to vote on referenda and popular initiatives. They can send their ballot remotely or come to the ballot office, on the date of the vote, to deposit it. Then, starting at 12pm, officials in Swiss municipalities start counting the ballots and report the results as soon as they finish. We make predictions using the partial national results, *i.e.*, using only the results in the municipalities that are done counting and have reported their results. In Figure A.1, the homepage of Predikon shows the predictions in real time for current votes and those of past votes. We built an interactive tool to visualize the projections of the municipalities in the latent ideological space of Section 4.3.2, with PCA (see Figure A.2) and t-SNE (see Figure A.3).

A.2 Climpact

A.2.1 Web Platform

To collect data about people's perception from real users, we built the Climpact platform and opened it for students on our university campus. Users answers questions in a quiz that asks them to compare pairs of actions. They answer in relative terms, *i.e.*, they indicate the relative order of magnitude between two actions, as shown in Figure A.4. Once the quiz is finished, they have access to their answers that they can compare against the correct values (see Figure A.5. Each action has its own page, and we display the perceived carbon footprint and the true values of several actions on one plot (see Figure A.6.

[1] https://www.predikon.ch

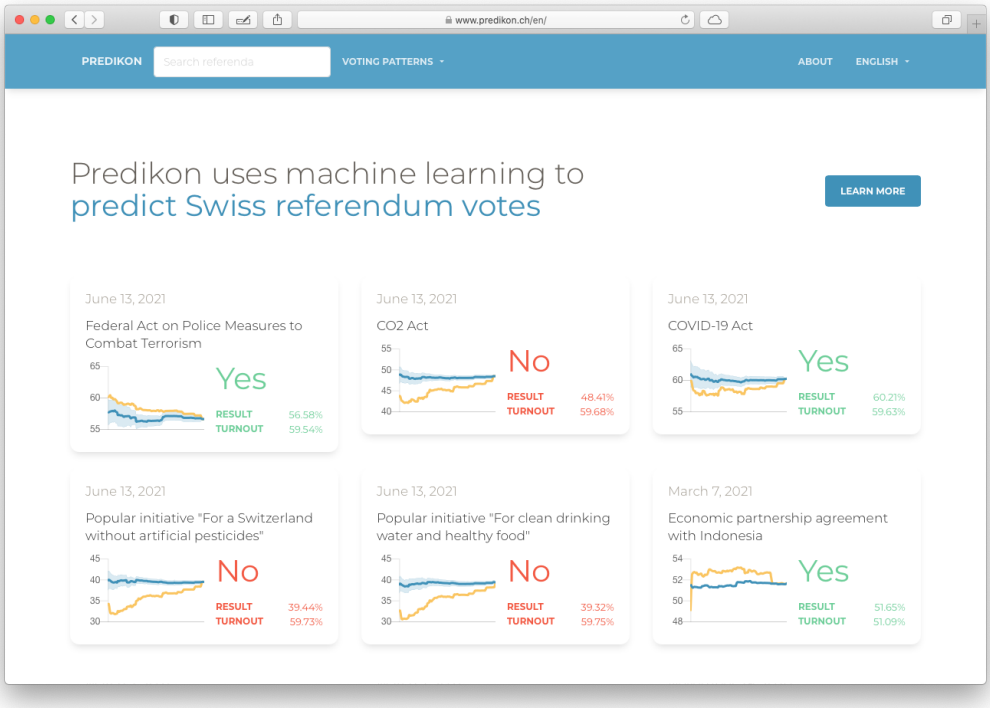


Figure A.1 – Home page of Predikon showing predictions of past votes.

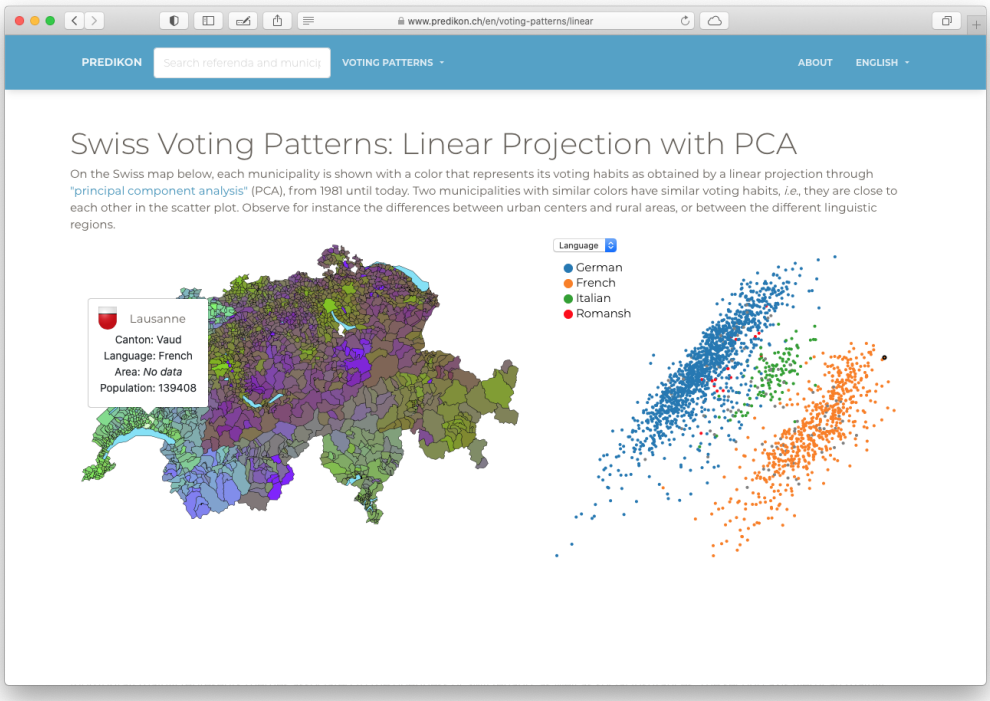


Figure A.2 – Projection of municipalities in ideological space using PCA.

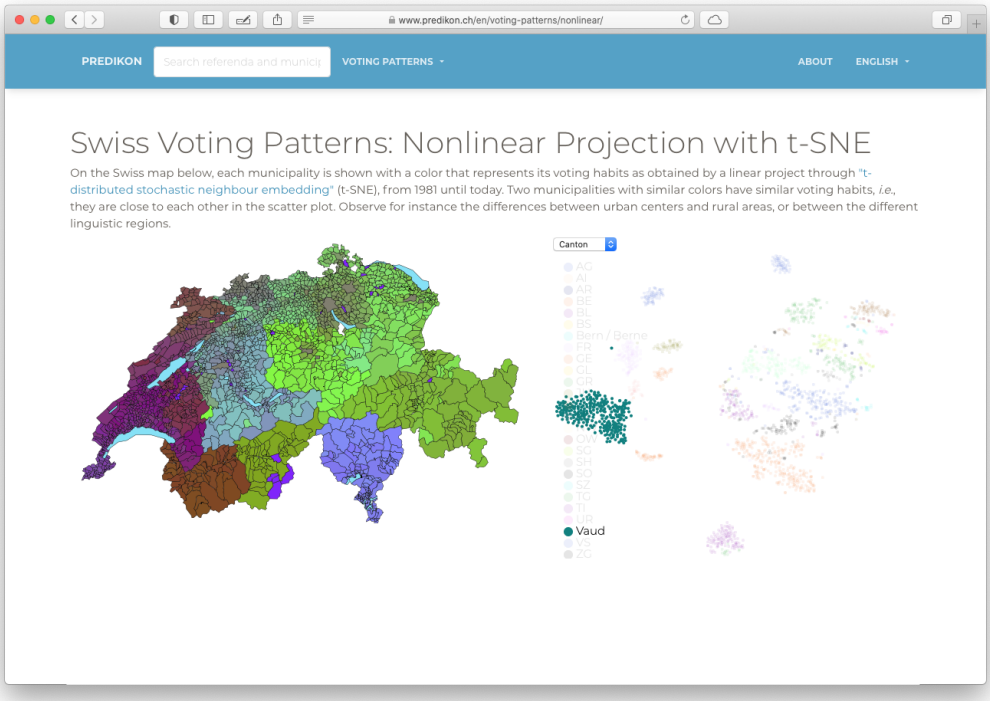


Figure A.3 – Projection of municipalities in ideological space using t-SNE.

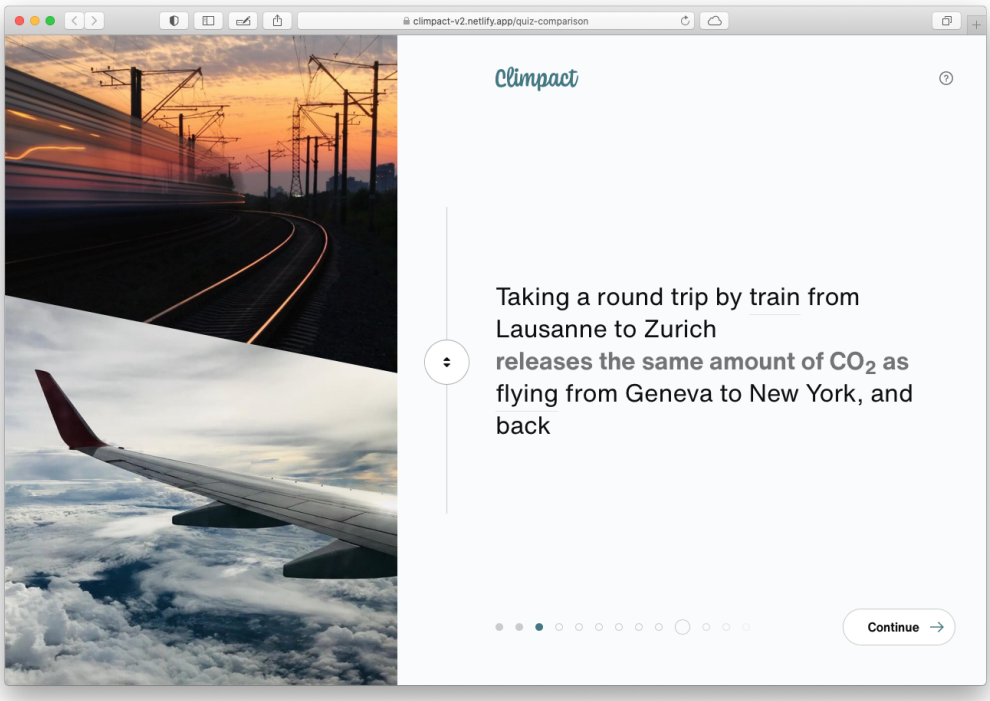


Figure A.4 – Users answer simple questions in the form of pairwise comparisons.

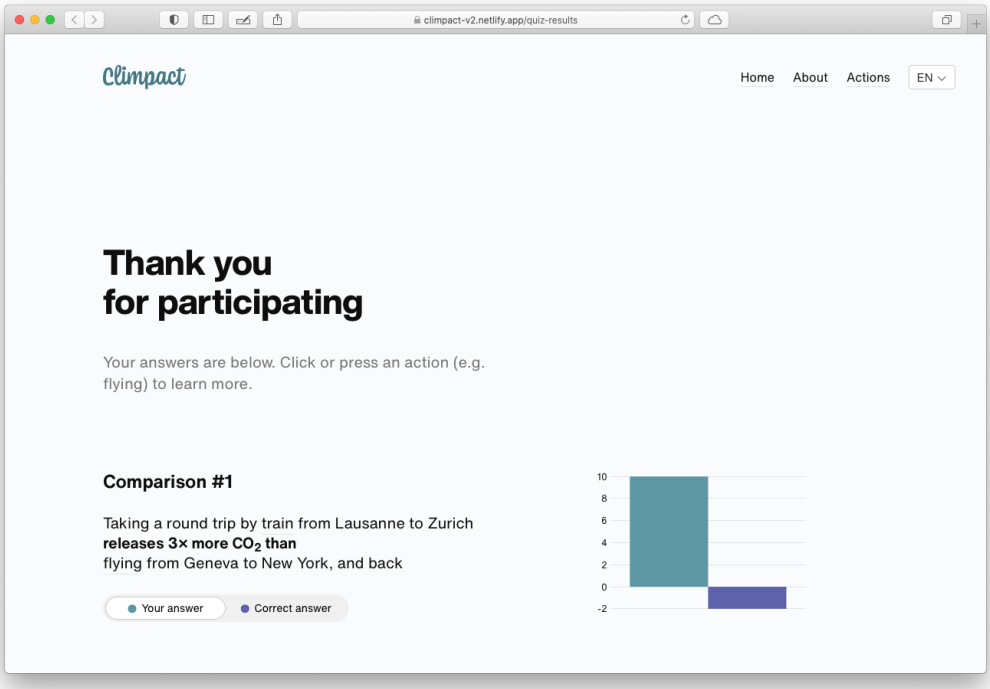


Figure A.5 – After the quiz, users can compare their answers with the correct ones.

A.2.2 List of Actions

We provide here the full list of actions, together with the true carbon footprint associated with each of them. Because different countries use different sources of energy, we calculate the carbon footprint *relative* to the country where our university is located. The actions are ordered according to their true carbon footprint.

1. **Take the train in economy class on a 1000-km round-trip.**
   The train is a high-speed train with 360 seats. The seat-occupancy rate is 55% (average rate for these types of trains). We count the CO_2 emissions per passenger.
   Carbon footprint: 17 kgCO_2-equivalent.

2. **Dry your clothes with a dryer for one year.**
   A dryer emits CO_2 because it consumes electricity. We consider a dryer of average quality. The electricity is consumed from a grid with average CO_2 rate.
   Carbon footprint: 40 kgCO_2-equivalent.

3. **Light your house with LED bulbs.**
   LED bulbs emit CO2 because they consume electricity to generate light. The electricity is consumed from a grid with average CO_2 rate.
   Carbon footprint: 40 kgCO_2-equivalent.

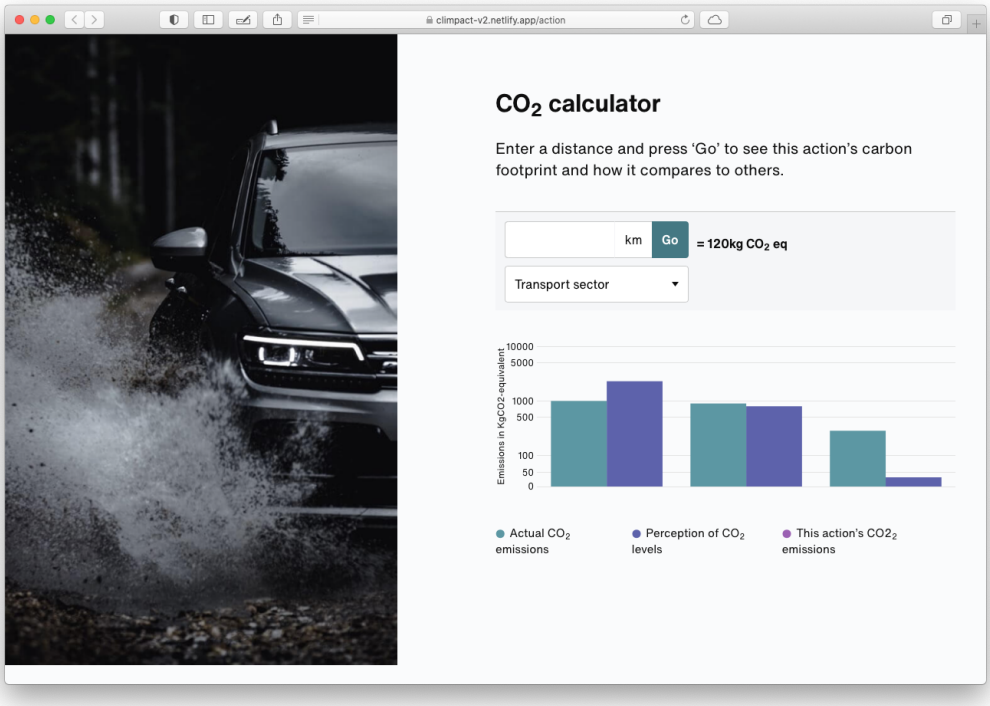


Figure A.6 – Each action has a page, where the perceive value for this action and the true values are displayed on a plot and compared with other actions.

4. **Take the bus on a 1000-km round-trip.**
   The bus is a standard-size bus with 60 seats. The seat-occupancy rate is 50% (average rate for buses). We count the CO_2 emissions per passenger.
   Carbon footprint: 45 kgCO_2-equivalent.

5. **Drive an electric car alone on a 1000-km round-trip.**
   The car is a compact electric car that consumes 15 kWh/100km. The electricity is consumed from a grid with average CO_2 rate. There are no other passengers in the car. We count the CO_2 emissions per passenger.
   Carbon footprint: 45 kgCO_2-equivalent.

6. **Car-share with three other persons on a 1000-km round-trip.**
   The car is a mid-sized gasoline car that consumes 7 l/100km. There are four persons in the car. We count the CO_2 emissions per passenger.
   Carbon footprint: 75 kgCO_2-equivalent.

7. **Eat local and seasonal fruits and vegetables for one year.**
   Growing food emits CO_2 because it requires fertilizing and driving agricultural machines. The goods are then transported to grocery shops and to your home.
   Carbon footprint: 89 kgCO_2-equivalent.

8. **Eat eggs and dairy products for one year.**
   The production of eggs and dairy products (milk, cheese, etc.) emits CO_2 because of water and land consumption, animal methane, and fossil fuel consumption for transportation and heating. We consider an average citizen consuming 50 kg of eggs and dairy products per year.
   Carbon footprint: 100 kgCO_2-equivalent.

9. **Throw all waste in the same trash for one year.**
   Throwing all waste (PET, glass, cardboard, etc.) in the same trash, *i.e.*, without recycling, emits CO_2 because more energy is needed to extract, transport, and process raw materials. Incinerators also burn more waste, and organic waste decomposition generates methane.
   Carbon footprint: 200 kgCO_2-equivalent.

10. **Light your house with incandescent bulbs.**
    Incandescent bulbs emit CO_2 because they consume electricity to generate light. The electricity is consumed from a grid with average CO_2 rate.
    Carbon footprint: 239 kgCO_2-equivalent.

11. **Fly in economy class for a 800-km round-trip.**
    The plane is a standard aircraft for short-distance flights with 180 seats. The seat-occupancy rate is 80%. We count the CO_2 emissions per passenger.
    Carbon footprint: 270 kgCO_2-equivalent.

12. **Drive alone for a 1000-km round-trip.**
    The car is a mid-sized gasoline car that consumes 7 l/100km. There are no other passengers in the car. We count the CO_2 emissions per passenger.
    Carbon footprint: 300 kgCO_2-equivalent.

13. **Heat your house with a heat pump for one year.**
    A heat pump emits CO_2 because it consumes electricity to generate heat. The house is of average size. The electricity is consumed from a grid with average CO_2 rate.
    Carbon footprint: 400 kgCO_2-equivalent.

14. **Eat imported and out-of-season fruits and vegetables for one year.**
    Growing food emits CO_2 because it requires fertilizing and driving agricultural machines. Importing food emits CO_2 because of fossil fuel consumption for transportation. Out-of-season food emits CO_2 because it grows in greenhouse that needs to be heated. The goods are then transported to grocery shops and to your home.
    Carbon footprint: 449 kgCO_2-equivalent.

15. **Eat meat for one year.**
    Meat production emits CO_2 because of water and land consumption, animal methane, and fossil fuel consumption for transportation and heating. We consider an average citizen consuming 50 kg of meat per year.
    Carbon footprint: 800 kgCO_2-equivalent.

16. **Fly in economy class for a 12000-km round-trip.**
    The plane is a standard aircraft for long-distance flights with 390 seats. The seat-occupancy rate is close to 100%. We count the CO_2 emissions per passenger.
    Carbon footprint: 2300 kgCO_2-equivalent.

17. **Heat your house with an oil furnace for one year.**
    An oil furnace emits CO_2 because it burns fuel to generate heat. The house is of average size.
    Carbon footprint: 3300 kgCO_2-equivalent.

18. **Fly in first class for a 12000-km round-trip.**
    The plane is a standard aircraft for long-distance flights with 390 seats. The seat-occupancy rate is close to 100%. We count the CO_2 emissions per passenger. Passengers flying in first class use more space than passengers in economy.
    Carbon footprint: 9000 kgCO_2-equivalent.

A.3 Kickoff.ai

We built Kickoff.ai to predict football matches from the top-5 European leagues (France, Spain, Italy, England, and Germany) and two international competitions (European

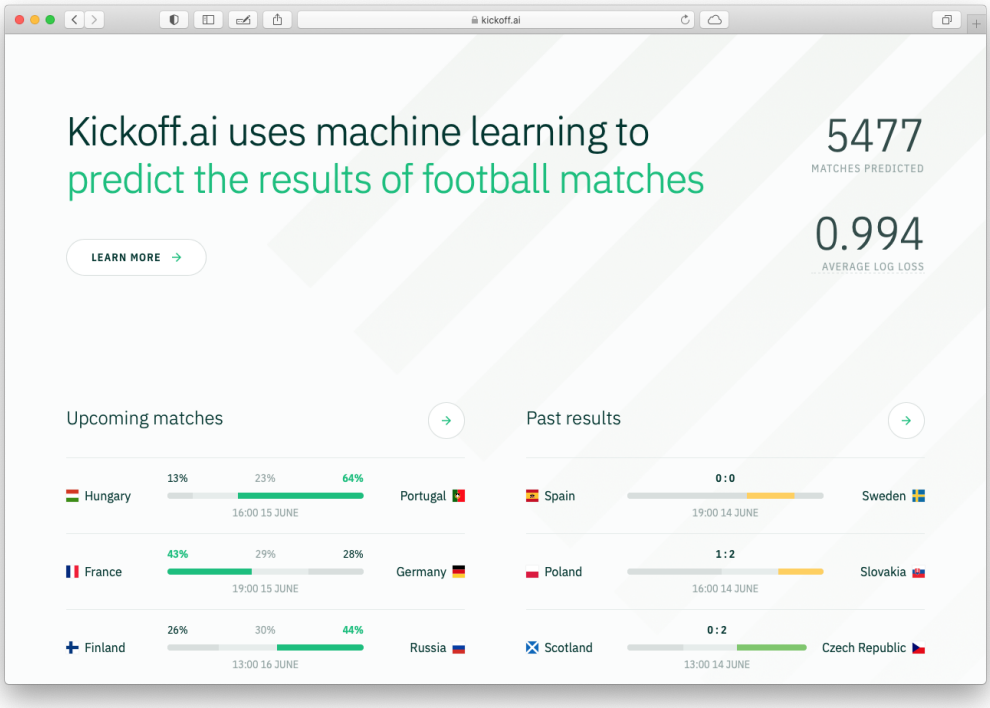


Figure A.7 – Home page of Kickoff.ai that shows the predictions for upcoming matches.

Championship and World Cup). Our predictions use the Kickscore model of Chapter 6. We provide a predicted probability for the victory of Team A, of Team B, or a draw (see Figure A.7). Each match has its own page, own which we display the learned latent skills of the two teams (see Figure A.8.

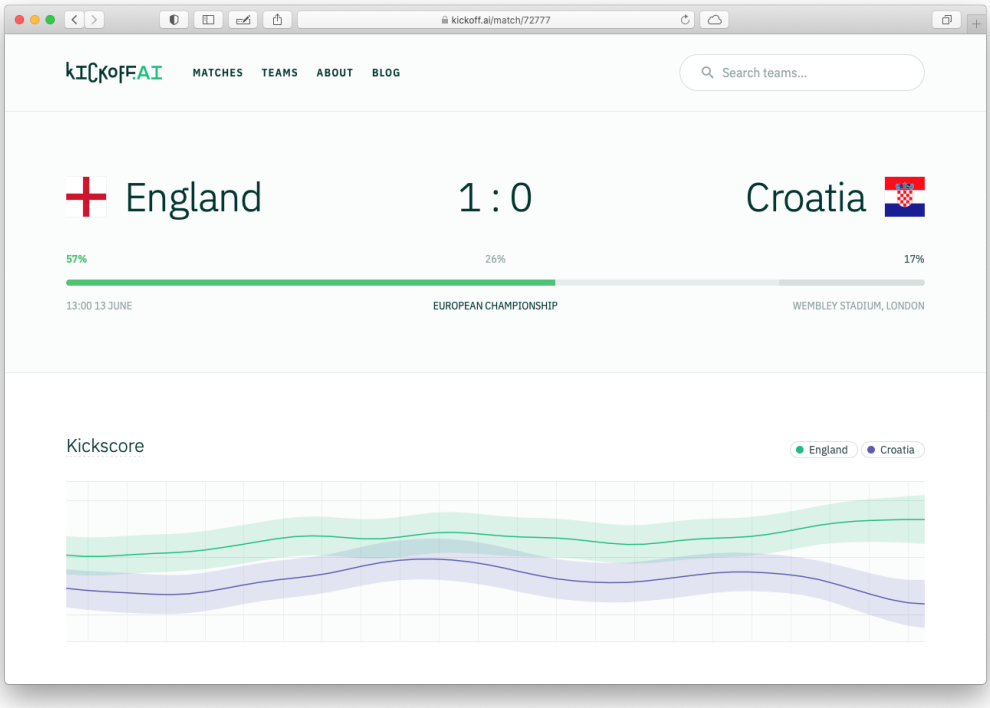


Figure A.8 – Each match has a page, where we show a visualization of the latent skill of teams.

Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: A system for large-scale machine learning. In *Proceedings of OSDI'16*, Savannah, GA, USA, Nov. 2016. [Cited on page 19]

[2] B. T. Adler and L. de Alfaro. A content-driven reputation system for the Wikipedia. In *Proceedings of WWW'07*, Banff, AB, Canada, May 2007. [Cited on pages 14, 16, 18, 21, 23, and 61]

[3] B. T. Adler, L. de Alfaro, I. Pye, and V. Raman. Measuring author contributions to the Wikipedia. In *Proceedings of WikiSym'08*, Porto, Portugal, Sept. 2008. [Cited on page 16]

[4] A. Agresti. *Categorical Data Analysis*. Wiley, third edition, 2012. [Cited on page 96]

[5] A. Ammar. *Ranked Personalized Recommendations Using Discrete Choice Models*. PhD thesis, Massachusetts Institute of Technology, 2015. [Cited on page 3]

[6] I. Baller. Specialists, party members, or national representatives: Patterns in co-sponsorship of amendments in the european parliament. *European Union Politics*, 18(3):469–490, 2017. [Cited on page 61]

[7] R. Bamler and S. Mandt. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 380–389, 2017. [Cited on page 72]

[8] L. H. Bean. How to predict elections. 1948. [Cited on page 84]

[9] E. Belanger. Finding and using empirical data for vote and popularity functions in France. *French Politics*, 2(2):235–244, 2004. [Cited on page 84]

[10] 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. [Cited on page 69]

[11] M. E. Ben-Akiva. *Structure of passenger travel demand models.* PhD thesis, Massachusetts Institute of Technology, 1973. [Cited on pages 2 and 3]

[12] 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. [Cited on page 84]

[13] A. Birlutiu and T. Heskes. Expectation propagation for rating players in sports competitions. In *Proceedings of PKDD 2007*, Warsaw, Poland, Sept. 2007. [Cited on page 110]

[14] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006. [Cited on pages 19 and 27]

[15] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. [Cited on page 101]

[16] M. Blumenthal. The poblano model, 2008. URL `https://web.archive.org/web/20090414152429/http://www.nationaljournal.com/njonline/mp_20080507_8254.php`. Accessed: 2021-06-15. [Cited on page 84]

[17] J. H. Boyd and R. E. Mellman. The effect of fuel economy standards on the US automotive market: an hedonic demand analysis. *Transportation Research Part A: General*, 14(5-6):367–378, 1980. [Cited on page 3]

[18] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004. [Cited on page 67]

[19] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. [Cited on pages 3, 7, 48, 97, and 112]

[20] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, pages 707–720. Springer, 2002. [Cited on page 84]

[21] 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. [Cited on page 84]

[22] A. Bronner and C. Monz. User edits classification using document revision histories. In *Proceedings of EACL 2012*, Avignon, France, Apr. 2012. [Cited on page 16]

[23] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. [Cited on page 52]

[24] H. Bühlmann and P. J. Huber. Pairwise comparison and ranking in tournaments. *The Annals of Mathematical Statistics*, 34(2):501–510, 1963. [Cited on page 3]

[25] N. S. Cardell and F. C. Dunbar. Measuring the societal impacts of automobile downsizing. *Transportation Research Part A: General*, 14(5-6):423–434, 1980. [Cited on page 3]

[26] F. Caron and A. Doucet. Efficient Bayesian inference for generalized Bradley–Terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012. [Cited on page 3]

[27] E. Cencig and L. Sabani. Voting behaviour in the European Parliament and economic governance reform: does nationality matter? *Open Economies Review*, 28 (5):967–987, 2017. [Cited on page 47]

[28] I. Chalkidis and D. Kampas. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, 27(2): 171–198, 2019. [Cited on page 50]

[29] S. Chen and T. Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of WSDM'16*, San Francisco, CA, USA, Feb. 2016. [Cited on page 109]

[30] 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. [Cited on page 84]

[31] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2017. [Cited on page 3]

[32] W. Chu and Z. Ghahramani. Extensions of Gaussian processes for ranking: Semi-supervised and active learning. In *Proceedings of the NIPS 2005 Workshop on Learning to Rank*, Whistler, BC, Canada, Dec. 2005. [Cited on page 89]

[33] D. Chumbalov, L. Maystre, and M. Grossglauser. Scalable and efficient comparison-based search without features. In *International Conference on Machine Learning*, pages 1995–2005. PMLR, 2020. [Cited on page 4]

[34] L. Cicchi. The logic of voting behaviour in the European Parliament: new insights on party group membership and national affiliation as determinants of vote. 2013. [Cited on page 47]

[35] M. Cinelli, G. De Francisci Morales, A. Galeazzi, W. Quattrociocchi, and M. Starnini. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9), 2021. [Cited on page 1]

[36] E. E. Coman. Reassessing the influence of party groups on individual members of the European Parliament. *West European Politics*, 32(6):1099–1117, 2009. [Cited on page 55]

[37] J. Corbet and G. Kroah-Hartman. 2017 Linux kernel development report. Technical report, The Linux Foundation, 2017. [Cited on pages 14 and 28]

[38] D. Cosley, D. Frankowski, L. Terveen, and J. Riedl. SuggestBot: Using intelligent task routing to help people find work in Wikipedia. In *Proceedings of IUI'07*, Honolulu, HI, USA, Jan. 2007. [Cited on page 33]

[39] S. R. Cosslett. Efficient estimation of discrete-choice models. *Structural analysis of discrete data with econometric applications*, 3:51–111, 1981. [Cited on page 3]

[40] R. Costello and R. Thomson. The policy impact of leadership in committees: Rapporteurs' influence on the European Parliament's opinions. *European Union Politics*, 11(2):219–240, 2010. [Cited on page 56]

[41] R. Coulom. Whole-history rating: A Bayesian rating system for players of time-varying strength. In *Proceedings of CG 2008*, Beijing, China, Sept. 2008. [Cited on pages 94 and 112]

[42] P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel. TrueSkill through time: Revisiting the history of chess. In *Advances in Neural Information Processing Systems 20*, Vancouver, BC, Canada, Dec. 2007. [Cited on pages 94 and 112]

[43] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979. [Cited on page 16]

[44] L. de Alfaro and B. T. Adler. Content-driven reputation for collaborative systems. In *Proceedings of TGC 2013*, Buenos Aires, Argentina, Aug. 2013. [Cited on pages 14, 16, and 21]

[45] L. de Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. Reputation systems for open collaboration. *Communications of the ACM*, 54(8):81–87, 2011. [Cited on page 16]

[46] 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. [Cited on page 73]

[47] G. F. do Brasil. Dados abertos, 2021. URL `https://www.gov.br/cgu/pt-br/acesso-a-informacao/dados-abertos`. Accessed: 2021-06-15. [Cited on page 36]

[48] G. Druck, G. Miklau, and A. McCallum. Learning to predict the quality of contributions to Wikipedia. In *Proceedings of WikiAI 2008*, Chicago, IL, USA, July 2008. [Cited on pages 14, 16, 21, and 61]

[49] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul):2121–2159, 2011. [Cited on page 52]

[50] D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, University of Cambridge, 2014. [Cited on pages 99 and 113]

[51] 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. [Cited on page 84]

[52] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 1936. [Cited on page 69]

[53] A. Elo. *The Rating Of Chess Players, Past & Present*. Arco Publishing, 1978. [Cited on pages 3, 94, 106, and 112]

[54] V. Etter, J. Herzen, M. Grossglauser, and P. Thiran. Mining democracy. In *Proceedings of the second ACM Conference on Online Social Networks*, 2014. [Cited on pages 75 and 82]

[55] 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. [Cited on pages 68, 69, 72, 74, 82, and 85]

[56] L. Fahrmeir and G. Tutz. Dynamic stochastic models for time-dependent ordered paired comparison systems. *Journal of the American Statistical Association*, 89 (428):1438–1449, 1994. [Cited on pages 94 and 112]

[57] D. Finke. Proposal stage coalition-building in the European Parliament. *European Union Politics*, 13(4):487–512, 2012. [Cited on page 61]

[58] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017. [Cited on page 2]

[59] L. R. Ford, Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957. [Cited on page 3]

[60] F. Franch. (wisdom of the crowds) 2: 2010 uk election prediction with social media. *Journal of Information Technology & Politics*, 10(1):57–71, 2013. [Cited on page 84]

[61] W. C. Fuller, C. F. Manski, and D. A. Wise. New evidence on the economic determinants of postsecondary schooling choices. *Journal of Human Resources*, pages 477–498, 1982. [Cited on page 2]

[62] K. Garimella, G. De Francisci Morales, A. Gionis, and M. Mathioudakis. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, pages 913–922, 2018. [Cited on page 1]

[63] GitHub. The state of the Octoverse 2017, 2017. URL `https://octoverse.github.com/`. Accessed: 2021-06-15. [Cited on page 14]

[64] W. A. Glenn and H. A. David. Ties in paired-comparison experiments using a modified Thurstone–Mosteller model. *Biometrics*, 16(1):86–109, 1960. [Cited on page 97]

[65] M. E. Glickman. *Paired Comparison Models with Time-Varying Parameters*. PhD thesis, Harvard University, 1993. [Cited on page 112]

[66] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 48(3): 377–394, 1999. [Cited on pages 94 and 112]

[67] A. A. Goett, K. Hudson, and K. E. Train. Customers' choice among retail energy suppliers: The willingness-to-pay for service attributes. *The Energy Journal*, 21(4), 2000. [Cited on page 2]

[68] Google. word2vec, 2013. URL `https://code.google.com/archive/p/word2vec/`. Accessed: 2021-06-15. [Cited on page 50]

[69] S. Government. Swiss open government data, 2021. URL `https://opendata.swiss/en/`. Accessed: 2021-06-15. [Cited on page 36]

[70] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018. [Cited on page 2]

[71] J. Guiver and E. Snelson. Bayesian inference for Plackett–Luce ranking models. In *Proceedings of ICML 2009*, Montréal, QC, Canada, June 2009. [Cited on page 3]

[72] S. Guo, S. Sanner, T. Graepel, and W. Buntine. Score-based Bayesian skill learning. In *Proceedings of ECML PKDD 2012*, Bristol, United Kingdom, Sept. 2012. [Cited on pages 97 and 112]

[73] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6: 437–450, 2018. [Cited on page 62]

[74] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2125–2126, 2016. [Cited on page 1]

[75] A. Halfaker and D. Taraborelli. Artificial intelligence service "ORES" gives Wikipedians X-ray specs to see through bad edits, Nov. 2015. URL `https://blog.wikimedia.org/2015/11/30/artificial-intelligence-x-ray-specs/`. Accessed: 2021-06-15. [Cited on pages 14 and 23]

[76] J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *Proceedings of MLSP 2010*, Kittilä, Finland, Aug. 2010. [Cited on pages 103, 104, and 113]

[77] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998. [Cited on page 3]

[78] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein. Imputing missing data for gene expression arrays. 1999. [Cited on page 84]

[79] S. Heindorf, M. Potthast, B. Stein, and G. Engels. Vandalism detection in Wikidata. In *Proceedings of CIKM'16*, Indianapolis, IN, USA, Oct. 2016. [Cited on page 14]

[80] D. A. Hensher and W. H. Greene. The mixed logit model: the state of practice. *Transportation*, 30(2):133–176, 2003. [Cited on pages 3 and 63]

[81] D. A. Hensher, J. M. Rose, J. M. Rose, and W. H. Greene. *Applied choice analysis: a primer*. Cambridge university press, 2005. [Cited on page 4]

[82] R. Herbrich, T. Minka, and T. Graepel. TrueSkill™: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19*, Vancouver, BC, Canada, Dec. 2006. [Cited on pages 95, 97, 106, and 112]

[83] J. Hilton, N. Cammarata, S. Carter, G. Goh, and C. Olah. Understanding rl vision. *Distill*, 2020. doi: 10.23915/distill.00029. https://distill.pub/2020/understanding-rl-vision. [Cited on page 2]

[84] S. Hix. Parliamentary behavior with two principals: preferences, parties, and voting in the European Parliament. *American Journal of Political Science*, pages 688–698, 2002. [Cited on pages 47 and 55]

[85] S. Hix and B. Høyland. Empowerment of the European parliament. *Annual Review of Political Science*, 16:171–189, 2013. [Cited on page 47]

[86] S. Hix, A. Noury, and G. Roland. Voting patterns and alliance formation in the European Parliament. *Philosophical transactions of the royal society b: biological sciences*, 364(1518):821–831, 2008. [Cited on page 47]

[87] N. Houlsby, J. M. Hernández-Lobato, F. Huszár, and Z. Ghahramani. Collaborative Gaussian processes for preference learning. In *Advances in Neural Information Processing Systems 25*, Lake Tahoe, CA, Dec. 2012. [Cited on pages 3 and 89]

[88] P. B. O. W. House. Open government initiative, 2018. URL `https://obamawhitehouse.archives.gov/open`. Accessed: 2021-06-15. [Cited on page 36]

[89] D. R. Hunter. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32(1):384–406, 2004. [Cited on page 3]

[90] S. Hurka. Changing the output: The logic of amendment success in the European Parliament's ENVI committee. *European Union Politics*, 14(2):273–296, 2013. [Cited on page 61]

[91] A. Immer, V. Kristof, M. Grossglauser, and P. Thiran. Sub-matrix factorization for real-time vote prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2280–2290, 2020. [Cited on page 65]

[92] S. Javanmardi, D. W. McDonald, and C. V. Lopes. Vandalism detection in Wikipedia: A high-performing, feature-rich model and its reduction through lasso. In *Proceedings of WikiSym'11*, Mountain View, CA, USA, Oct. 2011. [Cited on page 16]

[93] Y. Jiang, B. Adams, and D. M. German. Will my patch make it? and how fast? case study on the Linux kernel. In *Proceedings of MSR 2013*, San Francisco, CA, USA, May 2013. [Cited on pages 28, 29, 32, and 61]

[94] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998. [Cited on page 62]

[95] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of ACL 2017*, July 2017. [Cited on pages 51 and 62]

[96] D. Karlis and I. Ntzoufras. Bayesian modelling of football outcomes: using the Skellam's distribution for the goal difference. *IMA Journal of Management Mathematics*, 20(2):133–145, 2009. [Cited on page 112]

[97] R. Kennedy, S. Wojcik, and D. Lazer. Improving election prediction internationally. *Science*, 355(6324):515–520, 2017. [Cited on page 84]

[98] M. E. Khan and W. Lin. Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. In *Proceedings of AISTATS 2017*, Fort Lauderdale, FL, USA, Apr. 2017. [Cited on page 102]

[99] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. [Cited on pages 17, 18, 69, and 72]

[100] A. Kreppel. What affects the European Parliament's legislative influence? an analysis of the success of EP amendments. *JCMS: Journal of Common Market Studies*, 37(3):521–537, 1999. [Cited on page 61]

[101] A. Kreppel. Moving beyond procedure: an empirical analysis of European Parliament legislative influence. *Comparative Political Studies*, 35(7):784–813, 2002. [Cited on page 61]

[102] 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. [Cited on page 84]

[103] V. Kristof, V. Quelquejay-Leclère, R. Zbinden, L. Maystre, M. Grossglauser, and P. Thiran. A user study of perceived carbon footprint. *arXiv preprint arXiv:1911.11658*, 2019. [Cited on page 87]

[104] V. Kristof, M. Grossglauser, and P. Thiran. War of words: The competitive dynamics of legislative processes. In *Proceedings of The Web Conference '20*, pages 2803–2809, 2020. [Cited on page 35]

[105] V. Kristof, A. Suresh, M. Grossglauser, and P. Thiran. War of words II: Enriched models for law-making processes. In *Proceedings of The Web Conference '21*, 2021. [Cited on pages 35 and 116]

[106] J. B. Kruskal. An overview of sequence comparison: Time warps, string edits, and macromolecules. *SIAM Review*, 25(2):201–237, 1983. [Cited on page 21]

[107] S. Kumar, R. West, and J. Leskovec. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In *Proceedings of the 25th international conference on World Wide Web*, pages 591–602, 2016. [Cited on page 1]

[108] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of ICUIMC'08*, Suwon, Korea, Jan. 2008. [Cited on pages 17 and 24]

[109] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. [Cited on page 2]

[110] M. K. Lee, D. Kusbit, A. Kahng, J. T. Kim, X. Yuan, A. Chan, D. See, R. Noothigattu, S. Lee, A. Psomas, et al. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW): 1–35, 2019. [Cited on page 4]

[111] Z. Lefkofridi and A. Katsanidou. Multilevel representation in the European Parliament. *European Union Politics*, 15(1):108–131, 2014. [Cited on page 55]

[112] A. Levi, O. Mokryn, C. Diot, and N. Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of RecSys'12*, Dublin, Ireland, Sept. 2012. [Cited on pages 17 and 24]

[113] M. S. Lewis-Beck. Election forecasting: Principles and practice. *The British Journal of Politics and International Relations*, 7(2):145–164, 2005. [Cited on page 84]

[114] R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959. [Cited on pages 3, 8, and 9]

[115] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. [Cited on pages 59 and 75]

[116] D. J. MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992. [Cited on page 89]

[117] M. J. Maher. Modelling association football scores. *Statistica Neerlandica*, 36(3): 109–118, 1982. [Cited on pages 97, 108, and 112]

[118] C. F. Manski and D. McFadden. Alternative estimators and sample designs for discrete choice analysis. [Cited on page 3]

[119] H. Margetts, P. John, S. Hale, and T. Yasseri. *Political turbulence: How social media shape collective action*. Princeton University Press, 2015. [Cited on page 1]

[120] J. Marschak. Binary choice constraints on random utility indicators. 1959. [Cited on page 3]

[121] L. Maystre. Efficient learning from comparisons. Technical report, EPFL, 2018. [Cited on page 4]

[122] L. Maystre and M. Grossglauser. Fast and accurate inference of Plackett–Luce models. In *Advances in Neural Information Processing Systems 28*, Montréal, QC, Canada, Dec. 2015. [Cited on page 3]

[123] L. Maystre, V. Kristof, A. J. González Ferrer, and M. Grossglauser. The player kernel: Learning team strengths based on implicit player contributions. Preprint, `arXiv:1609.01176 [cs.LG]`, Sept. 2016. [Cited on page 96]

[124] L. Maystre, V. Kristof, and M. Grossglauser. Pairwise comparisons with flexible time-dynamics. pages 1236–1246, 2019. [Cited on page 93]

[125] G. McElroy and K. Benoit. Party policy and group affiliation in the European Parliament. *British Journal of Political Science*, 40(2):377–398, 2010. [Cited on page 47]

[126] D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, 1973. [Cited on pages 3, 8, and 95]

[127] D. McFadden. The measurement of urban travel demand. *Journal of public economics*, 3(4):303–328, 1974. [Cited on page 2]

[128] D. McFadden. Economic choices. *American Economic Review*, 91(3):351–378, 2001. [Cited on pages 1 and 4]

[129] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, Lake Tahoe, Nevada, USA, Dec. 2013. [Cited on page 50]

[130] V. Miller. How much legislation comes from europe? *Economic Indicators*, 7(10), 2010. [Cited on page 37]

[131] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001. [Cited on page 101]

[132] MIT Election Data and Science Lab. U.S. President 1976–2016, 2017. URL `https://doi.org/10.7910/DVN/42MVDX`. Accessed: 2021-06-15. [Cited on page 76]

[133] M. Mühlböck. National versus European: Party control over members of the European Parliament. *West European Politics*, 35(3):607–631, 2012. [Cited on page 55]

[134] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012. [Cited on pages 67 and 85]

[135] S. Negahban, S. Oh, and D. Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 5(1):266–287, 2017. [Cited on page 3]

[136] M. E. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006. [Cited on page 47]

[137] H. Nickisch and C. E. Rasmussen. Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008. [Cited on page 102]

[138] H. Nickisch, A. Solin, and A. Grigorievksiy. State space Gaussian processes with non-Gaussian likelihood. In *Proceedings of ICML 2018*, Long Beach, CA, USA, July 2018. [Cited on page 113]

[139] R. Noothigattu, S. Gaikwad, E. Awad, S. Dsouza, I. Rahwan, P. Ravikumar, and A. Procaccia. A voting-based system for ethical decision making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. [Cited on page 4]

[140] Norwegian Centre for Research Data. German parliamentary elections, 2020. URL `https://nsd.no/european_election_database/country/germany/parliamentary_elections.html`. Accessed: 2021-06-15. [Cited on page 77]

[141] C. of the Representatives of the Governments of the Member States. *Treaty of Lisbon Amending the Treaty on European Union and the Treaty Establishing the European Community*. European Union, 2007. [Cited on page 38]

[142] A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society, Series B*, 40(1):1–42, 1978. [Cited on page 113]

[143] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in. [Cited on page 2]

[144] E. Parliament. The ordinary legislative procedure, 2018. URL `https://www.europarl.europa.eu/olp/en/ordinary-legislative-procedure/overview`. Accessed: 2021-06-15. [Cited on page 38]

[145] E. Parliament. Rules of procedure of the european parliament - rule 174, 2018. URL `http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+RULES-EP+20180731+RULE-174+DOC+XML+V0//EN&language=EN&navigationBar=YES`. Accessed: 2021-06-15. [Cited on page 42]

[146] E. Parliament. Questions and answers on issues about the digital copyright directive, 2019. URL `https://www.europarl.europa.eu/news/en/press-room/20190111IPR23225/`. Accessed: 2021-06-15. [Cited on page 57]

[147] E. Parliament. Rules of procedure of the european parliament - rule 180, 2021. URL `https://www.europarl.europa.eu/doceo/document/RULES-9-2019-07-02-RULE-180_EN.html`. Accessed: 2021-06-15. [Cited on page 47]

[148] R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975. [Cited on page 3]

[149] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in Wikipedia. In *Proceedings of ECIR 2008*, Glasgow, Scottland, Apr. 2008. [Cited on page 16]

[150] 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. [Cited on page 84]

[151] G. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests.* Danmarks Pædagogiske Institut, 1960. [Cited on pages 14, 17, and 48]

[152] G. Rasch. *Probabilistic models for some intelligence and attainment tests.* ERIC, 1993. [Cited on page 9]

[153] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006. [Cited on pages 94, 98, and 104]

[154] S. Reece and S. Roberts. An introduction to Gaussian processes for the Kalman filter expert. In *Proceedings of ICIF 2010*, Edinburgh, UK, July 2010. [Cited on page 103]

[155] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000. [Cited on pages 14 and 16]

[156] M. H. Ribeiro, R. Ottoni, R. West, V. A. Almeida, and W. Meira Jr. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 131–141, 2020. [Cited on page 1]

[157] P. C. Rigby, D. M. German, and M.-A. Storey. Open source software peer review practices: A case study of the Apache server. In *Proceedings of ICSE'08*, Leipzig, Germany, May 2008. [Cited on page 28]

[158] 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. [Cited on page 84]

[159] K. T. Rodolfa, E. Salomon, L. Haynes, I. H. Mendieta, J. Larson, and R. Ghani. Case study: predictive fairness to reduce misdemeanor recidivism through social service interventions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 142–153, 2020. [Cited on page 1]

[160] Y. Saatçi. *Scalable Inference for Structured Gaussian Process Models*. PhD thesis, University of Cambridge, 2012. [Cited on pages 104 and 113]

[161] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. 2017. [Cited on page 3]

[162] M. J. Salganik and K. E. C. Levy. Wiki surveys: Open and quantifiable social data collection. *PLoS ONE*, 10(5):1–17, 2015. [Cited on pages 4, 87, and 95]

[163] S. Sarkar, B. P. Reddy, S. Sikdar, and A. Mukherjee. StRE: Self attentive edit quality prediction in Wikipedia. In *Proceedings of ACL 2019*, pages 3962–3972, 2019. [Cited on page 61]

[164] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of SIGIR'02*, Tampere, Finland, Aug. 2002. [Cited on pages 17 and 24]

[165] H. Sepehri Rad and D. Barbosa. Identifying controversial articles in Wikipedia: A comparative study. In *Proceedings of WikiSym'12*, Linz, Austria, Aug. 2012. [Cited on page 61]

[166] H. Sepehri Rad, A. Makazhanov, D. Rafiei, and D. Barbosa. Leveraging editor collaboration patterns in Wikipedia. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 13–22, 2012. [Cited on page 61]

[167] N. Silver. Pollster ratings v3.0, 2008. URL `https://fivethirtyeight.com/features/pollster-ratings-v30/`. Accessed: 2021-06-15. [Cited on pages 76 and 84]

[168] A. Solin. *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. PhD thesis, Aalto University, 2016. [Cited on pages 103 and 113]

[169] A. Solin and S. Särkkä. Explicit link between periodic covariance functions and state space models. In *Proceedings of AISTATS 2014*, Reykjavik, Iceland, Apr. 2014. [Cited on page 104]

[170] H. Stern. Are all linear paired comparison models empirically equivalent? *Mathematical Social Sciences*, 23(1):103–117, 1992. [Cited on pages 108 and 112]

[171] A.-A. Stoica, C. Riederer, and A. Chaintreau. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *Proceedings of the 2018 World Wide Web Conference*, pages 923–932, 2018. [Cited on page 1]

[172] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019. [Cited on page 1]

[173] R. Sumi, T. Yasseri, A. Rung, A. Kornai, and J. Kertész. Edit wars in Wikipedia. In *Proceedings of SocialCom 2011*, Boston, MA, USA, Oct. 2011. [Cited on page 61]

[174] Swiss Federal Statistical Office (via opendata.swiss). Real-time data on referenda on vote days, 2021. URL `https://opendata.swiss/en/dataset/echtzeitdaten-am-abstimmungstag-zu-eidgenoessischen-abstimmungsvorlagen`. Accessed: 2021-06-15. [Cited on page 80]

[175] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013. [Cited on page 103]

[176] B. Tabibian, I. Valera, M. Farajtabar, L. Song, B. Schölkopf, and M. Gomez-Rodriguez. Distilling information reliability and source trustworthiness from digital traces. In *Proceedings of WWW'17*, Perth, WA, Australia, Apr. 2017. [Cited on page 16]

[177] The Swiss Confederation. Democracy, 2019. URL `https://www.ch.ch/en/demokratie/`. Accessed: 2021-06-15. [Cited on page 66]

[178] The Swiss Confederation. Popular vote, 2019. URL `https://www.admin.ch/gov/en/start/documentation/votes.html`. Accessed: 2021-06-15. [Cited on page 66]

[179] L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4): 273–286, 1927. [Cited on pages 1, 3, 7, 94, 95, 97, and 112]

[180] L. L. Thurstone. The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology*, 21(4):384–400, 1927. [Cited on pages 3 and 87]

[181] K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, second edition, 2009. [Cited on pages 4 and 10]

[182] 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. [Cited on page 84]

[183] G. Tsebelis, C. B. Jensen, A. Kalandrakis, and A. Kreppel. Legislative procedures in the European Union: An empirical analysis. *British Journal of Political Science*, 31(4):573–599, 2001. [Cited on page 61]

[184] 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. [Cited on page 84]

[185] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937. [Cited on page 1]

[186] A. M. Turing. Computing machinery and intelligence. In *Parsing the turing test*, pages 23–65. Springer, 2009. [Cited on page 1]

[187] E. Union. European data portal, 2021. URL `https://www.europeandataportal.eu/en`. Accessed: 2021-06-15. [Cited on page 36]

[188] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. [Cited on page 27]

[189] 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. [Cited on page 84]

[190] M. Vojnovic and S.-Y. Yun. Parameter estimation for generalized Thurstone choice models. In *Proceedings of ICML 2016*, New York, NY, USA, June 2016. [Cited on page 3]

[191] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2012. [Cited on page 99]

[192] F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*, pages 109–117. PMLR, 2013. [Cited on page 3]

[193] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, Vancouver, BC, Canada, Dec. 2010. [Cited on page 16]

[194] S. J. Westwood, S. Messing, and Y. Lelkes. Projecting confidence: How the probabilistic horse race confuses and demobilizes the public. *The Journal of Politics*, 82(4):1530–1544, 2020. [Cited on page 117]

[195] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, Vancouver, BC, Canada, Dec. 2009. [Cited on pages 16 and 23]

[196] Wikipedia. Wikipedia article depth, 2017. URL `https://meta.wikimedia.org/wiki/Wikipedia_article_depth`. Accessed: 2021-06-15. [Cited on page 20]

[197] Wikipedia. Wikipedia:Wikipedians, 2017. URL `https://en.wikipedia.org/wiki/Wikipedia:Wikipedians`. Accessed: 2021-06-15. [Cited on page 13]

[198] H. C. Williams. On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and planning A*, 9(3):285–344, 1977. [Cited on page 3]

[199] World Economic Forum. How much data is generated each day?, 2019. URL `https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/`. Accessed: 2021-06-15. [Cited on page 2]

[200] S. Wynes and K. A. Nicholas. The climate mitigation gap: education and government recommendations miss the most effective individual actions. *Environmental Research Letters*, 12(7), 2017. [Cited on page 87]

[201] A. B. Yardım, V. Kristof, L. Maystre, and M. Grossglauser. Can who-edits-what predict edit survival? In *Proceedings of KDD'19*, London, United Kingdom, Aug. 2018. [Cited on pages 13 and 61]

[202] T. Yasseri, R. Sumi, A. Rung, A. Kornai, and J. Kertész. Dynamics of conflicts in Wikipedia. *PloS one*, 7(6), 2012. [Cited on page 61]

[203] T. Yasseri, A. Spoerri, M. Graham, and J. Kertész. The most controversial topics in Wikipedia: A multilingual and geographical analysis. In P. Fichman and N. Hara, editors, *Global Wikipedia: International and Cross-Cultural Issues in Online Collaboration*. Scarecrow Press, 2014. [Cited on page 26]

[204] J. I. Yellot, Jr. The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977. [Cited on page 3]

[205] P. Yin, G. Neubig, M. Allamanis, M. Brockschmidt, and A. L. Gaunt. Learning to represent edits. In *International Conference on Learning Representations*, 2018. [Cited on page 62]

[206] E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1928. [Cited on pages 1, 3, 7, 14, 94, 97, and 112]

[207] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015. [Cited on page 62]

[208] D. Zhou, S. Basu, Y. Mao, and J. C. Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems 25*, Lake Tahoe, CA, USA, Dec. 2012. [Cited on page 16]

Victor Kristof

Ph.D. Student in Computer Science

EPFL IC INDY, Station 14
1015 Lausanne, Switzerland
☎ *+41 78 835 93 03*
✉ *victor.kristof@epfl.ch*
victorkristof.me

Profile

- Six-year research experience in **machine learning** and **data mining**
- Strong skills in **probabilistic modeling**, **data visualization**, and **software engineering**
- I develop **interpretable models** of voting, law-making, sports, peer-production, and human perception

Education

2015–2021 **Ph.D. in Computer Science**, *Ecole Polytechnique Fédérale de Lausanne (EPFL)*, Switzerland.
- Advised by Prof. Patrick Thiran and Prof. Matthias Grossglauser in the *Information and Network Dynamics Lab*
- Thesis title: *Discrete-Choice Mining of Social Processes*

2017 & 2018 (4 months) **Visiting Ph.D. Student**, *Columbia University*, New York City, USA.
- Visited Prof. Augustin Chaintreau in the *Mobile Social Lab* during the Summer

2009–2015 **B.Sc. and M.Sc. in Communication Systems**, *EPFL*, Switzerland.
- **GPA: 5.47/6** (Ranking: 3/53)
- Thesis title: *Mining and Modeling Real-Time Communications to Enable Smarter Business Interactions*

2011–2012 **Exchange Year**, *Tecnológico de Monterrey*, Mexico.

Projects

2019–2021 **Climpact**, www.climpact.ch.
- Designed a Bayesian model to estimate people's perception of their carbon footprint from pairwise comparisons

2016–2021 **Kickoff.ai**, www.kickoff.ai.
- Designed a Bayesian model to predict the outcomes of soccer matches outperforming the betting odds
- Developed a web platform to provide predictions for the top-5 European leagues (featured in the media)

2014–2021 **Predikon**, www.predikon.ch.
- Designed a statistical model able to predict voting results with less than 1% error
- Developed a web platform for visualizing voting patterns and predicting voting results (featured in the media)

2014 **Sezam**, www.sezam.ch.
- Developed an iOS application (Objective-C) to help users manage entrance-door codes
- Application downloaded by Mom, Dad, and 3000+ other people

Professional Experience

2015 (6 months) **Data Scientist Intern**, *Interact.io*, Berlin.
- Designed a classifier (96% accuracy) to discriminate human-created emails from automatically generated ones
- Integrated into a web application and deployed to production (Python, HTML/CSS, JavaScript)

2013–2014 (20 months) **Mac OS X Developer**, *Pryv*, Lausanne.
- Developed a macOS application to enable users to upload their data to the company service (Objective-C)

2012 (6 months) **Web Developer**, *AVNTK*, Guadalajara.
- Increased traffic by 150% by adapting their translation tool (from English to Basic English) into a web application

Skills

Data Science: Stochastic Optimization, Representation Learning, Active Learning, Bayesian Statistics, Databases

Programming: Python, Torch, TensorFlow, Objective-C, Java, Scala, JavaScript, HTML/CSS, Bash, LaTeX, SQL, Vim

Other Commitments & Interests

Association **President**, *Swiss Youth for Climate*.
- Active board member for four years in a 250-member NGO advocating and raising awareness on climate change
- Raised $100,000. Organized and presented at conferences for 100+ persons. Managed groups of 5 to 15 people.
- Participated in the UN conference on climate change (COP22 and COP24) as delegate and civil-society observer
- Increased members by 85% and supporters by 140% during two years of Presidency

Grants
- Google and Microsoft Travel Grant for the *Climate Change AI Workshop* at NeurIPS 2019
- EPFL Tech Transfer Office Grant ($15,000) to develop Kickoff.ai

Services
- Co-organizer of the "AI & Democracy" Track at Applied Machine Learning Days 2021.
- Reviewer for AAAI 2020. Member of the faculty teaching committee. Webmaster for my lab website.

Sports: I played soccer for 14 years. I ran a 32-kilometer trail and two half-marathons. Now, I bike and sail.

Languages: **French** (mother tongue), **English** (fluent), **Spanish** (fluent), and **German** (good knowledge)

Publications

I am the first or co-first author in all publications.

WWW 2021 **War of Words II: Enriched Models of Law-Making Processes**
V. Kristof, A. Suresh, M. Grossglauser, P. Thiran

KDD 2020 **Sub-Matrix Factorization for Real-Time Vote Prediction**
A. Immer, V. Kristof, M. Grossglauser, P. Thiran
Oral presentation with 5% acceptance rate

WWW 2020 **War of Words: The Competitive Dynamics of Legislative Processes**
V. Kristof, M. Grossglauser, P. Thiran

CCAI 2019 **A User Study of Perceived Carbon Footprint**
V. Kristof, V. Quelquejay, R. Zbinden, L. Maystre, M. Grossglauser, P. Thiran
Climate Change AI Workshop (NeurIPS 2019)

KDD 2019 **Pairwise Comparisons with Flexible Time-Dynamics**
L. Maystre, V. Kristof, M. Grossglauser
Oral presentation with 10% acceptance rate

KDD 2018 **Can Who-Edits-What Predict Edit Survival?**
B. Yardim, V. Kristof, L. Maystre, M. Grossglauser
Oral presentation with 10% acceptance rate

MLSA 2016 **The Player Kernel: Learning Team Strengths Based on Implicit Player Contributions**
L. Maystre, V. Kristof, A. J. González Ferrer, M. Grossglauser
Machine Learning and Data Mining for Sports Analytics Workshop (ECML-PKDD 2016)

Teaching

I was a teaching assistant for the following classes.

2017–2021 **Stochastic Models for Communication Systems** (Bachelor)

2019 **Probability and Statistics** (Bachelor)

2017–2018 **Internet Analytics** (Bachelor)

2015–2017 **Machine Learning** (Master)

Student Projects Supervision

2021 **Mining European Parliament Speeches**
Mahmoud Sellami (Master)

2021 **Vote Prediction from Swiss Voting Booklets**
Yasser Haddad (Master)

2021 **Lawgit: An Interactive Tool to Visualize the Amendment Process in the European Union**
Max Stieber (Master)

2021 **Understanding the Dynamics of Delegations to International Climate Negotiations**
Jan Linder (Bachelor)
Manuscript in preparation

2021 **Who Makes Law? Understanding the Structure of Lobbying in Brussels**
Antoine Magron (Bachelor), co-supervised with Aswin Suresh

2020–2021 **The Carbon Footprint of a Swiss Citizen**
Alexis Barrou, Edouard Catting, and Blanche Dalimier (Master), co-supervised with Dr. Jérôme Payet

2020 **Mining Party Interventions to International Climate Negotiations**
Tatiana Cogne (Bachelor)
Manuscript in preparation

2019 **Visualizing Voting Patterns in Switzerland**
Ragnor Comerford (Bachelor)

2018–2019 **Mining and Modelling Swss Referendum Votes**
Alexander Immer (Master)
Published at KDD 2020

2019 **A User Study of Perceived Carbon Footprint**
Valentin Quelquejay-Leclère and Robin Zbinden (Bachelor), co-supervised with Lucas Maystre
Published at the *Climate Change AI Workshop* (NeurIPS 2019)

2018 **Linear Models for Legislative Edit Predictions**
Guillaume Mollard (Master)

2018 **Text Models for Legislative Edit Predictions**
Brune Bastide (Master)

2018 **Graph Embedding for Hybrid Networks**
Khuram Javed (Master), co-supervised with Sébastien Henri

2018 **How Do Fake News Go Viral?**
Julie Djeffal (Master), co-supervised with William Trouleau

2017 **Predictive Models of Edits in Peer-Production Systems**
Batuhan Yardim (Master), co-supervised with Lucas Maystre
Published at KDD 2018

2017 **How Do Tweets Relate to Political Polls?**
Antoine Mougeot (Bachelor), co-supervised with William Trouleau

2016 **Gaussian Process Classification for Predicting Football Matches**
Antonio Gonzalez Ferrer (Master), co-supervised with Lucas Maystre
Published at the *Machine Learning and Data Mining for Sports Analytics Workshop* (ECML-PKDD 2016)

Talks

2021 **Predikon: Sub-Matrix Factorization for Real-Time Vote Prediction**
Open Government Data Workshop at the Swiss Federal Statistical Office

2021 **Round Table: The Role of Artificial Intelligence in Politics**
Panelist at the Open Geneva Festival

2021 **War of Words: The Competitive Dynamics of Legislative Processes**
Annual Congress of the Swiss Political Science Association (SPSA)

2021 **War of Words: The Competitive Dynamics of Legislative Processes**
Chair of Governance and Regulation (Paris-Dauphine)

2021 **Discrete Choice Mining of Social Processes**
ETHZ Computational Social Science Lab

2020 **Climpact: A User Study of Perceived Carbon Footprint**
ETHZ Workshop "Data Science in Climate and Climate Impact Research"

2019 **Artificial Instincts: Will There Ever Come a Day When There's No Reason to Watch Sports?**
Dell Technologies "AI: Hype or Reality" Podcast

2018 **Who Will Win? Predicting in a Dynamic World**
SoftwareONE

2018 **Discrete Choice Models for Data Mining**
Invited Lecture for EPFL's *Database Systems* Course

2018 **Une société artificiellement intelligente**
Tribune de Genève (Opinion)

2017 **Data Science and Politics**
Economiesuisse

Press

2021 **Machine Learning is Becoming a Technology**
Geneva Solutions

2021 **A Journey With Predikon**
Center for Digital Trust (C4DT)

2020 **Predikon: Dieses Tool soll die Resultate heute exakt vorhersagen**
Nau.ch

2018 **Millennials, peut-on leur laisser les clés?**
Radio Télévision Suisse (RTS Infrarouge)

2018 **Un algorithme de l'EPFL prédit qui va remporter le Mondial de football**
Radio Télévision Suisse (RTS Le 12H30)

2016 **Des algorithmes pour prédire l'avenir?**
ARTE Futuremag

2016 **L'intelligence artificielle envahit l'Euro 2016**
Radio Télévision Suisse (RTS La Matinale)

2016 **Artificial intelligence predicts Euro 2016 match results**
Swissinfo

2016 **Euro 2016: AI 'Robot Oracle' predicts France and England wins**
Metro

2014 **Ebikon, la commune qui vote plus suisse que les autres**
Le Temps

2014 **Der Traum von «Predikon»**
Tages-Anzeiger

2014 **Die politische Swissminiature**
Neue Zürcher Zeitung