Human-machine Collaboration in Real-World Machine-Learning Applications

Claudia Veronica Roberts

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Professor Arvind Narayanan

January 2023

Abstract

Automation tools like machine learning are a necessity in our big data world. Thanks to the Internet and advancements in all facets of computer and storage technology, almost everyone has a voice in the Internet connected world. However, there are still very real physical limits in our physical world. This dichotomy—the seemingly limitless nature of technology enabled data colliding with the physical limits of the real world—has made automation tools a necessity, and predictive models powered by machine learning algorithms are one such tool.

The promise of machine learning to accurately predict future human behavior and human preferences has lead practitioners and researchers alike to apply machine learning automation tools to tasks such as product recommendations and speculatory activities such as long term job applicant success. However, due to the mercurial nature of humans, developing mathematical intermediaries to attempt to model and predict human behavior is challenging and not a straight-forward task. One way of harnessing the power of machine-learning backed automation to help reduce the scale of many real-world applications in more challenging domain settings is by having humans and machines collaborating in non-trivial ways. In this dissertation, we delineate the various ways in which humans and machines collaborate in challenging real-world applications. Moreover, we highlight three specific ways in which we can use human-machine collaboration to keep or increase utility and reduce real-world harm when using these systems in the wild: (*i*) humans enabling computers with domain specific knowledge, (*ii*) computers providing humans with algorithmic explanations, (*iii*) humans and computers working together in decision making.

Acknowledgements

Advisor: Arvind Narayanan

Collaborators: Ehtsham Elahi, Tony Jebara, Ashok Chandrashekhar, Maria Dimakopoulou, Qifeng Qiao

Committee members: Adji Bousso Dieng and Barbara Engelhardt (Readers); Andrés Monroy-Hernández and Matt Salganik (Examiners)

Institutions: Princeton University, Princeton Computer Science Department, CS Staff, Center for Information Technology Policy (CITP), Princeton Department of Sociology, Netflix, Inc., Adobe Research, Spotify, Hyperglue

Recommendation letter writers: Arvind Narayanan, Jennifer Rexford, Howard Miller, Nick Porcino, Katherine Breeden, Tony Jebara, Maria Dimakopoulou, Jaswinder Pal Singh, Walter Chang, Thomas Funkhouser, Matt Salganik

Advocates: Julie Yun, Robert Fish, Chris Kuenne, Zhihong Ding, Sara Perkovic, Andrew Appel, Karen Rouse

Funding: NDSEG Fellowship, National GEM Consortium Fellowship, Siebel Scholarship, NPSC Fellowship

Thank you God for all the precious guardian angels you have placed in my life in the form of family, friends, partners, collaborators, mentors, professors, colleagues, staff, and community members; they have shrouded me in love and light and guided my steps.

Special thanks: Dwight C. and Rosa G. Roberts, Laura M. Roberts, Stefan Wójcik, William Massey, Nicki Mahler, Jean Butcher, Laura Cummings-Abdo, Melissa Moss, Wesley L. Harris Scientific Society, Crossfit Ex Novo

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automation tools like machine learning are a necessity in our big data world. Thanks to the Internet and advancements in all facets of computer and storage technology, almost everyone has a voice in the Internet connected world. This near global accessibility to technology means instead of hundreds of applications for a job, there are thousands, instead of thousands of artists and musicians uploading content there are millions, instead of having a handful of participants filling out surveys there are thousands or diverse participants. However, there are still very real physical limits in our physical world. There are still only a finite number of hours in a day, a fixed number of dorm room beds available at a university, fixed screen real estate on a phone or laptop, and a limited number of qualified personnel to review insurance claim applications. This dichotomy—the seemingly limitless nature of technology enabled data colliding with the physical limits of the real world—has made automation tools a necessity, and predictive models powered by machine learning algorithms are one such tool.

Machine learning automation has been applied to a wide array of domains with varying levels of success. Machine learning automation has been applied very successfully in areas such as computer vision, robotics, and algorithmic help desk assistants, for example. The promise of machine learning to accurately predict future human behavior and human preferences has also lead practitioners and researchers alike to apply machine learning automation tools to tasks such as product recommendations and speculatory activities such as long term job applicant success. However, due to the mercurial nature of humans, developing mathematical intermediaries to attempt to model and predict human behavior is challenging and not a straight-forward task. One way of harnessing the power of machine-learning backed automation to help reduce the scale of many real-world applications in more challenging domain settings is by having humans and machines collaborating in non-trivial ways. In this dissertation, we delineate the various ways in which humans and machines collaborate in challenging real-world applications. Moreover, we highlight three specific ways in which we can use human-machine collaboration to keep or increase utility and reduce real-world harm when using these systems in the wild: (*i*) humans enabling computers with domain specific knowledge, (*ii*) computers providing humans with algorithmic explanations, (*iii*) humans and computers working together in decision making.

This dissertation is structured as follows:

Chapter 2: This chapter shows how the first mode of human-machine collaboration—humans enabling computers with domain specific knowledge—increases utility of machine learning models in the task of grade point average (GPA) prediction.

The Fragile Families Challenge is a mass collaboration social science data challenge whose aim is to learn how various early childhood variables predict the long-term outcomes of children. We describe our two-step approach to the Fragile Families Challenge. In step 1, we use a variety of fully automated approaches to predict child academic achievement. We fit 124 models, which involve most possible combinations of 8 model types, 2 imputation strategies, 2 standardization approaches, and 2 automatic variable selection techniques using 2 different thresholds. Then, in step 2, we attempt to improve on the results from step 1 with manual variable selection based on a detailed review of the codebooks. We manually selected 3,694 variables believed to be predictive of academic achievement, using a comprehensive review of student success literature to guide the decision-making process. The best models from step 1 were re-estimated using the manually selected variables. We show that manual variable selection improved the majority of the top 10 models in step 1, but did not improve the best of the top 10. Results indicate that variable selection inspired by social science methodologies can, in most cases, significantly improve models trained completely automatically.

Chapter 3: This chapter shows how the second mode of human-machine collaboration—computers providing humans with algorithmic explanations—can improve utility and reduce harm of machine learning models in the task of movie recommendations.

We evaluate two popular local explainability techniques, LIME and SHAP, on a movie recommendation task. We discover that the two methods behave very differently depending on the sparsity of the data set. LIME does better than SHAP in dense segments of the data set and SHAP does better in sparse segments. We trace this difference to the differing bias-variance characteristics of the underlying estimators of LIME and SHAP. We find that SHAP exhibits lower variance in sparse segments of the data compared to LIME. We attribute this lower variance to the completeness constraint property inherent in SHAP and missing in LIME. This constraint acts as a regularizer and therefore increases the bias of the SHAP estimator but decreases its variance, leading to a favorable bias-variance trade-off especially in high sparsity data settings. With this insight, we introduce the same constraint into LIME and formulate a novel local explainabilty framework called Completeness-Constrained LIME (CLIMB) that is superior to LIME and much faster than SHAP.

Chapter 4: This chapter shows a second example of the second mode of human-machine collaboration—computers providing humans with algorithmic explanations.

When generating local explanations of an opaque machine learning model by a variety of methods, we run into the problem of evaluating the explanations and determining the best one. We argue that evaluating an explanation of a model prediction has two components, faithfulness of the explanation to the opaque model and ease of human understanding of the explanation. In this work, we aim to develop quantitative ways to evaluate the faithfulness of the local explanations when explaining state-of-the-art movie recommendation models. We propose the quantitative evaluation of faithfulness in terms of an approximation error between the local explanation and the opaque model. We show that this approximation error can be minimized to obtain a new local explanation technique. The proposed approximation error is an intuitive way to reason about the behavior of local explanation methods compared to the axiomatic approach adopted in the local explainability research. Therefore we use the proposed approximation error to compare widely used local explanation methods in terms of their faithfulness/fidelity to the opaque model. Finally for the ease of human understanding component, we describe different ways to present results of an explanation model in terms of simplified feature inputs by optimizing the same approximation error in the transformed feature space.

Chapter 5: This chapter shows how the third mode of human-machine collaboration—humans and computers working together in decision making—maintains utility while reducing harm of machine learning models in the task of image personalization.

Personalization is an integral part of most web-service applications and determines which experience to display to each member. A popular algorithmic framework used in industrial personalization systems are contextual bandits, which seek to learn a personalized treatment assignment policy in the presence of treatment effects that vary with the observed contextual features of the members. In order to keep the optimization task tractable, such systems can myopically make independent personalization decisions that can conspire to create a suboptimal experience in the aggregate of the member's interaction with the web-service. We design a new family of online learning algorithms that benefit from personalization while optimizing the aggregate impact of the many independent decisions. Our approach selectively interpolates between any contextual bandit algorithm and any context-free multi-armed bandit algorithm and leverages the contextual information for a treatment decision only if this information promises significant gains over a decision that does not take it into account. Apart from helping users of personalization systems feel less targeted, simplifying the treatment assignment policy by making it selectively reliant on the context can help improve the rate of learning. We evaluate our approach on several datasets including a video subscription web-service and show the benefits of such a hybrid policy.

Chapter 6: In this chapter, we delineate the various ways in which humans and machines collaborate in the challenging real-world applications of the Fragile Families Challenge and recommender systems.

In this thesis, we cover three specific modes of human-machine collaboration in the aforementioned two contexts. However, upon doing a literature review of the full set of 17 research papers submitted to the Fragile Families Challenge and a literature review of existing recommender system research papers, we are able to draw a more complete picture of the various ways in which humans and machines collaborate in these settings.

Chapter 7: I conclude in this chapter. Why are humans so keen to collaborate with machines in the automatic machine-learning backed processing of real-world big data? Because their ultimate goal is to be able to read the minds of other humans. If companies and governments can read the minds of the people, then they can accurately predict their behavior and preferences in the future.

Chapter 2

Humans Enabling Computers with Domain Specific Knowledge[1]

This chapter is based on "Friend Request Pending: A Comparative Assessment of Engineering and Social Science Inspired Approaches to Analyzing Complex Birth Cohort Survey Data." It shows how the first mode of human-machine collaboration—humans enabling computers with domain specific knowledge—increases utility of machine learning models in the task of grade point average (GPA) prediction.

The Fragile Families Challenge is a mass collaboration social science data challenge whose aim is to learn how various early childhood variables predict the long-term outcomes of children. We describe our two-step approach to the Fragile Families Challenge. In step 1, we use a variety of fully automated approaches to predict child academic achievement. We fit 124 models, which involve most possible combinations of 8 model types, 2 imputation strategies, 2 standardization approaches, and 2 automatic variable selection techniques using 2 different thresholds. Then, in step 2, we attempt to improve on the results from step 1 with manual variable selection based

[1] This chapter was originally published with the following citation: Claudia V. Roberts. "Friend Request Pending: A Comparative Assessment of Engineering and Social Science Inspired Approaches to Analyzing Complex Birth Cohort Survey Data." In *Socius: Sociological Research for a Dynamic World.* 2019.

on a detailed review of the codebooks. We manually selected 3,694 variables believed to be predictive of academic achievement, using a comprehensive review of student success literature to guide the decision-making process. The best models from step 1 were re-estimated using the manually selected variables. We show that manual variable selection improved the majority of the top 10 models in step 1, but did not improve the best of the top 10. Results indicate that variable selection inspired by social science methodologies can, in most cases, significantly improve models trained completely automatically.

2.1 Overview

The Fragile Families and Child Wellbeing Study (FFCWS) is a longitudinal, birth cohort study run by researchers at Princeton University and Columbia University [95]. The Study follows a group of nearly 5,000 American children born between 1998 and 2000 and includes a large oversample of children born to unmarried parents [76]. The aim of the study is to characterize the relationships and conditions of unmarried parents and to study the cognitive development, mental and physical health, and social relationships of children born into such families.

The Fragile Families Challenge (FFC) is a mass collaboration social science data challenge designed to harness the predictive power of the FFCWS dataset [83]. The FFC invites community members to use the data to build models that best predict six key outcomes: grade point average (GPA), grit, material hardship, eviction, job loss, and job-training. In this paper, we focus on predicting GPAs only. It is our personal belief that a child's GPA is very important as it sets the tone for the rest of a child's life and influences the range of opportunities afforded to the child (e.g., college acceptances, scholarships, admittance into competitive summer enrichment programs).

Out-of-the-box machine learning libraries such as SciKitLearn and access to open datasets hosted on popular platforms such as Kaggle enable users from across the globe to create sophisticated predictive models with sometimes impressive predictive accuracy without ever needing to understand the underlying data [71][1]. This is in stark contrast to traditional methods of predictive modeling and data analysis undertaken by researchers in non-engineering fields, specifically the social sciences. In survey research, a popular measurement technique used in applied social research, the data is oftentimes very complex [3]. They can span over many years, in the case of longitudinal studies, and are susceptible to various sources of error: coverage error, sampling error, non-response error, and measurement error [105]. Thus, best practices in survey research call researchers to spend substantial time with the data–to "make friends with their data"–and to refrain from "throwing their data into a computer and trying to analyze it in minutes" [108]. Failure to do so could lead to spurious results and misleading conclusions, and researchers run the risk of misidentifying associations as statistically significant [46].

McFarland and colleagues argue that while sociologists are driven by theory and the desire to explain the patterns observed in the data, engineers are focused on creating algorithmic tools to increase the predictive accuracy of their models, without placing much importance on the explanation [63]. But what if the only metric of success is predictive accuracy? To what extent would an engineer be rewarded for "befriending" the data? Using the Fragile Families Challenge (FFC) as our backdrop, we seek to answer whether engineers get better predictive results when they spend a little time learning the domain they are working in, and if so, how much better are these results.

In this paper, we will use the term "variables" to refer to survey questions in the codebook. The cookbook survey questions are our independent variables. We use the term "outcome" to refer to the dependent variable we are attempting to predict, which in this case is GPA. “Fitting” or “estimating a model” is the process by which we learn a mathematical relationship between a set of variables x and the dependent variable y. The term “sample” refers to a single observation or data point in the dataset, which in this case is a child.

We divided the project into 2 steps. In step 1, we used a completely automatic approach that does not consider the data (the norm in data mining) to fit 124 models for GPA prediction. In step 2, we attempt to improve upon our results. We use a strategy that combines engineering-centric statistical analysis techniques with classical, more manual social science methodologies: we examined each variable in the codebook, manually selecting the ones believed to be predictive of academic achievement based on a non-expert reading of domain-specific research. Results indicate that it in most cases, it pays off for engineers to “make friends” with the FFCWS codebooks. We were able to improve the predictive accuracy of 6 of the 10 top step 1 models, of which 4 saw significant improvements. However, manual variable selection did not improve the predictive ability of the 2 most accurate models from step 1.

In Section 2.2, we describe the procedures used to create the initial set of 124 models. Section 2.3 describes the process of creating the 15 manually curated variable sets. Section 2.4 is a presentation of the results; we show that we were able to improve the predictive ability of almost all the models and demonstrate the effect of each variable subset on the models. In Section 2.5, we look at the variables that most predict GPA as identified by the two most accurate models from this project. Finally, we end the discussion with closing remarks in Section 2.6. Additional supporting materials can be found at the supporting online appendix.

2.2 Step 1: Automatic Variable Selection

The goal of step 1 was to fit a model that could predict year-15 GPAs as accurately as possible using a purely automated approach.

2.2.1 Data Preprocessing

With 2,121 samples and 12,942 variables, the FFC dataset is a high-dimensional dataset. In settings where the number of variables far exceeds the number of samples, overfitting becomes a problem, and the learned model loses its ability to generalize [41]. Thus, it's important to preprocess the data to not only reduce the number of variables but to also handle missing values and standardize the data.

We tried many different approaches to data pre-processing. We tried almost all combinations of 4 different decisions: 2 types of automatic variable selection (F-test and mutual information) using 2 thresholds (10% and 20%), 2 types of imputation strategies (median and mode), and 2 standardization approaches (no standardization and standardization). Detailed information of the pre-processing steps can be found at the supporting online appendix.

2.2.2 Model Selection

We used the following 8 model types to fit a total of 124 models. This includes all possible combinations of 8 different model types, 2 types of automatic variable selection (F-test and mutual information) using 2 thresholds (10% and 20%), 2 types of imputation strategies (median and mode), and 2 standardization approaches (no standardization and standardization).

1. *Ordinary least squares linear regression* (OLS) [2]

[2] In the case where there were more variables than cases, SciKitLearn finds the minimum ℓ_2 norm solution via singular value decomposition [71].

2. *Least-angle regression** (LARS) [30]

3. *Ridge regression** (Ridge) [98]

4. *Elastic Net** (EN) [116]

5. *Orthogonal Matching Pursuit* (OMP) [17]

6. *Lasso regression** (Lasso) [97]

7. *Decision Tree regression* (DTR) [72]

8. *ε-Support Vector Regression with linear kernel** (SVR) [26]

The observant reader will notice that $8 \times 2 \times 2 \times 2 \times 2 = 128$ while we fit only 124. We fit Decision Tree models using only some type of automatic variable selection. We did not fit these models using the full variable set because decision trees are very susceptible to overfitting in high-dimensional settings such as this one, where the number of variables greatly outnumbers the number of samples [71]. This accounts for the missing 4 combinations [3].

2.2.3 Results

We used FFC holdout test set mean squared error (MSE) scores (FFC-HO-MSE) to evaluate the accuracy of the models. We chose the MSE metric because it is the metric used to rank and evaluate the predictive validity of the submissions made through the FFC web portal [83]. Results from step 1 are summarized in Table 2.1.

[3] The results for all 124 models can be found at the supporting online appendix.

2.3 Step 2: Manual Variable Selection

The goal of step 2 was to improve the predictive accuracy of the models generated in step 1 by combining the previous automatic approaches with manual ones inspired by survey research best practices.

2.3.1 Manual Variable Selection

Our first step in this second phase of the project was to get friendly with the codebooks. We went through each of the 12,942 variables, manually selecting the ones believed to be predictive of future academic achievement. To inform the decision-making process, we turned to a comprehensive review of student success literature, "What Matters to Student Success," a report commissioned for the National Postsecondary Education Cooperative (NPEC) in 2006 [47]. Specifically, we relied on the first section of the report, which discusses the effects of pre-college experiences on student success, such as family and peer support, academic preparation, motivation to learn, socioeconomic status, and demographics [47]. While the report is targeted at student success in college, research has shown that high school grades are also highly correlated with socioeconomic factors such as family income and educational attainment [118]. From the NPEC report, we collated a list of 57 pre-college factors that have been shown by social scientists to affect student success [4].

Next, we manually examined each variable in the codebook and made judgement calls to determine whether or not it was directly related to any one of the 57 factors. It should be noted that we did not calculate intercoder reliability [60]. Calculating and reporting the intercoder reliability of this manual process is an area for future work. The aftermath of this process was a custom set of 3,694 variables [5].

[4] The full list of 57 factors can be found at the supporting online appendix.

[5] The complete list of 3,694 variable labels can be found at the supporting online appendix.

In an effort to identify the particular groups of variables most predictive of academic achievement, we created 14 additional, more granular subsets from the manually selected set of 3,694 variables. For example, we created a variable set that contained only wave 3 variables and a different subset that contained only wave 5 variables.

We used a total of 16 variable sets in this project [6]: 1) the original set of 12,942 variables; 2) our manually curated set of 3,694 variables; and 3) 14 additional variable sets, each of which is a subset of the manually selected set of the 3,694 variables (wave 3 only, wave 5 only, etc.). Table 2.2 summarizes each of these 16 variable sets, and provides a shorthand label for each. We will use these shorthand labels to reference the various variable sets for the remainder of this paper.

2.3.2 Method

We re-estimated the 10 most accurate models from step 1 on each of the 15 manually created variable subsets to produce a total of 150 models in this second step of the project. We used the same data preprocessing procedures and imputation strategies used in step 1. As before, categorical variables were not identified and were not treated differently from the continuous ones. After data imputation, our manually curated variable set was reduced from 3,694 to 3,423 variables. The FFC submission pipeline remained the same.

2.4 Results

Manual variable selection indeed improved, and in some cases dramatically improved, the accuracy of the predictive models trained previously using purely automatic techniques. Table 2.3 shows that 8 of the 10 most accurate models were trained on the

[6] This count does not include the variable subsets created using SciKitLearn's automated univariate feature selection routine.

manually created variable sets. Figure 2.4 shows how substantially manual variable selection improved the FFC-HO-MSE values of the 3rd, 6th, 9th, and 10th most accurate models from step 1. After re-estimating model 6 on the ‘w5’ variable set, the model rose to become the second most accurate model across both phases of the project, according to FFC-HO-MSE. The accuracies of models 4, 5, and 7 were also improved, but the change in FFC-HO-MSE was more tempered. The two most accurate models from step 1 saw no improvement.

2.4.1 Effect of Specific Variable Groups on Model Accuracy

A secondary goal was to understand how the various variable groups affected the predictive accuracy of the models trained in step 2 (e.g., do wave 5 variables yield better results than wave 3 variables?). Figure 2.5 is a 16×10 heatmap of FFC-HO-MSE scores from the 10 most accurate step 1 models trained on each of the 16 variable sets from the project, including the full set of 12,942 variables (labeled ‘All’). The lower the MSE value and the darker the color, the better.

Variable sets ‘w1’, ‘w2’, and ‘w3’ appear to contain the weakest signal across almost all models, and variable set ‘w5’ appears to contain the strongest signal, closely followed by ‘w1_5’ and ‘w1_5_t_kind’. From ‘w1’ through ‘w5’ we see a gradual strengthening of color across several rows. This pattern and the previous observations suggest that later waves are more predictive of high school GPA than earlier waves. However, not all wave 5 data are created equally. Variable set ‘k’ contains variables asked of only the child in wave 5 and ‘t_k’ contains variables asked of the child and the child’s teacher in wave 5 [83]. Looking across both columns, we can visually see how FFC-HO-MSE values improved across more than half of the models when input from the teacher was removed. We see a similar phenomenon when comparing the ‘t_kind’ and ‘t_kind_k’ columns. The majority of the models seem to improve with added input from the child. It appears that no matter how attentive a parent, teacher, or caretaker may (or may not) be, only the child really knows what he or she is feeling and experiencing on a day-to-day basis. And many of questions asked of the child in wave 5 attempt to tease out precisely this, questions such as “Frequency kids picked on you or said mean things to you”, “I often feel lonely”, “Frequency kids take your things, like your money or lunch,” and “Amount of time on a weekday you watch TV and movies.”

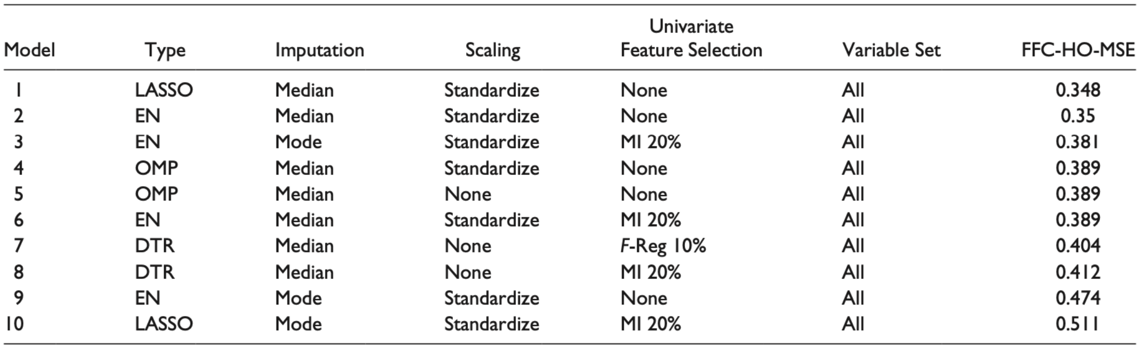
2.5 Variables That Most Predict GPA

An important goal of the FFC is to gain insight into the specific variables that most predict the 6 outcomes of interest–GPA, grit, material hardship, eviction, layoff, and job training. The hope being, that such insights may one day improve the lives of American children born into these “fragile families” [83]. Table 2.6 lists the variables that most predict year-15 GPAs according to the two most accurate models from this project based on FFC-HO-MSE scores. The most accurate model used Lasso, and the second most accurate model used Elastic Net. Coefficients are in parenthesis, and variables are listed in order of decreasing absolute coefficient value. Since the data were standardized, we were able to compare variable coefficients according to their relative significance to the prediction task. That is, the higher the absolute value of the coefficient, the higher the level of importance of that particular variable in predicting the desired outcome, which in this case is GPA. In prediction tasks such as these where we are predicting a real-valued outcome in a setting with multiple independent variables, coefficients can be interpreted as the following: holding all other variables fixed, the predicted outcomes increase (if the sign of the coefficient is positive) or decrease (if the sign of the coefficient is negative) by a factor of β_1 units for every one unit increase in x_1, where β_1 is the coefficient associated with the variable x_1 [22].

It is worth highlighting that while these two best models have almost equal predictive performance on the holdout data, 0.348 and 0.349 respectively, they exhibit very little overlap in the variables each model deems to be of most significance. We saw two different sets of variables returned by two different models of almost equal predictive accuracy, giving us two different pictures of which variables most predict year-15 GPAs. In his analysis of the two cultures of statistical modeling, Breiman argues that in a situation where "different models, all of them equally good...give different pictures of the relation between the predictor and response variables...the question of which one most accurately reflects the data is difficult to resolve" [15]. As engineers, these difficulties are further compounded by a lack of domain knowledge in the social sciences. Thus, we leave intuitive explanation of these results for future work and collaborations with social scientists. Furthermore, further research is required to calculate confidence intervals for the coefficients listed in this section and to begin interpreting the magnitude of the values and features returned.

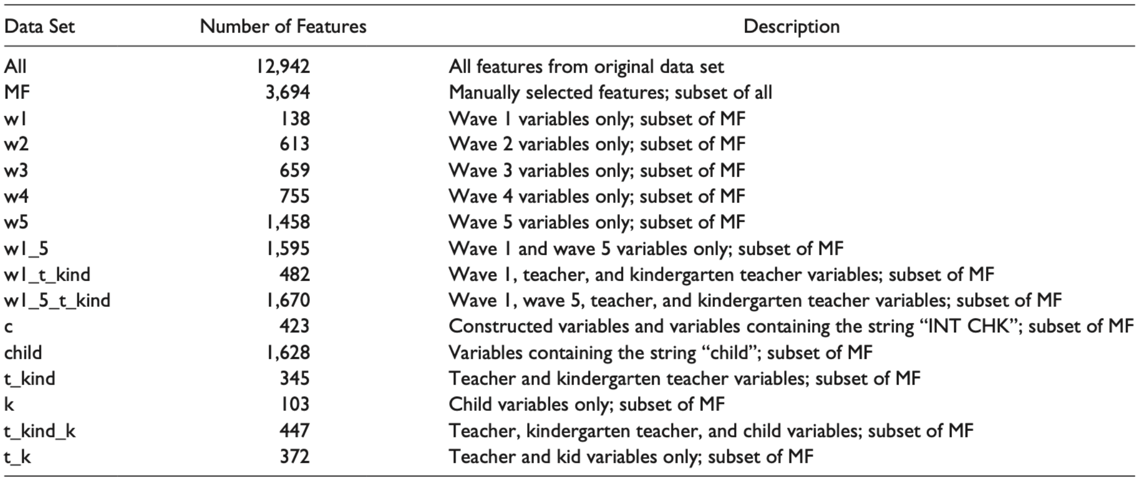
2.6 Summary

Using a two-step approach to the FFC, we were able to significantly improve the predictive accuracy of the majority of the models evaluated by using a combined approach of automatic and manual variable selection motivated by social science knowledge. We showed that such an approach, even though based on a non-expert reading of domain-specific research, can improve the accuracy of models trained automatically. We demonstrated that if one is not careful choosing their algorithms in such a data setting, then it pays to take a look at the codebooks. But there is still room for improvement, as our strategy was unable to improve the accuracy of the two most accurate models from step 1; this is an area for future work.



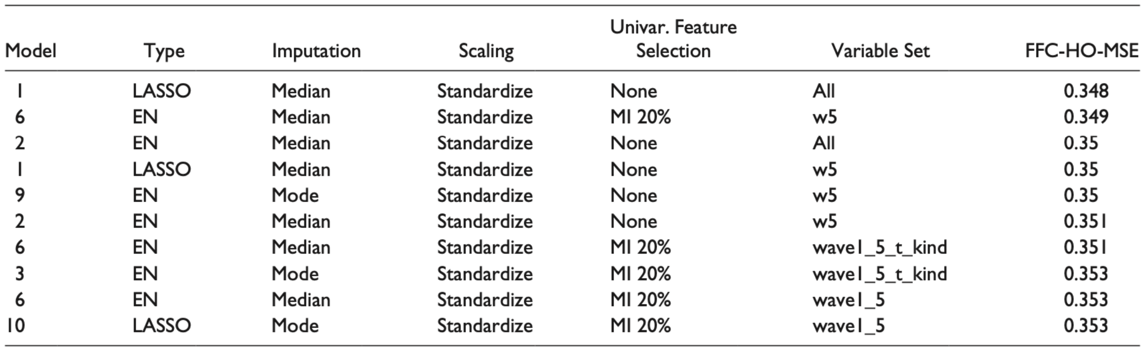
Model	Type	Imputation	Scaling	Univariate Feature Selection	Variable Set	FFC-HO-MSE
1	LASSO	Median	Standardize	None	All	0.348
2	EN	Median	Standardize	None	All	0.35
3	EN	Mode	Standardize	MI 20%	All	0.381
4	OMP	Median	Standardize	None	All	0.389
5	OMP	Median	None	None	All	0.389
6	EN	Median	Standardize	MI 20%	All	0.389
7	DTR	Median	None	*F*-Reg 10%	All	0.404
8	DTR	Median	None	MI 20%	All	0.412
9	EN	Mode	Standardize	None	All	0.474
10	LASSO	Mode	Standardize	MI 20%	All	0.511

Figure 2.1: Evaluation results for the 10 most accurate models from step 1. Models are numerically labeled and ordered by increasing FFC-HO-MSE. The lower the MSE the better.



Data Set	Number of Features	Description
All	12,942	All features from original data set
MF	3,694	Manually selected features; subset of all
w1	138	Wave 1 variables only; subset of MF
w2	613	Wave 2 variables only; subset of MF
w3	659	Wave 3 variables only; subset of MF
w4	755	Wave 4 variables only; subset of MF
w5	1,458	Wave 5 variables only; subset of MF
w1_5	1,595	Wave 1 and wave 5 variables only; subset of MF
w1_t_kind	482	Wave 1, teacher, and kindergarten teacher variables; subset of MF
w1_5_t_kind	1,670	Wave 1, wave 5, teacher, and kindergarten teacher variables; subset of MF
c	423	Constructed variables and variables containing the string "INT CHK"; subset of MF
child	1,628	Variables containing the string "child"; subset of MF
t_kind	345	Teacher and kindergarten teacher variables; subset of MF
k	103	Child variables only; subset of MF
t_kind_k	447	Teacher, kindergarten teacher, and child variables; subset of MF
t_k	372	Teacher and kid variables only; subset of MF

Figure 2.2: Descriptions of each of the 16 variable sets used in this project. This count does not include the variable subsets created using SciKitLearn's automated univariate feature selection routine. Includes the number of variables in each before imputation and the shorthand label used to reference each of the variable sets. These 16 variable sets include the original set of 12,942 variables, our manually curated set of 3,694 variables, and 14 additional variable sets, each of which is a subset of the manually selected set of 3,694 variables (wave 3 only, wave 5 only, etc.).



Model	Type	Imputation	Scaling	Univar. Feature Selection	Variable Set	FFC-HO-MSE
1	LASSO	Median	Standardize	None	All	0.348
6	EN	Median	Standardize	MI 20%	w5	0.349
2	EN	Median	Standardize	None	All	0.35
1	LASSO	Median	Standardize	None	w5	0.35
9	EN	Mode	Standardize	None	w5	0.35
2	EN	Median	Standardize	None	w5	0.351
6	EN	Median	Standardize	MI 20%	wave1_5_t_kind	0.351
3	EN	Mode	Standardize	MI 20%	wave1_5_t_kind	0.353
6	EN	Median	Standardize	MI 20%	wave1_5	0.353
10	LASSO	Mode	Standardize	MI 20%	wave1_5	0.353

Figure 2.3: Evaluation results from the 10 most accurate models across the entire project, i.e., steps 1 and 2 combined. Models are listed in order of increasing FFC-HO-MSE scores. The lower the MSE the better. The column 'Variable Set' contains the label name of the variable set used to train that particular model. Refer to Table 2.2 for a description of each variable set.

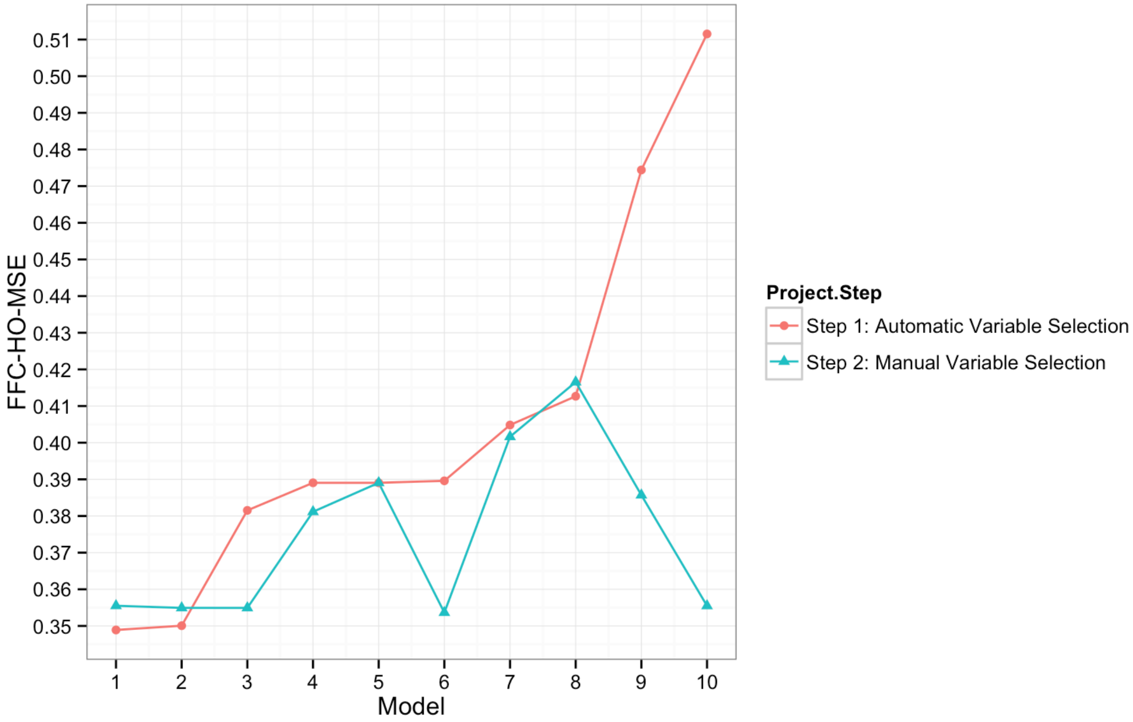


Figure 2.4: Effect of manual variable selection on the predictive ability of the 10 most accurate step 1 models. For the step 1 series, where the full set of 12,942 variables was used to fit the models, the MSE value is plotted for each model. Recall that in step 2, we re-estimated the top 10 step 1 models using each of the 15 manually created variable subsets (the full set of 3,694 manually curated variables plus 14 additional subsets taken from this set of 3,694 variables), giving us 15 MSE scores per model. Thus, for the step 2 series, for each model, we plot the holdout result based on the result with the best leaderboard score.

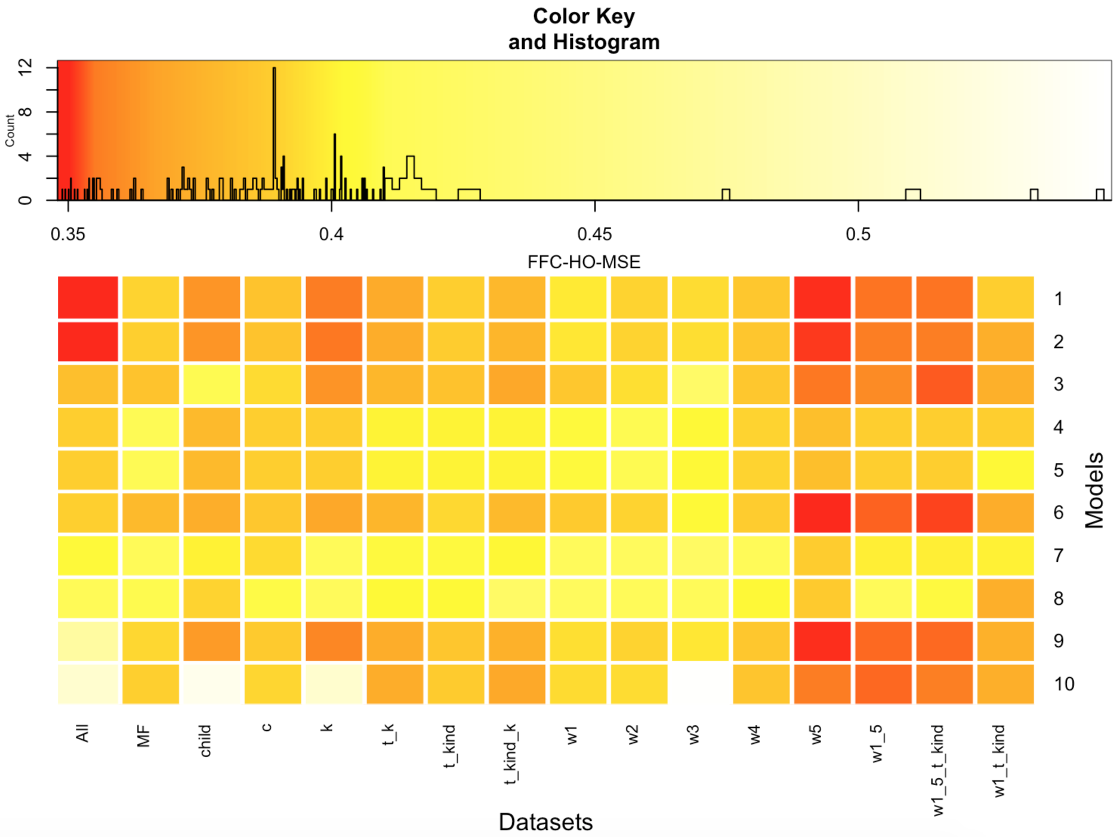
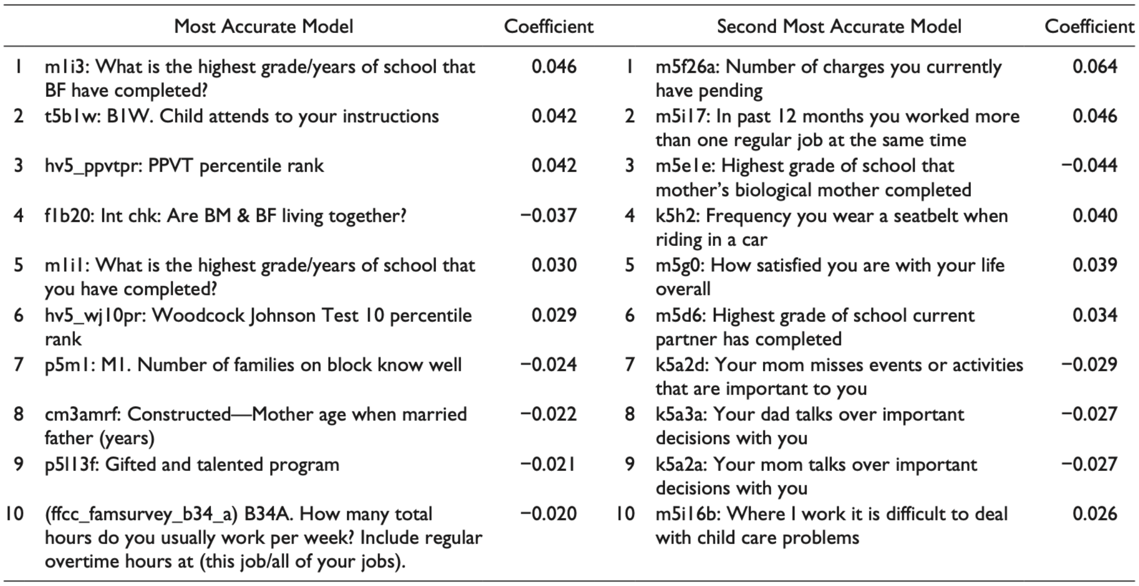


Figure 2.5: Heatmap of FFC-HO-MSE scores for the 10 most accurate step 1 models trained on each of the 16 variable sets from the project. The lower the MSE value and the darker the color the better. The lowest FFC-HO-MSE value, 0.348, is represented by the color red (Model 1, dataset 'All'). The highest FFC-HO-MSE value, 0.546, is represented by the color white (Model 10, dataset 'w3'). A baseline model that takes the mean of each outcome in the training data and predicts that mean value for all observations acheives an MSE value of 0.425 on the holdout data for GPA [83]. Refer to Table 2.2 for a description of each variable set.



	Most Accurate Model	Coefficient		Second Most Accurate Model	Coefficient
1	m1i3: What is the highest grade/years of school that BF have completed?	0.046	1	m5f26a: Number of charges you currently have pending	0.064
2	t5b1w: B1W. Child attends to your instructions	0.042	2	m5i17: In past 12 months you worked more than one regular job at the same time	0.046
3	hv5_ppvtpr: PPVT percentile rank	0.042	3	m5e1e: Highest grade of school that mother's biological mother completed	−0.044
4	f1b20: Int chk: Are BM & BF living together?	−0.037	4	k5h2: Frequency you wear a seatbelt when riding in a car	0.040
5	m1i1: What is the highest grade/years of school that you have completed?	0.030	5	m5g0: How satisfied you are with your life overall	0.039
6	hv5_wj10pr: Woodcock Johnson Test 10 percentile rank	0.029	6	m5d6: Highest grade of school current partner has completed	0.034
7	p5m1: M1. Number of families on block know well	−0.024	7	k5a2d: Your mom misses events or activities that are important to you	−0.029
8	cm3amrf: Constructed—Mother age when married father (years)	−0.022	8	k5a3a: Your dad talks over important decisions with you	−0.027
9	p5l13f: Gifted and talented program	−0.021	9	k5a2a: Your mom talks over important decisions with you	−0.027
10	(ffcc_famsurvey_b34_a) B34A. How many total hours do you usually work per week? Include regular overtime hours at (this job/all of your jobs).	−0.020	10	m5i16b: Where I work it is difficult to deal with child care problems	0.026

Figure 2.6: Variables that most predict GPA within the two most accurate models from the project. With a FFC-HO-MSE value of 0.348, the most accurate model used Lasso, median imputation, standardized variables, no additional univariate feature selection, and was trained on the full FFC feature set (variable set labeled 'All'). The second most accurate model, FFC-HO-MSE score of 0.349, used Elastic Net, median imputation, standardized variables, univariate feature selection (20%) using the mutual information scoring function, and was trained on the wave 5 ('w5') manual feature subset. Since there is some disagreement about how to produce confidence intervals around estimates that come from regularized models, we chose not to include them.

Chapter 3

Part 1: Computers Providing Humans with Algorithmic Explanations[1]

This chapter is based on “CLIME: Completeness-Constrained LIME.” It shows how the second mode of human-machine collaboration, computers providing humans with algorithmic explanations, can improve utility and reduce harm of machine learning models in the task of movie recommendations.

We evaluate two popular local explainability techniques, LIME and SHAP, on a movie recommendation task. We discover that the two methods behave very differently depending on the sparsity of the data set, where sparsity is defined by the amount of historical viewing data available to explain a movie recommendation for a particular data instance. We find that LIME does better than SHAP in dense segments of the data set and SHAP does better in sparse segments. We trace this difference to the differing bias-variance characteristics of the underlying estimators of LIME and SHAP. We find that SHAP exhibits lower variance in sparse segments

[1] This chapter was originally submitted with the following citation: Claudia V. Roberts, Ehtsham Elahi, and Ashok Chandrashekar. “CLIME: Completeness-Constrained LIME.”

of the data compared to LIME. We attribute this lower variance to the completeness constraint property inherent in SHAP and missing in LIME. This constraint acts as a regularizer and therefore increases the bias of the SHAP estimator but decreases its variance, leading to a favorable bias-variance trade-off especially in high sparsity data settings. With this insight, we introduce the same constraint into LIME and formulate a novel local explainabilty framework called Completeness-Constrained LIME (CLIME) that is superior to LIME and much faster than SHAP.

3.1 Overview

Recommendation systems mediate our various online interactions on a daily basis by limiting and influencing our possible choices. Recommender system use cases include product recommendations, search engines, social media browsing, music and video streaming, online advertising, news dissemination, job candidate matching, and real estate recommendations. The recommendation system problem setting is a high sparsity problem. Because the user only has prior information on a tiny subset of the total number of items at her disposable, the system has very little interaction data for the vast majority of the available items. This makes the recommendation setting an important and challenging problem domain (see Section 3.2).

In the recommender domain, explanations can be an integral part of the user product experience and depending on the recommendation task, critical to the task description itself. Explanatory models provide explanations for why the underlying recommendation system model made the item selection, item position ranking, or point prediction estimate that it did. In this paper, we focus on local explanations, that is, explanations for a single prediction instance. Two popular, general purpose explanation frameworks whose aim is to faithfully explain the local predictions of machine learning models are Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) (discussed in Section 3.3). LIME is very easy to use, computationally fast, and works on tabular data, images, and text [67]. While SHAP is computationally much slower than LIME depending on the underlying prediction model, it has some important theoretical guarantees such as guaranteeing the fair distribution of the prediction across the features [61, 67].

The first research question we sought to answer was how do SHAP and LIME perform in the high sparsity recommendation system setting. We adapted LIME and SHAP to the task of explaining movie recommendations and evaluated the explanations using the delta-rank metric (described in Section 3.5). We observed that while SHAP outperforms LIME on aggregate, the two methods behave very differently depending on the sparsity of the data, where sparsity is defined by the amount of historical viewing data available to explain a movie recommendation for a particular data instance. LIME does better than SHAP in dense segments of the data set, and conversely, SHAP outperforms LIME in the sparse regions of the data set. Dense segments of the data set include data instances with plentiful historical interaction and viewing data while the sparse regions include data instances with very little historical viewing data. We performed a bias-variance analysis and traced this difference in performance to the differing bias and variance characteristics of the underlying estimators of LIME and SHAP (see Section 3.4.1. We show that SHAP exhibits lower variance and higher bias compared to LIME and we postulate that this is the reason why SHAP outperforms LIME in high sparsity data settings where the bias-variance trade-off is especially favorable.

We hypothesize that the reason for SHAP's lower variance is due to Shapley values satisfying the efficiency property or what other papers call the completeness axiom [91], the conservation property [13], or summation-to-delta property [87] (for the duration of this paper we will refer to this property as the completeness constraint). Under the completeness constraint, SHAP's explanatory model is said to have a fair attribution of feature importance as it captures the contribution of each feature in the underlying model's prediction at data instance in question. We argue that this completeness constraint acts as a regularizer and therefore increases the bias and decreases the variance of the SHAP estimator. With these collective insights supported by our analysis, we introduce this constraint into LIME; we call this new local explainability technique Completeness-Constrained LIME (CLIME) (formulated in Section 3.4). Our experiments show that CLIME indeed lowers the variance of the LIME estimator and improves its performance in sparse data settings (results presented in Section 3.5). CLIME allows users to enjoy some of the theoretical guarantees of SHAP and maintain the off-the-shelf ease of LIME whilst being computationally faster than LIME and improving performance in high sparsity data settings, common in recommendation tasks.

Our contributions are summarized as follows:

- A comparison between SHAP and LIME in a movie recommendation setting, specifically analyzing their performance in sparse and dense regions of a publicly available data set

- A bias-variance analysis of SHAP and LIME in the sparse and dense data regions in a movie recommendation setting

- Formulation of a new model-agnostic, faithful, local explanation method called CLIME that includes one of the powerful properties of SHAP while being as fast as LIME and maintaining some of the desirable qualities of LIME

- Analysis connecting bias and variance to the completeness constraint

3.2 Problem Motivation

When determining what items to present to a user, these systems necessarily pare down the complete set of possible items from the millions to a small handful. The recommendation system problem setting is a high sparsity problem where the recommending system has very little interaction data between all the available users and all the available items [48, 42, 81, 8, 20]. Recommendation systems can also suffer from the long-tail phenomenon—there is an outsized amount of user interaction data for a tiny subset of the available item set and an extremely large number of items which effectively have no interaction data [52]. Further contributing to the high sparsity nature of online recommenders is the highly dynamic and in some cases transitory nature of the data. Users and product items are constantly coming and going, whether physically or in terms of relevancy, and user tastes are ever evolving.

An important aspect of recommendation systems is their corresponding explanatory models. This tight coupling of recommendation system models and explanation models is unique to the recommendation system setting. In the computer vision domain, explanations might come in the form of a visual saliency map that indicates the specific pixels that most contributed to the prediction of "cat" in an image classification task, for example. In the natural language processing task of sentiment analysis, an explanation model might highlight the particular words in a social media comment that most contributed to the comment being flagged as inappropriate by the model. In both of these cases, explanations serve largely as sanity checks to ensure that the learned mathematical model is picking up on the right features. Explanations in these artificial intelligence domains help build confidence that the trained machine learning models are doing the right thing for the right reasons and not picking up on spurious features.

The goals for providing explanations in recommendation systems and for sometimes explicitly exposing them to the user as a product feature are numerous and as follows: transparency, validation, trust building, persuasion, effectiveness, efficiency, satisfaction, communicating relevancy, comprehensibility, educating [99]. Previous studies have shown that accompanying recommendations with their explanations lead to higher user acceptance of recommendations though care must be taken because poorly designed explanations can be less performant than the base case of no explanations at all [39, 35, 100]. In the computer vision example of image classification (and other mundane automation tasks), if the user of the system is 100% confident that the system is correct 100% of the time then there is no need for explanations—a cat is a cat, is a cat yesterday, today, and tomorrow. In the highly dynamic world of item recommendation where there are competing incentives, explanations can be used to surprise and delight users as well as build trust amongst multiple stakeholders. Today, a user might hate horror movies but tomorrow, that same user might be delighted to be recommended a particular horror movie because it is top trending in the country and he wants to be part of that moment, part of the cultural zeitgeist.

Evaluating the explanations of a single model prediction instance is separated into two components 1) faithfulness of the explanation 2) ease of human understanding [77, 94, 25]. An explanatory model is said to be locally faithful if the predictive behavior of the explanatory model in the vicinity of the single instance of interest is consistent with the predictive behavior of the underlying recommender model in the same vicinity. An explanatory model is said to be intelligible or interpretable if the explanation for a single recommendation instance is readily understood by a human. Evaluating the ease of human understanding of a local explanation is highly subjective and task dependent and not the focus of this paper. Studying the faithfulness of an explanation model is important because a low-fidelity explanation, an explanation that does not closely approximate the behavior of the underlying model, means that the explanation model is not accurately or honestly describing the underlying recommender model's decision making process [40, 59].

3.3 Mathematical Desiderata

Two of the most popular model-agnostic local explanation methods are LIME and SHAP. LIME learns a separate interpretable model trained on a new data set of random permutations of the original data instance we are seeking to explain [77]. SHAP explains the prediction of an individual data instance by computing Shapley values [61]. Shapley values is a game theoretic technique that estimates the contribution of each feature to the prediction also by perturbing the original input data instance [86].

In this section, we lay down the mathematical foundation and build up the theoretical scaffolding necessary for understanding our ensuing contributions.

3.3.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) is a framework for training a secondadry interpretable model, or surrogate model, to explain the individual predictions coming from any opaque classifier [77]. The LIME algorithm for training a surrogate model works as following. First, select some data instance $x \in \mathbb{R}^d$ for which you want an explanation, i.e. you want an explanation for why an opaque recommender model f predicted that user feature vector x would play/not play a movie with probability $f(x)$. LIME requires that in order for the explanation to be understandable to humans, the data should be transformed into an interpretable representation such as a binary vector $x' \in \{0,1\}^{d'}$ denoting the presence/absence of interpretable components, e.g. user watched/did not watch movie A in the past. Next, generate a new data set Z of perturbed samples $z' \in \{0,1\}^{d'}$ by drawing nonzero elements of x' uniformly at random. Now that we have a new set of data instances Z in the neighborhood of x', we need labels for them. To obtain the labels needed for our new explanatory model, we transform the perturbed samples z' back into their original representation $z \in \mathbb{R}^d$ and interrogate the opaque model for each instance

$f(z)$. Because we randomly generated the perturbed samples z' we would like to capture the fact that some samples z might be closer or farther to the original data instance of interest x and thus should be weighted accordingly. This weighting scheme is captured by the proximity measure $\pi_x(z)$, which measures the proximity between an instance x to z.

Finally, using this new weighted data set Z and ground truth labels generated by obtaining $f(Z)$ we train a new model $g \in G$ where G is a class of interpretable models such as decision trees, linear models, etc. This new model g is our interpretable, explanatory surrogate model $\xi(x)$ for explaining $f(x)$:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{3.1}$$

L is any loss function of your choice which measures how unfaithful g is at approximating the behavior of f in the local neighborhood of x. We want to minimize this loss function so that the behavior of g mimics the behavior of f as closely as possible in the locality defined by π_x. $\Omega(g)$ is a complexity term of the model—we want this to be low, e.g. we prefer fewer features in the case of linear models. In the original LIME paper, the authors use the square loss function L with ℓ_2 penalty. Typically, $g(z')$ is chosen to be a linear function i.e. $g(z') = \Phi^T z' + \phi_0$ which makes the above a weighted linear regression problem to solve for Φ and intercept ϕ_0.

$$L(f, \Phi, \phi_0, \pi_x) = \sum_{z, z' \in Z} \pi_x(z)(f(z) - (\phi_0 + \Phi^T z'))^2 \tag{3.2}$$

Some of the advantages of LIME include 1) off-the-shelf easy to use implementation available 2) relatively fast computationally 3) works with tabular data, text, and images 4) opaque model can change without needing to change the explanation model implementation [67]. Some of the previously reported disadvantages of LIME include 1) many hyperparameters to set whose choices heavily influence the resulting explanation and leads to many scientific degrees of freedom (perturbation sampling strategy, neighborhood definition, selection of g) 2) instability of explanation output as mentioned in Section 3.6 3) no theoretical guarantees [68, 67]. To the best of our knowledge, we are the first to shine a light on LIME's decreased performance in high sparsity data regions as well as highlight its comparatively good performance in dense data regions as compared to SHAP in a recommender setting.

3.3.2 SHAP

Like LIME, SHapley Additive exPlanations is an attribution method, that is, a method that describes the prediction of a single data instance as the sum of the effects each feature had on the prediction [67]. Shapley values is an explanation framework that explains the prediction of an individual data instance by computing Shapley values [61, 86, 67]. We choose the model-agnostic Kernel Shap formulation (denoted as SHAP in the rest of the paper) which describes the local explanation as a weighted linear regression similar to LIME as shown in equation [1] with $g(z') = \Phi^T z' + \phi_0$. The regression loss function and the weights are given by:

$$\begin{aligned} L(f, \Phi, \phi_0, \pi_x) &= \sum_{z, z' \in Z} \pi_x(z)(f(z) - (\phi_0 + \Phi^T z'))^2 \\ \pi'_x(z') &= \frac{d' - 1}{(d' \ choose \ |z'|)|z'|(d' - |z'|)} \end{aligned} \tag{3.3}$$

where d' is the dimensionality of x' and $|z'|$ is the number of non-zero elements in z'. In contrast to LIME, generation of the data set Z is very different in SHAP. In SHAP, Z is defined as the power set of all non-zero indices in x'. Hence, Z has a size of $2^{d'}$ if we exhaustively enumerate all possible subsets. (Typical software implementations do allow putting an upper limit on the number of samples in Z). Therefore, one of the computational complexities of SHAP is generating this data set Z. Another (minor) difference from LIME is that the regularization parameter in Shapley values regression $\Omega(g) = 0$.

SHAP's Completeness Constraint Property

As shown in [61], this choice of weighting function $\pi'_x(z') = \infty$ when $|z'| \in \{0, d'\}$. This means for $z' = 0 \implies z = \emptyset$ (a null or baseline feature vector) and $\phi_0 = f(\emptyset)$ and $z' = x' \implies \phi_0 + \sum_{i=1}^{d'} \phi_i = f(x)$ (since all zeroes can be dropped from x' as missing/zero features have no contribution, therefore x' is simply a vector of all ones and $\Phi^T x' = \sum_{i=1}^{d'} \phi_i$). This is the so-called completeness constraint. SHAP calls this the local accuracy property [61], Shapley values calls it the efficiency property [86] and yet other papers call it the completeness axiom [91], the conservation property [13], or the summation-to-delta property [87]. For the duration of this paper we will refer to this property as the completeness constraint.

The completeness constraint $f(x) = f(\emptyset) + \sum_{i=1}^{d'} \phi_i$ has two immediate computational implications:

- The intercept of the regression function is set to $f(\emptyset)$ and is no longer a free parameter and thus, does not need to be estimated.
- Φ has $d' - 1$ degrees of freedom. For example, the d'-th element of Φ can be written as $\phi_{d'} = f(x) - f(b) - \sum_{i=1}^{d'-1} \phi_i$.

We would like to comment that $z' = 0$ does not need to correspond to a literal zero/empty feature vector and can be chosen to be any feature vector b as long as $f(x) \neq f(b)$. We do however use a zero feature vector as the null/baseline feature vector in this paper (more details to come in Section 3.5) and use $\emptyset$ to denote it for the remainder of this paper. Qualitatively, the completeness property has a number of implications:

- Under the completeness constraint, the Φ is said to have a fair attribution of feature importance as it captures the contribution of each feature in the underlying model's prediction at data instance x. LIME is simply a best-fit line and the learned linear function may not be equal to f at the data instance x.

- If the data instance x and the baseline b is different in only one feature, then the differing feature is given a non-zero attribution under the completeness constraint (since $f(x) \neq f(b) \forall x \neq b$). To see how we might end up with zero attribution for features without this constraint we reference the example given in [91]. Consider a function $f(x) = 1 - \text{ReLU}(1 - x))$ and say we want a local explanation at $x = 2$. This function changes from 0 to 1 at $x = 1$ and after that it becomes flat. A local explainability method like LIME may result in a regression line with 0 slope due to the local flatness of the function. But choosing $b = 0$ where $f(0) = 0$ would force Shapley values to learn a regression model with a non-zero slope. Therefore, for highly non-linear recommendation models that may have many such flat regions, the completeness constraint helps generate accurate explanations in such "zero-gradient" sub-regions in the feature landscape.

Some of the advantages of SHAP include 1) the prediction of a single instance is fairly distributed among the feature values 2) game theoretic guarantees afforded to it by Shapley values [67]. Some of the previously reported disadvantages of SHAP include 1) slow computation due to high computational complexity 2) like LIME, SHAP is also vulnerable to adversarial attacks and has issues with explanation instability [67]. To the best of our knowledge, we are the first to show SHAP's decreased performance in dense regions of the data set its superior performance in sparse regions of the data set as compared to LIME in a recommender setting. Furthermore, we are the first to trace this difference in performance to SHAP's lower variance in high sparsity data settings, which we show is a result from SHAP satisfying the completeness constraint.

3.4 CLIME: Completeness-Constrained LIME

3.4.1 Experimental Results Motivating CLIME

In this section, we briefly summarize our initial experimental findings on a movie recommender explanation task that served as the catalyst for the resultant body of research. Full implementation details along with a detailed description of the evaluation metric we used can be found in Section 3.5.

Knowing how important explanations can be to the product experience of recommendation systems and knowing that these systems suffer greatly from having either no previous interaction data (the cold-start problem) or very little historical interaction data (in comparison to the available item set), we wanted to evaluate how well SHAP and LIME perform in varying data sparsity settings. Sparsity is defined by the amount of historical interaction data available to explain a recommendation for a particular data instance. In our first experiment, shown in Figure 3.2, we iteratively removed the $top-k$ most important features from the data instance of interest x. We observed that as we increased the number of features that we removed from x, the gap in performance between SHAP and LIME widened, with SHAP outperforming LIME. In a second experiment, shown in Figure 3.3, we divided our movie recommendation data set into eight equal sized groups based on sparsity, i.e. based on the amount of interaction data each data instance had. We observed that SHAP significantly outperformed LIME in the sparsest groups and that LIME outperformed SHAP in the densest groups. Dense segments of the data set include data instances with plentiful historical viewing data while the sparse regions include data instances with very little historical viewing data. This interesting reversal of performance based on the sparsity of the data has been observed previously in machine learning research [14] and has been found to be closely related to the bias-variance characteristics of models.

Both SHAP and LIME attempt to predict the behavior of an underlying model in the neighborhood of the given data instance x. Their ability to provide the correct explanations is therefore tied to their generalization ability in the local neighborhood around x. We can decompose the generalization capability in terms of their bias and variance. To be precise, since both SHAP and LIME are regression models, their generalization error can be measured in terms of the following mean-squared error.

$$\begin{aligned} MSE(x;\Phi) &= \mathbb{E}[(f(x) - \hat{\Phi}(x))^2] \\ &= (f(x) - \mathbb{E}[\hat{\Phi}(x)])^2 + \mathbb{E}[(\hat{\Phi}(x) - \mathbb{E}(\hat{\Phi}(x))^2] \\ &= Bias^2 + Variance \end{aligned} \tag{3.4}$$

where $\hat{\Phi}$ is an estimator of Φ. (Note: if the underlying model is non-stochastic, there is no residual error term).

Bootstrapping is one straightforward way to compute the bias and variance of any model. For the explanation models, the bootstrapping procedure proceeds by generating P local perturbations of x by randomly zeroing out features. For the p-th perturbed vector, we solve the explanation model to get $\hat{\Phi}_p$. So the empirical average $\mathbb{E}[\hat{\Phi}(x)] \approx \frac{\sum_{p=1}^{P} \hat{\Phi}_p(x)}{P}$ can be plugged-in to estimate the bias and variance in the above equation. Note that this bias and variance is meant to capture the behavior of the explanation model in the neighborhood of x.

With these analysis tools, we conducted a bias-variance analysis of SHAP and LIME (results shown in Figure 3.5) on the same eight sparsity groups from the previous evaluation experiment. We observed that in all segments of the dataset, SHAP exhibited higher bias and lower variance. In the sparsest segments, there was a big variance reduction with a small increase in bias resulting in a favorable bias-variance trade-off. This favorable bias-variance trade-off leads to SHAP improving upon LIME significantly in the sparsest regions of the dataset. In the denser regions, there is a small variance reduction with a large increase in the bias resulting in SHAP's poor performance compared to LIME. This analysis provides strong evidence that the behavior of SHAP and LIME with respect to data sparsity can be easily explained in terms of their bias-variance characteristics. We hypothesize that this bias-variance difference arises due to the completeness constraint (present in SHAP and missing in LIME) which we discuss in the next section.

3.4.2 The Bias-Variance and Completeness Constraint Connection

Our findings showed that SHAP and LIME perform differently depending on the density or sparsity of the data instance whose prediction we seek an explanation for. We showed that this difference is statistically significant. After conducting a bias-variance analysis of SHAP and LIME, we observed that SHAP exhibits lower variance than LIME in high sparsity data regions. As we stated in Section 3.2, high sparsity data regions are common in recommendation systems and thus, it is important that these explanation frameworks perform well in high sparsity settings. We posit that the completeness constraint property, inherent in SHAP and missing in LIME, is an important reason for why SHAP outperforms LIME in sparse data settings. In this section, we reason how the completeness constraint is tied to the observable bias-variance characteristics of SHAP, thus foreshadowing the motivation behind our novel completeness-constrained explanation model.

Given that SHAP enjoys the same game theoretic grounding as Shapley values, including the completeness constraint, we asked ourselves the following research question, "How is the completeness constraint connected to the bias-variance behavior exhibited by SHAP in sparse data regions?" The completeness constraint was originally motivated by the desire for attribution methods to fairly distribute the prediction among the features and served as a solution to the gradient saturation problem mentioned in Section 3.3.2. However, given our interest in explanations for recommender systems, we take an entirely different approach to analyzing its role in the performance of SHAP vs. LIME in sparse data settings.

Since the completeness constraint limits the flexibility of the explanation model, by eliminating both the intercept and one degree of freedom from Φ, we argue that it plays the same role as a regularizer. In other words, the limited flexibility prevents the explanation model's regression function from fully fitting the behavior of the underlying model in the neighborhood of the data instance x, thus resulting in increased bias. But this reduced flexibility would also reduce variance of the explanation model. As long as this bias-variance trade-off is favorable (for example in sparse settings), we expect to see improved accuracy in predicting the behavior of f from explanation models with the completeness constraint. Studying the bias-variance trade-off of the completeness constraint is a novel approach and forms the basis of our work.

3.4.3 Formulation of CLIME

As mentioned in Section 3.3, LIME has highly desirable qualities such as off-the-shelf ease of use that makes it an attractive choice over the computationally slower but theoretically more sound SHAP. We propose introducing the completeness constraint into LIME to take advantage of the favorable bias-variance characteristics of SHAP. Additionally, adding this constraint into LIME would provide the fair attribution property found in SHAP and help protect against generating erroneous/zero explanations in locally flat sub-regions. We now introduce our straightforward formulation of Completeness-Constrained LIME (CLIME).

We set up CLIME identically to LIME. We have the data instance $x \in \mathbb{R}^d$ and its interpretable binary representation $x' \in \{0,1\}^{d'}$, a new data set Z comprised of perturbed data samples z' (z in the original feature space) and their corresponding labels $f(z)$, and the proximity weighting function $\pi_x(z)$, all identical to LIME. In order to introduce the completeness constraint into LIME, we borrow the concept of a baseline feature vector $b \in R^d$ from SHAP. Like SHAP, the choice of b is problem dependent. We explain our choice of b for the recommendation model we use in Section 3.5.

CLIME is the solution to the following constraint least squares problem,

$$\begin{aligned} &\min_{\Phi} \sum_{z,z' \in Z} \pi_x(z)(f(z) - (f(x) + \Phi^T(z' - x')))^2 \\ &\text{s.t. } \Phi^T x' = f(x) - f(b) \end{aligned} \tag{3.5}$$

Note that the intercept of the above regression function is $f(b)$ like SHAP. The solution $\Phi \in R^{d'}$ is a vector of coefficients and is interpreted in the same way as the solution for LIME and SHAP. Fortunately, we do not have to solve the above constraint optimization directly since that would make CLIME computationally slower than LIME. The completeness constraint is a linear constraint, and we can eliminate the constraint by the following substitution. First, note that $\Phi^T x' = \sum_{j=1}^{d'} \phi_j$. Therefore, we can substitute out $\phi_{d'} = f(x_0) - f(b) - \sum_{j=1}^{d'-1} \phi_j$ in the above equation. Let $c = f(b) + x'_{d'}(f(x) - f(b))$ and $r(z') = (z'_{1:d'-1} - z'_{d'})$, then the first $d' - 1$ components of Φ (denoted below as $\Phi_{1:d'-1}$) are obtained by the following unconstrained least squares minimization

$$\min_{\Phi_{1:d'-1}} \sum_{z,z' \in Z} \pi_x(z)(f(z) - (c + r(z')^T \Phi_{1:d'-1}))^2 \tag{3.6}$$

The last component of Φ (denoted as $\Phi_{d'}$ above) is obtained by back substituting in the linear constraint. This way of solving for Φ results in an algorithm that should be as fast as LIME as the problem dimension is reduced to having one less degree of freedom compared to LIME and there is no intercept to estimate.

3.5 Experiments

3.5.1 Experimental Setup

Model

We use a Multinomial Variational Autoencoder (Mult-VAE) [57] trained on the MovieLens 20M data set [38] as the recommendation model whose predictions we want to explain. MovieLens is a data set of users that interacted with movies on the MovieLens website. For the Mult-VAE model, each user is a represented as a bag-of-words of movies that they interacted with. Therefore, the feature vector x_u for a user u can be represented as k-hot binary vector of size 20,108 (total number of movies in the data set) with $1's$ for the interacted movies and $0's$ for the rest. For any user represented as this k-hot encoded vector, Mult-VAE model can score the entire collection of 20,108 movies. Typically, these scores are then used to rank the entire collection of movies (in descending order) to generate personalized recommendations/rankings.

Data Preparation

Adapting LIME and SHAP for movie recommendation system explanations was a non-trivial task. For our local explanability experiments, we use the validation split of 10,000 users outside of the training set. For each validation user u, we generated the personalized ranking from the Mult-VAE model and use the top-ranked movie t_u for local explanability. Therefore the data instance x_u is the k-hot vector and $f_{t_u}(x)$ is the score of the Mult-VAE model for the top-ranked movie. Note that the corresponding interpretable version of x is a vector x' of size d' of all ones where d' is the number of non-zero entries in x. From this vector x, the data set Z can be generated by sampling the non-zero indices and therefore are binary vectors of size d'. This data set generation strategy is same for LIME and CLIME whereas it is different for SHAP, as described in Section 3.3. We do control for the number of samples in Z

and keep it fixed to 5,000 for the three explanation methods. Our evaluation metric (described next) requires a ranking of non-zero movies in x, therefore we turn off the ℓ_1 penalty in SHAP and any feature selection heuristic in LIME so that we may get explanation coefficients Φ for all non-zero movies in the data instance x_u. We keep the rest of the parameters fixed to their default values. For both SHAP and CLIME, the choice of baseline is a zero feature vector meaning a null user without any interaction history. The Mult-VAE model outputs an unpersonalized score for each movie when this zero feature vector is used as input. The unpersonalized score is proportional to the number of non-zero interactions for each movie in the training data(typically called the training data popularities of movies in the recommendation models literature).

Evaluation Metric

We quantitatively evaluate the explanation methods using the delta-prediction metric (also seen in other papers as the "change in log-odds" [88, 87, 61, 85]) and adapt it to the recommendation task and call it the delta-rank metric. Given a ranking of non-zero movies in x_u according to the explanation model coefficient $\phi_i, i = 1, ..., d'$, for each validation user u, take the *top-k input movies* according to the explanation model coefficients and remove them from x_u. This gives a modified data instance x_{um} which is the same as x_u except for the missing movies that we removed. Compute the output ranking from the Mult-VAE model with x_{um} as the input. Calculate the difference in the rank of the movie t_u, which was the top ranked movie earlier. The idea is that if the movies that were removed from x_u were really important for the Mult-VAE to rank t_u at the top, we should expect to see a big drop in the ranking of t_u. We remove a large number of movies (for example up-to 30) by taking a few of them at a time (for example 6 at a time) and plot the change in the rank (or delta-rank) as we remove each batch of 6 movies. We expect the delta-rank to be

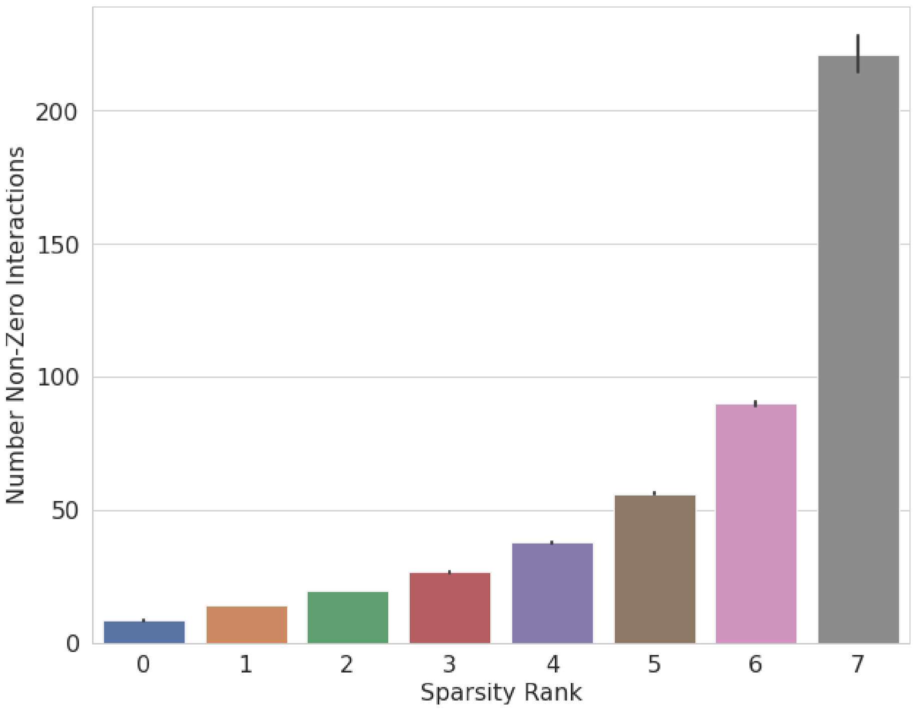


Figure 3.1: Number of non-zero movie interactions in each sparsity segment

negative if important features are removed, and the magnitude of the drop to be proportional to the importance of features removed (therefore lower the better). We compute summary statistics of this delta-rank metric for all validation users.

Since we are interested in comparing the bias-variance and delta-rank performance of SHAP, LIME and CLIME for different sparsity settings, we partition the 10,000 validation users in eight equal sized buckets according to the number of non-zero movie interactions in feature vector x_u. In the results below, we label the data set segment with the highest sparsity as Sparsity Rank $= 0$ and the lowest sparsity segment as Sparsity Rank $= 7$. Figure-3.1 describes the sparsity characteristics of each segment.

Our results can be fully reproduced using the the Jupyter Notebooks found in the supplementary materials.

3.5.2 Results

Delta-rank Comparison Among LIME, SHAP, CLIME

As shown in Figure-3.2, both CLIME and SHAP outperform LIME significantly whereas the difference between CLIME and SHAP is insignificant up to top-20 features. This validates our hypothesis that introducing the completeness constraint into LIME does indeed result in improved local explanability. We also compare the three methods according to sparsity using the eight segments described above (Figure-3.3). We see the expected outcome—the overall delta-rank improvements come from the sparse segments of the data set where CLIME and SHAP outperform LIME. We attribute this improvement to an overall favorable bias-variance trade-off especially in the sparse segments of the MovieLens data set.

Computational Analysis

As mentioned earlier, integrating the completeness constraint into LIME results in an estimation problem of lower complexity and can be solved as fast as LIME. The second figure in Figure-3.2 shows this result.

Bias-Variance Analysis of LIME, SHAP, CLIME

We use a validation set of size 1,000 for bias-variance computation (down from 10,000 to keep the computation time in check) and we solve LIME, SHAP and CLIME estimation problems for 50 bootstrapped perturbation of each validation example. Figure-3.4 shows that indeed CLIME and SHAP exhibit higher bias and lower variance as we hypothesized in the earlier section. Moreover, Figure-3.5 shows that the variance reduction (compared to LIME) is directly proportional to the sparsity whereas increase in bias (compared to LIME) is inversely proportional to the sparsity. These results show that we get the best bias-variance trade-off in the most sparse

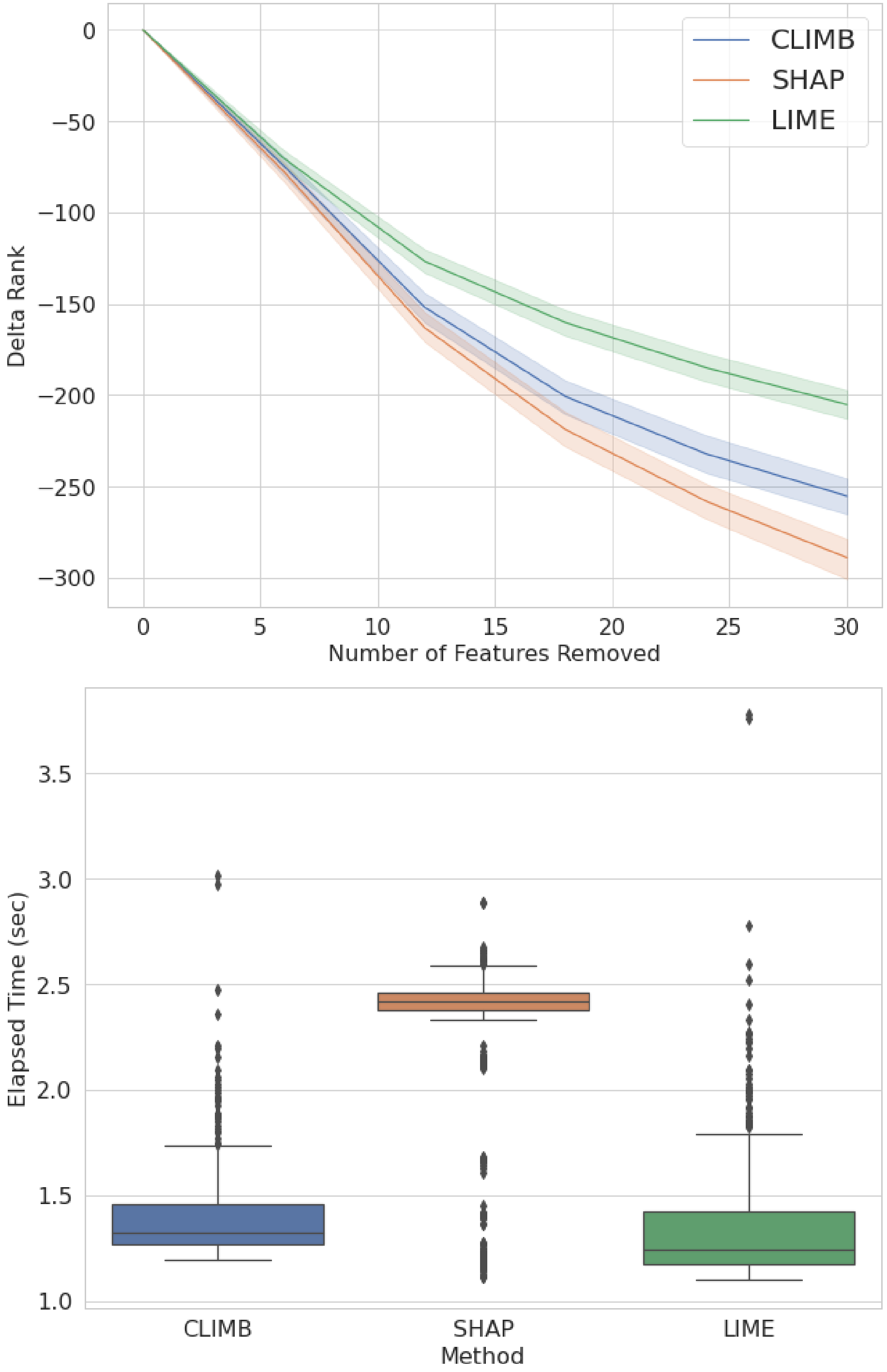


Figure 3.2: Comparing CLIME, SHAP and LIME according to delta-rank and computational speed

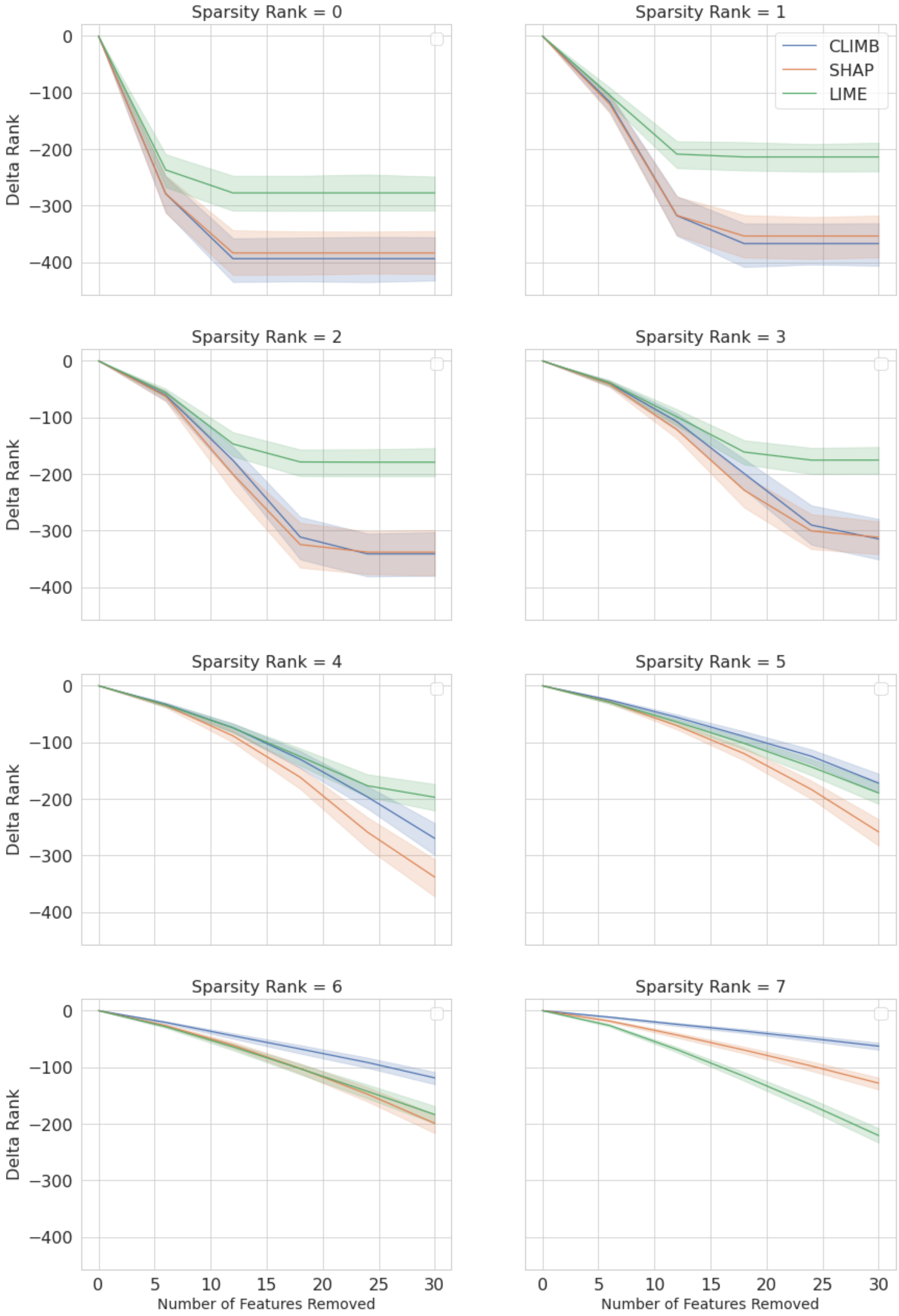


Figure 3.3: Comparing CLIME, SHAP and LIME in decreasing order of sparsity. Sparsity Rank = 0 is the data set segment with highest sparsity and Sparsity Rank = 7 has the least sparsity

segments of the data set. Our results also show the role the completeness constraint plays as a regularization technique, therefore significantly improving the performance of LIME by incorporating completeness constraint in it in the sparse segments of the MovieLens dataset.

Qualitatively Examining Local Explanations

We find examples where the delta-rank metric for CLIME is far better than LIME to build an intuition for how improvements in delta-rank affect the outward quality of the resulting explanations. "Star Wars : Empire Strikes Back" and "Harry Potter and The Goblet of Fire" are two such examples selected from the sparse region of the MovieLens data set. Looking at the explanations visually, the results for both CLIME and SHAP are identical and qualitatively much better than LIME (we highlight the explanations in red that subjectively seem to make little sense). Looking at these explanations and noting the improvements in the delta-rank metric, we conclude that these explanations not only visually make sense but are in-agreement with the underlying model. We note that the metric or a visual examination alone will not allow us to make this claim. We also include one example from the dense region of the data set, "Star Trek: The Wrath of Khan", where the delta-rank metric for LIME is superior to CLIME and SHAP. CLIME seems to include a number of seemingly unrelated movies in its explanations. According to our analysis, the bias-variance trade-off due to the completeness constraint is unfavorable in the dense regions and this is reflected in the subjective quality of the explanations as well.

3.6 Discussion and Related Work

As we highlighted in Section 3.2, the recommender setting requires domain specific consideration given the unique technical challenges it poses and the unique and var-

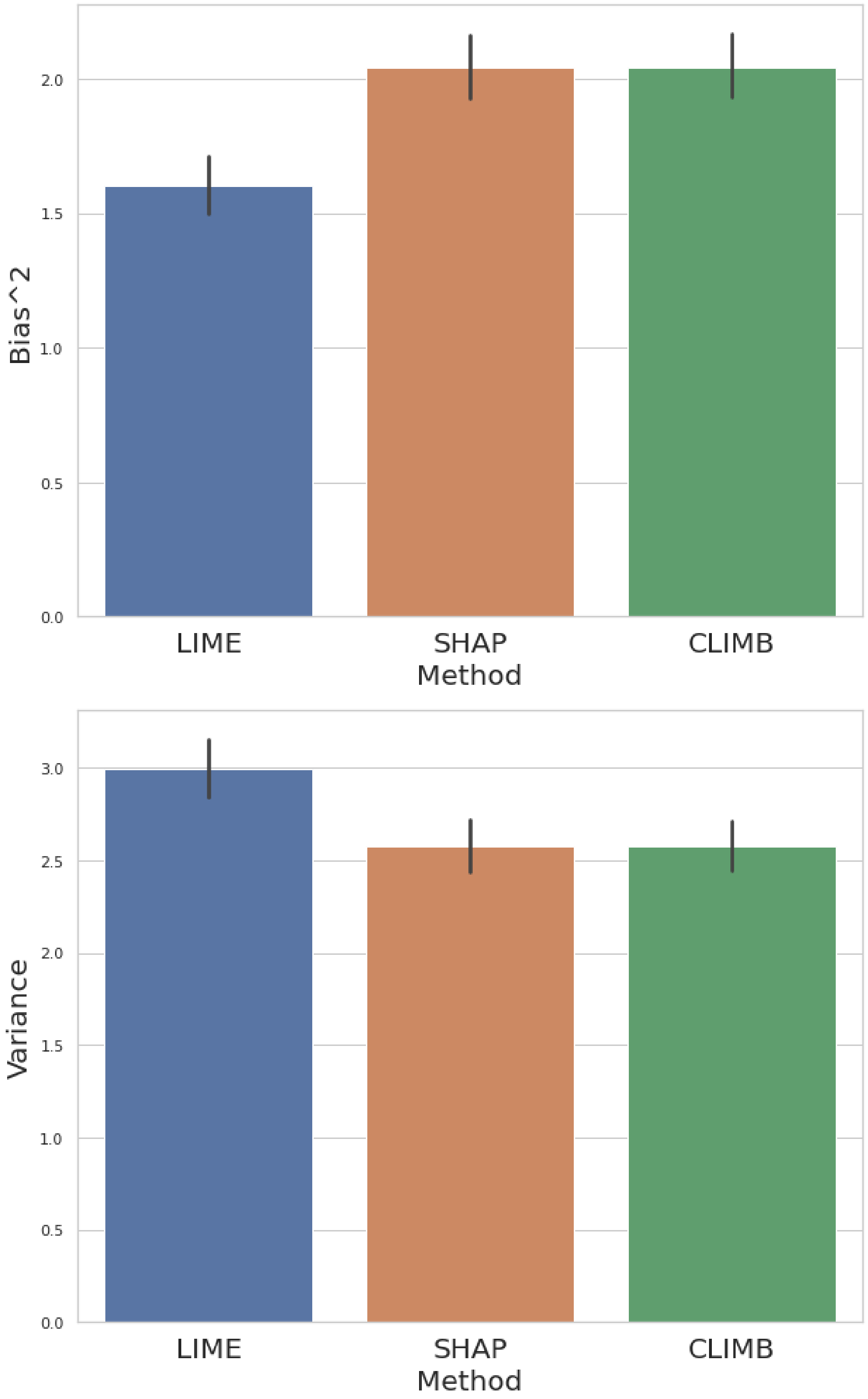


Figure 3.4: Comparing the overall Bias and Variance of CLIME, SHAP and LIME

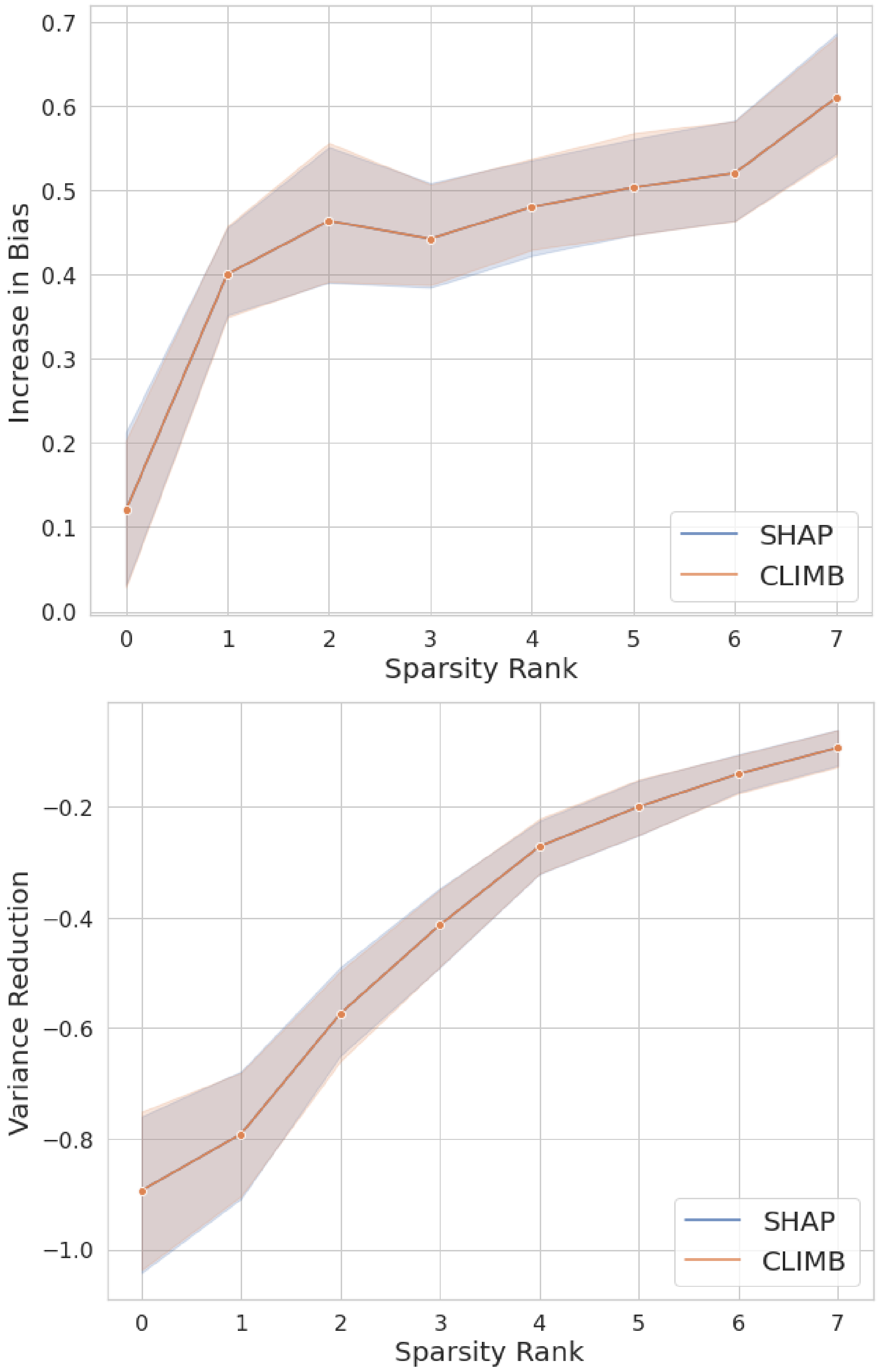
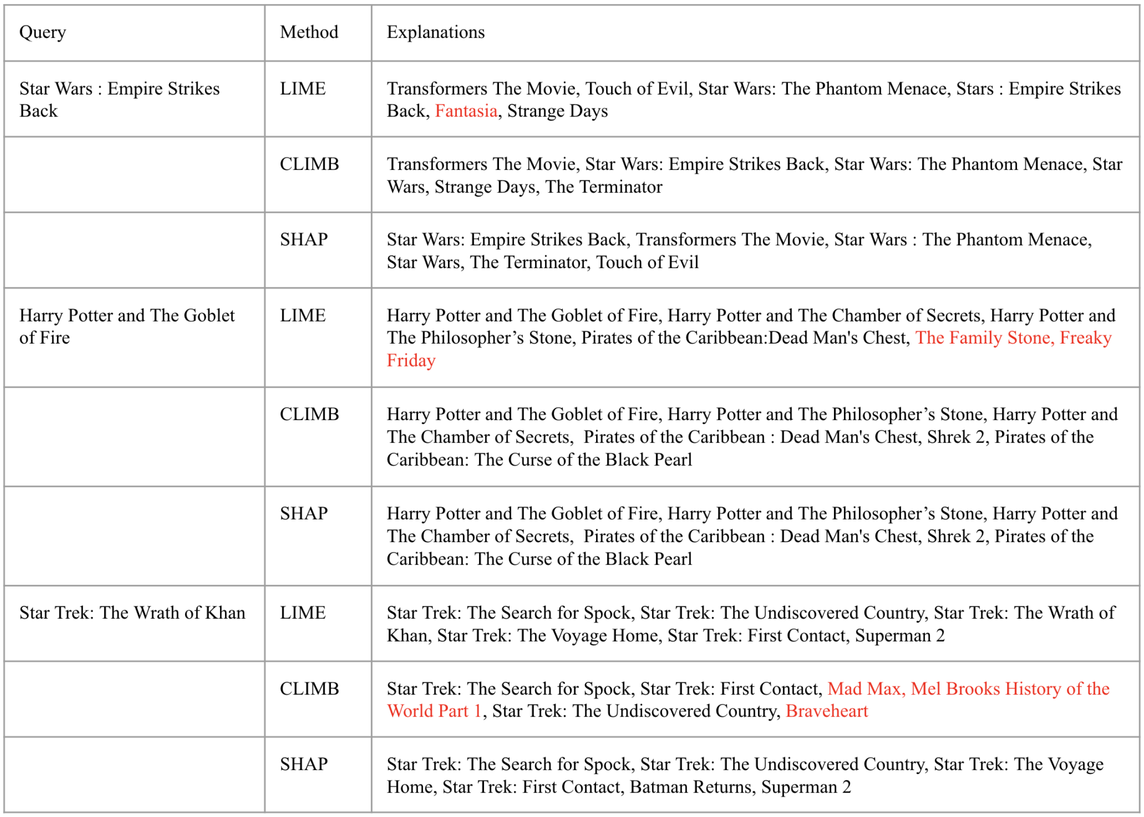


Figure 3.5: Comparing the Change in Bias and Variance of CLIME and SHAP relative to LIME with decreasing levels of sparsity



Query	Method	Explanations
Star Wars : Empire Strikes Back	LIME	Transformers The Movie, Touch of Evil, Star Wars: The Phantom Menace, Stars : Empire Strikes Back, Fantasia, Strange Days
	CLIMB	Transformers The Movie, Star Wars: Empire Strikes Back, Star Wars: The Phantom Menace, Star Wars, Strange Days, The Terminator
	SHAP	Star Wars: Empire Strikes Back, Transformers The Movie, Star Wars : The Phantom Menace, Star Wars, The Terminator, Touch of Evil
Harry Potter and The Goblet of Fire	LIME	Harry Potter and The Goblet of Fire, Harry Potter and The Chamber of Secrets, Harry Potter and The Philosopher's Stone, Pirates of the Caribbean:Dead Man's Chest, The Family Stone, Freaky Friday
	CLIMB	Harry Potter and The Goblet of Fire, Harry Potter and The Philosopher's Stone, Harry Potter and The Chamber of Secrets, Pirates of the Caribbean : Dead Man's Chest, Shrek 2, Pirates of the Caribbean: The Curse of the Black Pearl
	SHAP	Harry Potter and The Goblet of Fire, Harry Potter and The Philosopher's Stone, Harry Potter and The Chamber of Secrets, Pirates of the Caribbean : Dead Man's Chest, Shrek 2, Pirates of the Caribbean: The Curse of the Black Pearl
Star Trek: The Wrath of Khan	LIME	Star Trek: The Search for Spock, Star Trek: The Undiscovered Country, Star Trek: The Wrath of Khan, Star Trek: The Voyage Home, Star Trek: First Contact, Superman 2
	CLIMB	Star Trek: The Search for Spock, Star Trek: First Contact, Mad Max, Mel Brooks History of the World Part 1, Star Trek: The Undiscovered Country, Braveheart
	SHAP	Star Trek: The Search for Spock, Star Trek: The Undiscovered Country, Star Trek: The Voyage Home, Star Trek: First Contact, Batman Returns, Superman 2

Figure 3.6: Comparing explanations generated for two sparse and one dense queries

ious needs it has for explanations. To the best of our knowledge, we are the first to evaluate SHAP and LIME based on their performance in different data sparsity settings. More concretely, to the best of our knowledge, we are the first to evaluate these explanation models based on how they perform when explaining a recommendation for a data instance with very little historical interaction data versus when explaining a recommendation for a data instance with plentiful historical interaction data. We are also the first to connect this difference in data-sparsity-dependent performance to the differing bias-variance characteristics of SHAP and LIME and subsequently, the completeness constraint that is inherent in SHAP but missing in LIME. We then go on to prove this hypothesis by formulating a novel explanation method called Completeness-Constrained LIME (CLIME) that indeed improves the performance of LIME in sparse data settings.

Previous work comparing SHAP and LIME focuses on evaluating these explanation methods based on their stability or reproducibility, that is, their ability to return consistent explanations over numerous runs on the same input [115, 34, 104, 62, 102]. Other work evaluating explanation frameworks assesses their local fidelity or faithfulness to the original underlying model [66, 27, 21, 10, 102]. Additionally, a common paradigm when evaluating and comparing SHAP, LIME, and other explanation methods is to introduce a new evaluation metric and evaluate the explanations against this metric, e.g. effectiveness, efficiency, necessity, sufficiency, XAI Test, feature importance similarity, feature importance consistency, impact score, impact coverage [74, 69, 43, 49, 58, 31, 29]. Most recently, researchers evaluated the robustness of LIME and SHAP and found them to be vulnerable to adversarial attacks where the explanatory models can be manipulated to hide potentially harmful biases in the original model [89, 106].

3.7 Summary

In this chapter, we (*i*) provided motivation for why explanations for recommender systems require special consideration, (*ii*) showed the shortcomings LIME, a popular, easy to use explanation method, had in addressing the needs of recommender systems, which often operate in high sparsity data settings, (*iii*) traced the root of the issue to an important property that is found in another popular but slower explanation method, SHAP, (*iv*) incorporated this property into LIME to create a novel explanation framework called CLIME, and finally, (*v*) showed that CLIME is superior to LIME in high sparsity data settings, is as fast as LIME (much faster than SHAP), and is as easy to use as LIME.

Chapter 4

Part 2: Computers Providing Humans with Algorithmic Explanations[1]

This chapter is based on "COFFEE: Completeness-Constrained Faithful Explanations." It shows another example of the second mode of human-machine collaboration, computers providing humans with algorithmic explanations.

When generating local explanations of an opaque machine learning model by a variety of methods, we run into the problem of evaluating the explanations and determining the best one. We argue that evaluating an explanation of a model prediction has two components, faithfulness of the explanation to the opaque model and ease of human understanding of the explanation. In this work, we aim to develop quantitative ways to evaluate the faithfulness of the local explanations when explaining state-of-the-art movie recommendation models. We propose the quantitative evaluation of faithfulness in terms of an approximation error between the local explanation and the opaque model. We show that this approximation error can be minimized to

[1] This chapter was originally presented with the following citation: Claudia V. Roberts, Ehtsham Elahi, and Ashok Chandrashekar. "COFFEE: Completeness-Constrained Faithful Explanations."

obtain a new local explanation technique. The proposed approximation error is an intuitive way to reason about the behavior of local explanation methods compared to the axiomatic approach adopted in the local explainability research. Therefore we use the proposed approximation error to compare widely used local explanation methods in terms of their faithfulness/fidelity to the opaque model. Finally for the ease of human understanding component, we describe different ways to present results of an explanation model in terms of simplified feature inputs by optimizing the same approximation error in the transformed feature space.

4.1 Overview

With an ever increasing role of machine learning (ML) models in decisions that directly affect our lives, the importance of understanding the inner workings of these ML models is only increasing. Explainable ML or model explainability is an important area of research that deals with uncovering the inner working of complex ML models. The classical examples used to motivate this line of research are ML models used in high-stakes applications such as criminal justice and credit lending [93]. However, even seemingly less critical applications like online recommendation systems (e-commerce, travel, music and movies etc.) have become an important part of our lives as these recommendation systems aim to help users in navigating very large catalogs and selecting the right items for consumption. The stakeholders of explainable ML research in recommender system applications include end-users (e.g. customers of online services with recommendation systems) and product researchers and engineers. For example, explaining product recommendations to users of an e-commerce service may help them build trust in the recommendations and can provide them answers to questions like, "why did you (the recommender system) recommend this product to me?" For the machine learning researcher or engineer developing the model, explanations provide means to debug and detect any issues and build trust on the robustness before deploying the model in the production system. In short, in a world that is increasingly reliant on ML driven automated services, understanding ML driven recommendations that help users make choices is very important [70].

There is generally a trade-off between the inherent interpretability of a model and its accuracy. A very limited class of simpler machine learning models are easy to interpret like linear models or decision trees. More complex machine learning models like deep neural networks and gradient boosted decision trees are much more accurate but are harder to interpret. In real world applications, these complex models are widely used because of their accuracy. This motivates the development of post-hoc model explanation techniques that are meant to provide insights into these complex models without having to sacrifice accuracy. There are two types of post-hoc explanation techniques. 1) Global explanation techniques which are concerned with understanding the overall behavior of the model and 2) Local explanation techniques which are meant to understand a model's decision for a given instance of data (referred to as the input query in this paper). The thrust of this work is on local explanations, which in the context of explaining the decisions made by movie recommenders, would amount to answering questions that an end-user might have (such as, "why did you recommend me that?") or perhaps answering questions that a research engineer may have about model's behavior for a particular input query, whose motivation may be to debug a model or make the model more robust.

Given the importance of post-hoc explainability, there are a large number of local explanability techniques available [77], [61], [91], [87]. Given the variety of techniques that are able to generate local explanations, the important question is how to evaluate the generated explanations and choose the best one. This is a big challenge in model explainability research. Traditionally explanations are evaluated using surveys sent out to human evaluators/editors. Users are asked for their preference for explanations coming from different methods. This is a very difficult way to evaluate because of time and cost needed to conduct these surveys. For a researcher developing novel model explanation techniques, the slow feedback cycle of a survey may reduce speed of innovation significantly. Moreover, it has been shown that interpreting survey results in explanability research is very challenging [50] because of inherent noise in survey responses. In this paper we propose to divide the evaluation of explanation into two components and aim to provide quantitative evaluation for one of the components.

We would like to argue that there are two components in the evaluation of a local explanation to an opaque model. a) Faithfulness of the explanation to the opaque model and b) ease of human understanding of the explanation. We view the two components as only loosely coupled and propose that we can make progress on each of them independently.

- **Faithfulness of explanation:** Model explanations should be evaluated on the basis of faithfulness to the underlying opaque model that we are trying to interpret. What does it mean for local explanation to be faithful to the model being explained? At a high level, it means that explanation is consistent with the behavior of the opaque model. Many local explanations are provided as scalar feature importance weights (also called feature attribution coefficients in some papers) for each of the dimension of the input query. We can interpret these feature importance weights to define a hyperplane in the feature space of the opaque model (see fig-4.1). Different local explanations correspond to different hyperplanes. For an explanation that is faithful and consistent with the opaque model, we would expect the corresponding hyperplane to form a good linear approximation of the model behavior in the neighborhood of the input query. This is the key insight in our development of quantitative evaluation for faithfulness of local explanations. We argue that it is important to realize that the explanations are meant to uncover an opaque model's behavior therefore we decouple the subjective human understanding aspect from the evaluation of faithfulness of explanations. As an example, if one were to apply a faithful local explanation method to a "bad" movie recommendation model, we may get an explanation that may not make any sense to the humans. For example, if the recommendation model treats horror movies as being similar to comedy movies, the local explanation method may provide comedy movies as top explanations for a horror movie recommendation. This is purely a reflection of the inner workings of the recommendation model and the explanation is faithful to the recommendation model although it is unlikely to be intuitive for a human user. This paper is primarily focused on evaluating the faithfulness/fidelity of explanations.

- **Ease of human understanding of explanation:**. While the focus of the paper is not on this component, we will provide a brief commentary here. As mentioned earlier, the stakeholders of model explanations are ultimately humans. Therefore it is important that they understand the provided explanation. We pose this problem as finding the best way to present model explanations to users and the task is similar to building a good user-interface (UI) design between the explanation and the end-user. A similar question has been explored in [44]. For example, to explain a product recommendation system in terms of the past purchases/interactions of the user, we get feature importance for each item in the interaction history and one simple UI would be to rank the items in terms of their importance. In our opinion, work on this UI design can be done irrespective of which method is generating the explanation as long as it is in the format that the UI anticipates (scalar feature importance weights for example). Other examples of UI design may require to change the representation of data as the explanations generated in the original features may still be too complicated for humans to understand. For example in the case of movie rec-

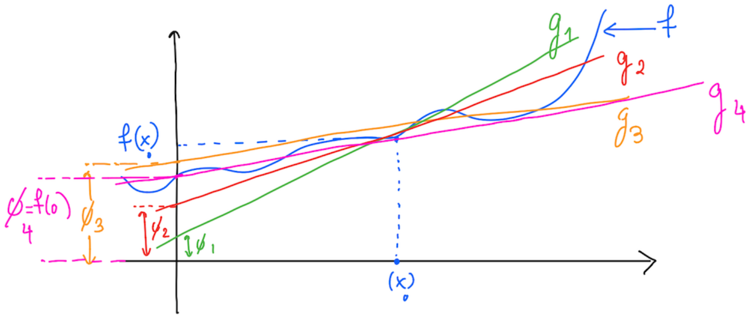


Figure 4.1: Different explanation models shown as lines along with the opaque model f

> ommendations, we may want to use an alternative representation of the movies based on the natural language based tags. With this change in representation for explanations, the UI can show a word-cloud style visualization of the metadata of the movies in the explanation. If the contract between the UI and the explanation is clearly established, we can independently iterate on finding the most faithful explanation while presenting the explanations in a way that the UI design research finds most intuitive for the end-user.

In this paper we are primarily focused on evaluating the faithfulness of local explanations using the geometrical perspective of local explanations as hyperplanes as mentioned above. We are particularly interested in local explanation hyperplanes that obey the completeness axiom [91] (also referred to as local accuracy axiom in [61]) since the completeness axiom ensures that the explanation provides a fair attribution to each feature in addition to protecting against the so-called "sensitivity" problem of local explanations [91]. We propose to evaluate these hyperplanes using an approximation error between the opaque model and the first-order Taylor approximation based on the slope of the hyperplane. We show that this approximation error can be minimized while imposing the completeness constraint. This leads to the development of a new local explanation technique that follows the completeness axiom and provides the most faithful explanations as measured by the approximation error. To summarize our contributions,

- We evaluate the faithfulness of local explanations by computing the approximation error between the opaque model and the hyperplanes representing the local explanations (Section 4.2).
- We show that the proposed evaluation of faithfulness can be optimized under the completeness constraint leading to a new local explanation technique. We name this new technique as COFFEE (a playful acronym for Completeness-Constrained Faithful Explanations) (Section 4.2).
- We showcase our ideas by comparing popular local explanation techniques LIME and SHAP with our proposed technique COFFEE on two state-of-the-art collaborative filtering models, EASE [90] and Multi-VAE [57]. We compare the approximation error for each of the techniques and also visually display the explanations generated (Section 4.3).
- Finally, we discuss ways to address the ease of human understanding aspect by considering a modified form of COFFEE. We show that we can perform a feature transformation to represent explanations in a more intuitive form and still optimize approximation error to get explanations in the simplified feature space (Section 4.4).

4.2 Approximation error for quantifying faithfulness of local explanations

We take the additive linear feature attribution perspective introduced in [77] [61], [91] and define the local explanation methods as models. To start, we assume there is an opaque model $f : \mathbb{R}^N - > \mathbb{R}$ that we want to explain at an input query $x_0 \in \mathbb{R}^N$. The local explanation for the input x_0 is a linear model in the feature space x with feature attribution ϕ_i for the ith-dimension of x.

$$g(x) = \phi_0 + \sum_{i=1}^{N} \phi_i x_i \tag{4.1}$$

Shapley values and many other local explanation schemes think of feature attribution relative to some *baseline counterfactual* feature vector b. These methods require the summation of feature attribution coefficients to equal to the difference in the model prediction at the input query x and the baseline counterfactual b i.e. $\Phi^T x = f(x) - f(b)$. This is the completeness axiom of local explainability. In the hyperplane interpretation, this implies forcing the intercept of the hyperplane to $f(b)$.

Based on this formulation, the unknowns to find are the intercept of the local explanation model ϕ_0 and the feature attribution coefficients $\Phi \in R^N$. Many of the existing local explanation techniques can be represented in this additive linear feature attribution form and they are simply different parameterizations of this linear form. In figure-4.1, we illustrate the opaque function f as well as different lines corresponding to different local explanations.

With the formulation of local explanations as hyperplanes, we evaluate the faithfulness of the explanation as an approximation error between the first order Taylor approximation based on the slope Φ of the hyperplane and the opaque model in a *neighborhood* around the input query x.

Given an opaque model f, a set X consisting of M samples $x_i, i = 1, ..., M$ from the neighborhood of input query x_0 and a local explanation vector Φ, we define the approximation error as the root mean squared error between the value of the opaque model f in the neighborhood X and the first-order Taylor approximation at x_0 based on the local explanation Φ

$$\text{RMSE}(\Phi; X, f) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - (f(x_0) + \Phi^T(x_i - x_0)))^2} \tag{4.2}$$

If we have k local explanations $\Phi_1, ..., \Phi_k$, we compute $\text{RMSE}(\Phi_1; X, f)$, ..., $\text{RMSE}(\Phi_k; X, f)$. The local explanation that leads to the smallest value of the RMSE is deemed the most faithful explanation of the opaque model as it provides the best approximation to the model behavior in the neighborhood X.

How to select the neighborhood set X? The concept of neighborhood is already used in many local explanation techniques (directly in LIME [77] and indirectly in Shapley values [61] and Integrated Gradients [91] by considering paths from the input query to the baseline counterfactual feature vector). We follow a similar approach to LIME. The neighborhood set consists of all points x_i such that $|x_i - x_0| \leq d$ i.e. it contains all points in a ball of radius d around the input query x_0. Given this neighborhood set X, the above RMSE metric captures the accuracy of the first-order Taylor approximation based on the local explanation hyperplane. Figure-4.2 illustrates this point. In the language of local explainability, the neighborhood set X encompasses the counterfactuals that we hope to answer through the explainability framework and are no farther away from x_0 than a distance d. In practice, we can randomly sample M points from the neighborhood and use that for the computation of RMSE as shown above. In the item recommendation systems application, the neighborhood set X we use is a ball of radius 1 in the normalized Levenshtein (edit) distance consisting of samples of points between the query x_0 and zero vector (this distance is normalized so that the max distance is 1). For the item recommendation application, this baseline counterfactual represents an input without any past interaction with any of the items in the recommendation dataset.

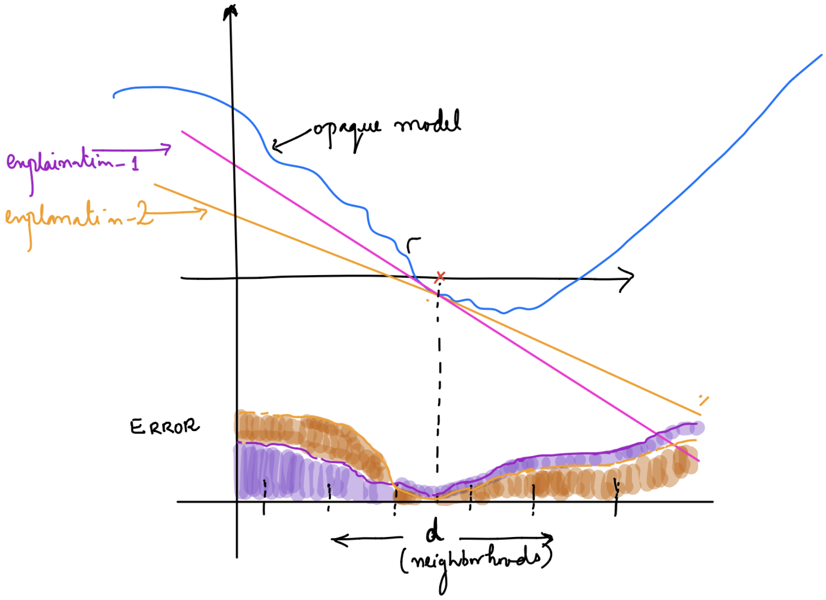


Figure 4.2: Computing RMSE over different neighborhoods for two different explanation models

The mean squared error defined above is a differentiable convex function in Φ and therefore can be minimized. However, we don't want to minimize it unconstrained as the unconstrained solution may not follow the completeness axiom. Fortunately, the completeness axiom is a linear constraint and we can easily enforce it to get a convex optimization problem with a linear constraint.

$$\begin{aligned} &\min_{\Phi} \sum_{i=1}^{m} (f(x_i) - (f(x_0) + \Phi^T(x_i - x_0)))^2 \\ &\text{s.t. } \Phi^T x_0 = f(x_0) - f(b) \end{aligned} \tag{4.3}$$

Solving this optimization programs gives us what we refer to as Completeness-Constrained Faithful Explanation (COFFEE).

4.3 Baseline Local Explanation techniques

4.3.1 Model Gradients as local explanations

If we ignore the completeness constraint and minimize the above MSE in a vanishingly small neighborhood around the input query, the solution Φ of the minimization problem is exactly equal to the model gradient. The proof follows from the intuition that the first-order Taylor approximation based on the gradient provides the best linear approximation to the function in a vanishingly small neighborhood. It makes intuitive sense to use gradients as local explanations because another way to look at local explainability around a query is to perform sensitivity analysis i-e how much the model's prediction changes if the query feature vector is perturbed ($\frac{dy}{dx}$)? Therefore, one of the popular ways to compute feature importance weights have been using the gradient of the model [12] i-e

$$\phi_i = \frac{\partial f(x)}{\partial x_i}, \forall i = 1, ..., N \tag{4.4}$$

The challenge of finding the explanation model is reduced to computing the gradient of the model at x_0. Given the gradient, the intercept is simply $\phi_0 = f(x) - \nabla f(x)^T x$ (by using the first-order Taylor approximation of the function around the input query x). ϕ_0 therefore need not equal to $f(b)$ for some baseline counterfactual b. In other words, gradients do not obey the completeness axiom and that is one of the main downsides of using gradients as local explanations. Despite this limitation, it is still interesting to compare other local explanation techniques to gradients in the context of the approximation error as the RMSE based on the gradient defines a lower bound on the approximation error in a vanishingly small neighborhood around the input query x and we can compare other approaches with that to see how much they deviate from the local gradients.

Many explainability frameworks use a simplified representation of the original feature space [61] which is essentially a discretization of the input query x into a binary vector. Many of the techniques we discuss depend on perturbing the input query x. Using this binarized representation x makes these perturbations particularly simple, we simple turn-off the "on" bits in the binary representation. The quantitative evaluation presented in this paper is generally applicable to continuous valued feature vectors too but we make use of this simplified representation where it offers advantages. This is the same perspective as adopted by [61] and many others in the explainability literature.

Finite difference method would be one of the simplest way to approximate the gradient of the model. By perturbing the i-the dimension of the input x by a scalar h_i, the partial derivative of the model is approximated as

$$\phi_i = \frac{\partial f(x)}{\partial x_i} \approx \frac{f(x+h) - f(x)}{h} \tag{4.5}$$

One of the challenges of using finite difference method for computing gradients is the magnitude of perturbation h added to the i-th dimension. Fortunately, the use of binarized representations of feature vectors makes it as simple as turning-off the i-th feature $(x \setminus \{i\})$, we can write the finite difference as

$$\phi_i = \frac{\partial f(x)}{\partial x_i} \approx f(x) - f(x \setminus i) \tag{4.6}$$

If gradients is the only available information about the model then that amounts to linearizing the model behavior around x (essentially the first-order Taylor approximation at x). ϕ_i therefore are interpreted as the *local explanation coefficient* for the i-th feature in the input query x; a positive value for ϕ_i indicates that the output of the model would increase if the value of the ith-feature is increased (and vice-versa if ϕ_i is found to be negative). There are some limitations to the gradient based perspective to local explainability. Describing those limitations are out-of-scope for this paper (refer to [91]) but we do consider methods that are meant to address those limitations (for e.g Shapley values). However, even with the simplicity of gradient based approach to local explainability, it may be challenging to compute them. For example,

- The model may not be available to us for computing the gradient. We generally don't assume that we have access to the inner details model. We only assume that we can query the model to get the output of the model for any given input.
- Even if we may have access to the model, it's possible that the model is non-differentiable. Tree based models are examples of this.

Therefore we are interested in model-agnostic gradient estimation techniques. Below we introduce three model-agnostic local explanation methods. Two of them are among the most popular methods, LIME [77] and Shapley values [61] and the third one is inspired from numerical gradient estimation literature that we present as a simple baseline for local model explanations. With gradient based method already used in model explainability research, we find it surprising that we have not been able to find this baseline in existing explainability research literature.

4.3.2 LIME

LIME solves for the additive linear attribution model in equation-4.1 by solving a weighted regression on a simulated dataset (X, y) where the input features X are generated by perturbing the input query x_0 and y is the response of the opaque model f. Φ is then the best fit hyper-plane to the simulated dataset.

$$\Phi = (X^T W X)^{-1} X W y \tag{4.7}$$

In this equation, W is the weight matrix that is a hyper-parameter in LIME. The intercept ϕ_0 can also be easily learned as part of the solution of the above linear system or it can be set equal to $f(b)$. However, LIME still wouldn't follow the completeness constraint as it is simply a best fit line and may not be equal to the value of f at x_0 ($\Phi^T x_0 + \phi_0 \approx f(x_0)$).

Later we show how our quantitative evaluation of LIME behaves when we try to play with different types of weight matrices. Using a linear regression to approximate the gradient is a well studied technique [91]. Moreover, [33] also notes that the the explanation model learned by LIME is simply the local gradient of f at the query x.

4.3.3 Shapley Values

Shapley values computation is another popular way for computing local explanations [61]. On first thought, it may not be very clear why it makes sense to compare Shapley values using the same approximation error for their fidelity to the opaque model. The way Shapley value computes the feature attribution vector Φ is by averaging the gradient over all possible paths between x_0 and b (the so-called path gradient between x and b). Since we are working with a binarized representation x, the baseline counterfactual b is typically chosen to be the empty set (same as the zero vector discussed above). What is the notion of paths between x and the empty set? Paths are the subsets and gradients are estimated as finite differences!

To make the connection clear, let's look the Shapley value equation that we have written in a slightly different way. In order to compute the feature attribution for the i-th feature, let's remove the i-th feature from the original feature vector x and consider the power set of all subsets of $x \setminus \{i\}$ of size s that we denote as $F_s \setminus \{i\}$. Then the i-th feature attribution coefficient is

$$\phi_i = \frac{1}{N} \sum_{s=0}^{N-1} \frac{1}{\binom{N-1}{s}} \sum_{S \subseteq F_s \setminus \{i\}} (f_{S \cup i}(S \cup i) - f_s(s)) \tag{4.8}$$

Lets try to understand this equation by looking at it from right side. First of all, the subtraction in the inner summation $f_{S \cup i}(S \cup i) - f_s(s)$ can be recognized as the finite difference when the i-th feature is added to the feature vector represented by the subset S. This finite difference is then computed over all possible subsets S of size s (there are $\binom{N-1}{s}$ of them) and then taken an average over. The final summation is simply considering subsets of all sizes and then computing an average over them.

4.3.4 Discussion

We would like to compare COFFEE's constrained optimization based approach with the axiomatic approach used to motivate Shapley values [61]. Of the three axioms that Shapley values fulfill, Consistency axiom is the key property that endows Shapley values with characteristics of model interpretability and it is quite different from our direct optimization of faithfulness (Other two axioms Completeness and Missingness are fulfilled by both Shapley values and COFFEE). In simple terms, Consistency axiom requires that the local explanation coefficient of a feature i for a model A should be numerically no less than the local explanation coefficient for the same feature in another model B if model A is more sensitive to that feature i-e $f_A(x) - f_A(x \setminus i) > f_B(x) - f_B(x \setminus i)$. This is another way of saying that Shapley values behave as gradients of the opaque model when small changes are introduced to its inputs. Moreover, this property is required to hold for a neighborhood that contains all points between the input query and the baseline counterfactual which implies the use of similar neighborhood that we are considering in our approximation RMSE (the neighborhood set in Shapley values contains combinatorial number of points, our neighborhood set can be considered a down-sampled version). Our proposed approximation RMSE addresses the faithfulness of local explanations directly. As we mentioned before, without the completeness constraint, solution to the approximation RMSE tends to the actual gradient in the limiting case of vanishingly small neighborhood around X.

Therefore, it is much more intuitive to reason what COFFEE explanations are : In the limiting case of small neighborhoods, COFFEE mimics the behavior of a gradient [33] as much as it is permitted by the completeness constraint.

One of the primary criticisms on LIME is that it does not obey the completeness axiom and therefore its explanations don't provide fair attribution across all the dimensions of the input query. We can view COFFEE as improving upon LIME by imposing the completeness constraint on LIME. To see that, recall that the error term in the MSE definition is $f(x)-(f(x_0)+\Phi^T(x-x_0))$ which reduces to $f(x)-(\Phi^T x+f(b))$ by replacing $\Phi^T x_0 = f(x_0) - f(b)$ using the completeness constraint. This is simply the error function of LIME (without the weight matrix) where have set the intercept to $f(b)$. LIME is a highly flexible framework and is very easy to use. Therefore, for someone who prefers to use LIME, COFFEE maintains all of its advantages and only serves as an improvement over it by making LIME explanations to have a fair attribution across all features in the input query.

4.4 Experiments

We compute the approximation RMSE for local explanations of collaborative filtering models trained on the MovieLens 20M dataset [38] which is a rich open-source dataset containing a large number of user, movie interactions. We follow the same procedure as [90], [57] to construct a train/validation/test split of this dataset and train the collaborative filtering models on the train split and compute the approximation RMSE on the validation split. Following the same procedure as in the existing literature [90] and [57], we take an instance of the validation set (a movie interaction history) and further split that in two parts. We treat the first part (a set of movies liked by the user, represented as a k-hot encoded binary vector) as input feature to collaborative filtering model. The local explanations are generated on these input features. We examine the output of the collaborative filtering model on the held-out movies in the 2nd part of the validation instance split. For each validation data instance, we generate a pair of movie interaction history list and a held-out movie to form a separate query to the explanation model. Two such example query movies are shown in figure-4.3. RMSE is computed for each such pair and then we summarize the RMSE metrics computed on the validation dataset.

We implement COFFEE using Scipy's optimization library using Sequential Least Squares Programming (SQLSP). For LIME, we use the software written as part of the original paper [77]. For Shapley values, we use the Kernel SHAP implementation distributed as part of the original paper [61]. For both LIME and Kernel SHAP, we turn off all feature selection and regularization so that we can compare Taylor approximations based on the learned coefficient with each other easily. We implement Finite difference for gradient estimation in native python. We share the notebooks with all experiments along with the paper for complete reproducibility of results.

4.4.1 Local Explainability of EASE

EASE is a collaborative filtering model [90] that achieves state-of-the-art performance on three widely used collaborative filtering datasets. As mentioned above, we study the local explanability of EASE model trained on MovieLens 20-Million dataset. EASE is an especially attractive model to start with because it is a linear model and all local explanation methods provide the same explanation; the gradient of the EASE model. Moreover, the gradient of the EASE model are readily available (it's the matrix of parameters) and this gives a way to directly compare the local explanations with the ground truth local explanations/gradients. To be precise, EASE learns a matrix $C \in R^{I \times I}$ where I is the number of items in the catalog. To generate predictions for the j-th item, we pick the j-th column from the matrix C c_j

and perform dot product with the input feature vector x

$$f_j(x) = c_j^T x$$

The gradient at x would be $\nabla_x f_j = c_j$. This local gradient is the global gradient too as EASE is a linear model. We would now like to compute the explanations using the four methods listed above. Fortunately, we can solve all the local explanation techniques analytically for the EASE model. For EASE, all local explanation techniques follow the completeness axiom too (shown below).

- **COFFEE:** The value of EASE model is $f(b) = 0$ for the baseline counterfactual $b = 0$ which implies a constraint of $\Phi^T x_0 = f(x_0)$. The error term $(f(x) - \Phi^T x = (c_j - \Phi)^T x)$ in the MSE gets to a minimum of 0 for $\Phi = c_j$ which also satisfies the completeness constraint.

- **Finite difference method:** Given that the input to the EASE model is already a binary vector, we make use of the simplified finite difference formulation in equation-4.5. The finite difference leaves out only the i-th component of the vector b_j with the respect canceled out. $\phi_i = f(x) - f(x \setminus \{i\}) = c_{i,j}$. $\phi_0 = 0$ according to the first-order Taylor approximation which is equal to $f(b)$ for $b = 0$

- **LIME:** Similar reasoning that applies to COFFEE also holds for LIME. The solution of the LIME weighting regression function is also $\Phi = c_j$. $\phi_0 = 0$ too.

- **Shapley values:** For Shapley values, the finite difference in the inner summation is equal to $c_{i,j}$ for all $S \subseteq F_s \setminus \{i\}$. Therefore, both the inner and outer summations are over a constant value $c_{i,j}$ resulting in $\phi_i = c_{i,j}$ and the local explanation $\Phi = c_j$. Since EASE model's output for the baseline counterfactual (empty set) is zero therefore the intercept $\phi_0 = 0$ for Shapley values hyperplane (same as COFFEE, LIME and Finite difference).

As for the approximation error RMSE, since all local explanation methods recover the gradient of the model and the local gradient is a global gradient too; therefore for any neighborhood around the input query, the first-order Taylor approximation gives a full recovery of the EASE model and we observe zero RMSE. Given the trivial result, we don't plot the result. Applying the local explanation techniques on EASE serves as a good sanity check (for both the techniques and software implementations) and allows us to derive results analytically.

4.4.2 Local Explainability of Multi-VAE

Linear nature of EASE restricts from getting a lot of insights into the different local explanation methods. Therefore, we apply COFFEE and the three baseline approaches on another collaborative filtering model, Multi-VAE [57], that is a deep neural network and is a non-linear model. We again choose b to be the zero input as before. Unlike EASE, Multi-VAE's output is not 0 for baseline counterfactual $b = 0$ therefore we use the version of LIME where we fix the intercept of the linear regression to $f(b)$. Finite difference method cannot incorporate this constraint on the intercept hence we only use it as an analysis tool below. We first visually compare the local explanations generated by COFFEE, LIME and Shapley Values. (We use [2] to get the box arts of movies) Figure-4.3 shows the input queries to all the local explanation methods and figure-4.4 and 4.5 show the top explanations from the explanation models. To the authors of this paper, all the explanation look visually plausible although there are some interesting differences between them. From these explanations, we can't guess which one is most faithful to the Multi-VAE model. This result clearly highlights the difficulty of evaluating different explanation models by a human and is a good exam-

(a) Action query to explain

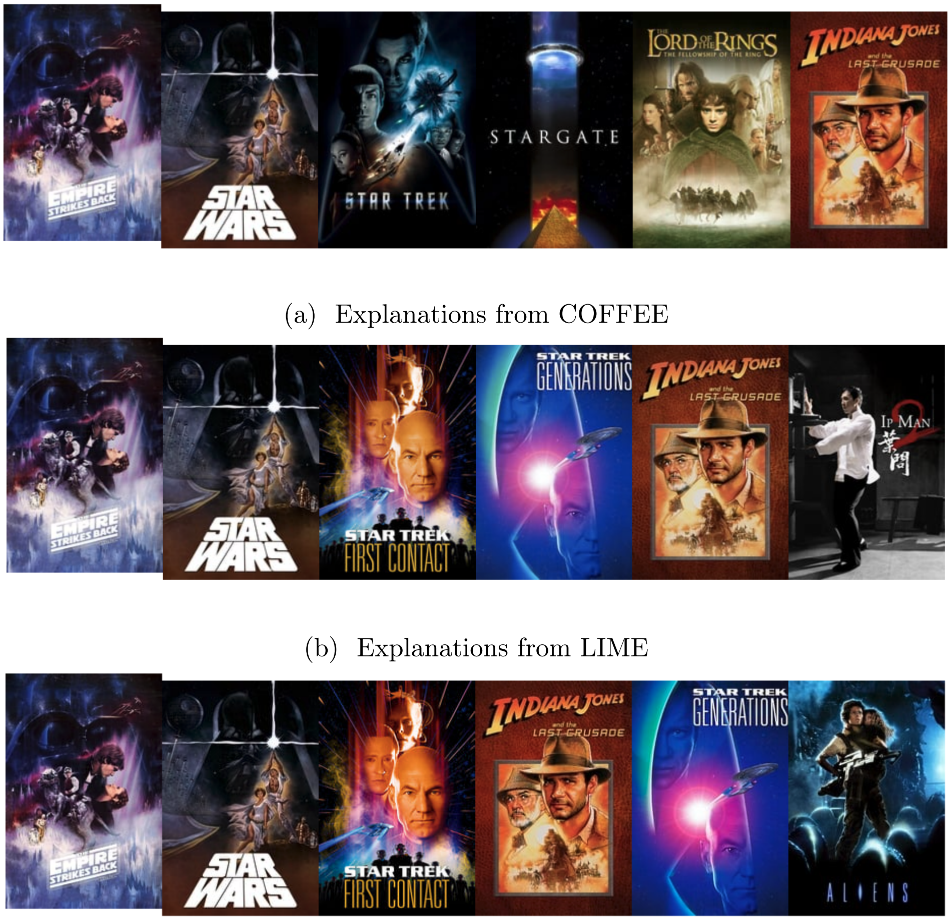
(b) Horror query to explain

Figure 4.3: Two example queries for which we want to understand the prediction coming from the underlying recommendation model. The user play history is a combination of horror, comedy, action and kids movies (146 movies total). (box arts from [2])

ple for our argument to separate the evaluating the faithfulness of the explanations from the ease of human intepretability.

After motivating the use case of quantitatively measuring faithfulness using the visual examples, we now present results that compare the faithfulness of local explanations. As mentioned earlier, the results are computed on a held-out dataset as mentioned. Computing the approximation RMSE on the entire validation dataset let's us compare the statistical significance of differences between the approaches. The results are in the figure-4.6 (first from left). It is encouraging to see that local explanations generated by COFFEE do turn out to be most faithful to the Multi-VAE model compared to LIME and Shapley values validating the correctness of our optimization procedure. Moreover, it is interesting to see that Shapley Value results in more faithful explanations compared to LIME.

To address any concern regarding the computational cost of COFFEE, we compare it with the Kernel SHAP implementation of Shapley values. Figure-4.6 (3rd from left) We find that median computation time of COFFEE is almost 60% faster than Shapley values computation which should make it easy to use in large scale explanation generation use cases. This is a strong reason to use COFFEE over SHAP

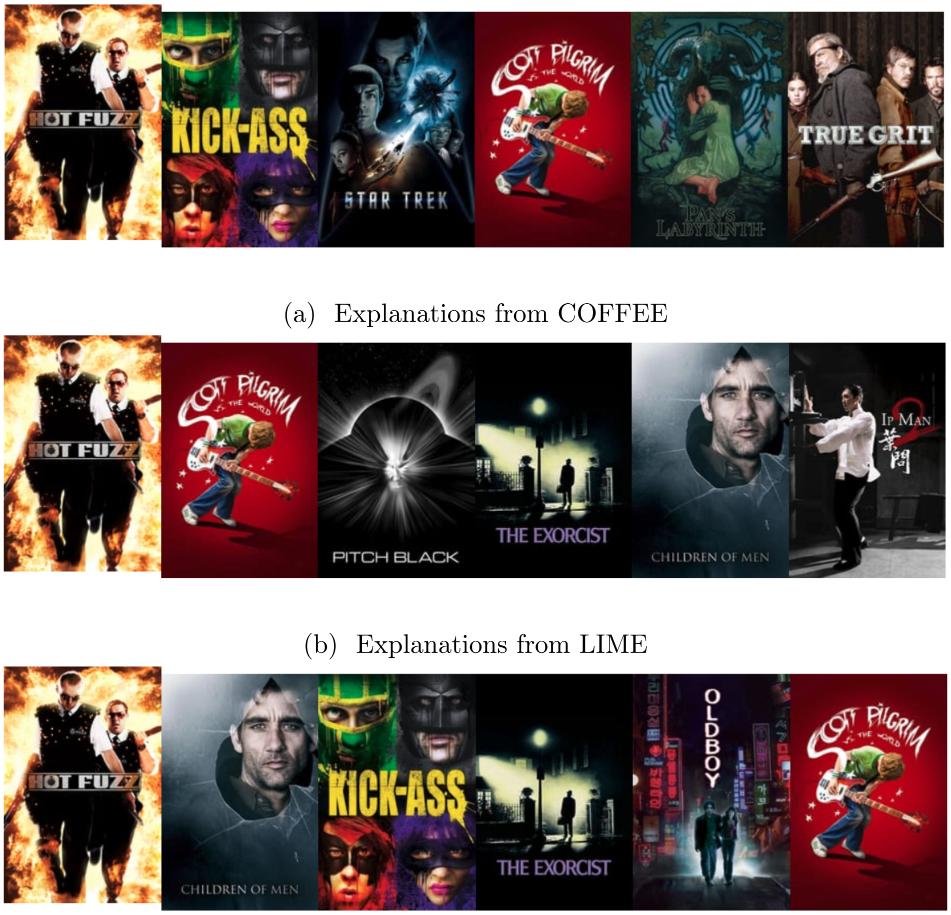


(a) Explanations from COFFEE

(b) Explanations from LIME

(c) Explanations from Shapley values

Figure 4.4: Comparing the visual ranking explanations from COFFEE, LIME and Shapley values for query in fig-4.3a Star Wars: Return of the Jedi. IP man in LIME explanations is difficult to reconcile. Lord of the rings (a fantasy) in COFFEE and Aliens (Space Horror) in Shapley values are interesting picks. (box arts from [2])



(a) Explanations from COFFEE

(b) Explanations from LIME

(c) Explanations from Shapley values

Figure 4.5: Comparing the visual ranking explanations from COFFEE, LIME and Shapley values for query in fig-4.3b (Shaun of the Dead). It's interesting that COFFEE picks Pans Labyrinth (a dark fantasy) and Star Trek as top explanations. Star Trek stars Simon Pegg who is the main cast member in Shaun of the Dead, too. LIME and Shapley contain some odd results like IP Man and Old Boy. (box arts from [2])

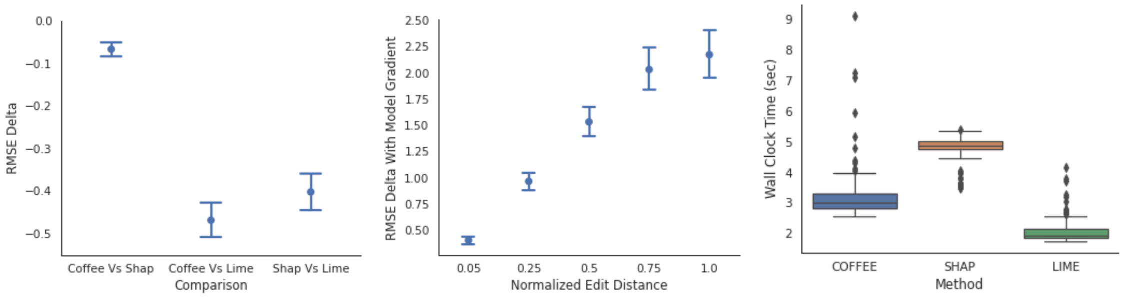


Figure 4.6: Comparing COFFEE, SHAP and LIME. From left to right. First figure compares the delta in RMSE for the three pairs of explanation models. Second figure compares the RMSE of COFFEE with model gradients (estimated using finite difference) over neighborhood of decreasing sizes. Third figure compares the computational speed of the three methods in wall clock time

as it satisfies completeness, is more faithful, visually comparable and computationally faster than Shapley Values.

As mentioned earlier, approximation RMSE based on the gradient of the model in vanishingly small neighborhoods provide a lower bound to the approximation RMSE. To see this, in figure-4.6 (second from left) we compare the approximation RMSE of COFFEE over neighborhoods of decreasing size with RMSE of gradients computed using Finite difference method (finite difference is computed in the smallest neighborhood that we constructed of normalized edit distance of 0.05). This result shows that it behaves more similar to gradients in small neighborhoods compared to the larger ones. Moreover, this result shows that Multi-VAE is a highly non-linear model. Although it is a deep neural network and it is expected to be a non-linear model but through over-regularization the model can behave like a linear model. Finite difference (as well as all local explanation methods) incur greater error in the wider neighborhoods, this indicates that the gradients computed by Finite difference only capture the model behavior locally in small neighborhoods around the input query. This is in stark contrast to the EASE model where the local gradients were global too and RMSE was zero no matter the size of the neighborhood.

4.4.3 Presenting explanations to stakeholders

The end-users of the output of explainability frameworks are humans. These end-users can be machine learning scientists, engineers or customers of a business like movie streaming that uses machine learning models to make recommendations. We began our discussion by separating out the faithfulness of explanations from the ease of human interpretation and the approximation RMSE captures the faithfulness aspect only. Even with the most faithful explanation, we need to figure out a way to present it to humans, the UI element of local explanability research as we referred to it in the start.

The easiest way would be to rank-order the features according to their attribution coefficients ϕ_i and display the features in the sorted order. That was the approach we followed when we presented the explanations in the context of a movie recommendation model (Multi-VAE) visually in the previous section. Using the ranking of movies to understand the explanations would be a challenging task for a human evaluator. It would not only be necessary for the user receiving the explanation to understand each of the movies in the explanation to understand the results but also there will be subjective judgement needed to make sense of the relative ordering of the explanations (Is Star Trek: The First Generation more similar to Star Wars: Return of the Jedi or Star Trek : First Contact?).

We now present a variant of COFFEE that can solve for the local explanations in an alternative feature representation. One may want to transform the features in an entirely new space (probably simpler and more interpretable feature space) and solve the explanation problem in the new feature space. In the case of movie recommendations, one such transformation is to represent the query video play history in some kind of natural language tags and solve the explanation problem in the natural language feature space. For a problem with I movies and T tags if we have video-to-tags mapping matrix G (of size $I \times T$), the transformation is a simple linear operator $X' = XG$. We can then use the same optimization program for COFFEE to solve for the explanation in the new space. The resulting explanations would still be constrained to follow the completeness axiom.

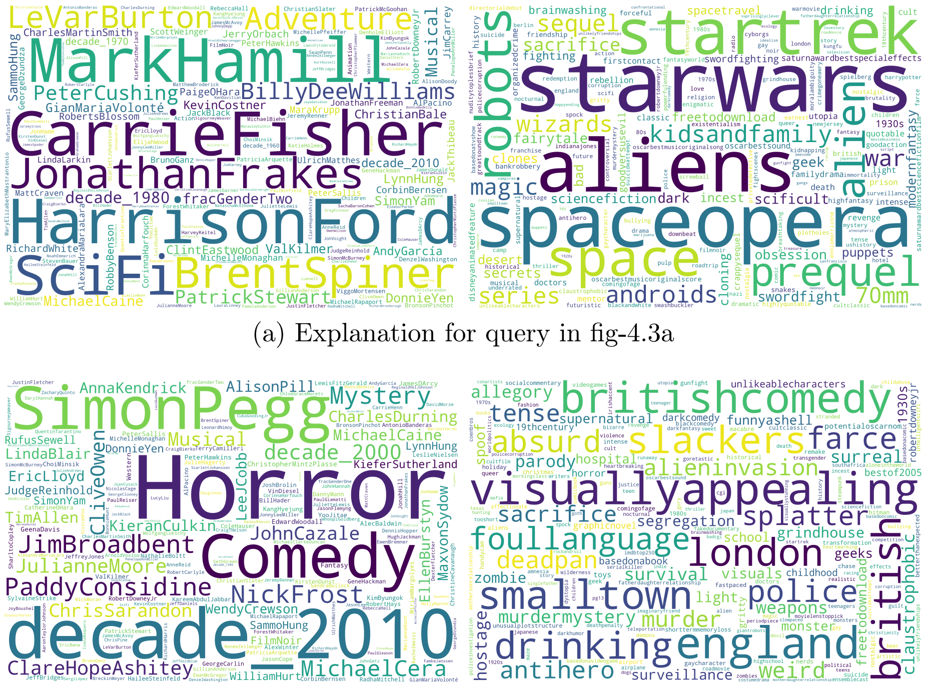
$$\begin{aligned} &\min_{\Phi} \sum_{i=1}^{m} (f(x_i) - (f(x) + \Phi^T G(x_i - x_0)))^2 \\ &\text{s.t. } \Phi^T G x_0 = f(x_0) - f(b) \end{aligned} \tag{4.9}$$

Notice that this feature transformation is very different from feature mapping approach that Shapley values or LIME propose (they require the feature mapping to be reversable whereas G may not be). This implies that in the optimization program opaque function still takes its original input, only the explanation vector operates on the transformed representation.

Figure-4.7 shows two different natural language explanations for the same query shown in figure-4.3 by solving the above optimization program in the transformed feature space for two choices of metadata information available in the MovieLens data, The Movie Database (TMDb) [2] and Tag-Genome [103]. It's very interesting to compare these natural language explanations with the explanations obtained in the original movies space shown in figures-4.4 and 4.5 and guess which one an end-user would prefer. Here we choose to use a word-cloud style presentation of the explanations where the explanation coefficient ϕ_i determine the size of the word in the word-cloud.

4.5 Related Work

There are two dimensions of this work. One is around evaluation of explanation models and the other is developing novel local explanation techniques. As we have previously mentioned, current evaluation of explanability methods almost always rely on human-in-the-loop evaluation using survey data [44]. Our evaluation of the faithful-



(a) Explanation for query in fig-4.3a

(b) Explanation for query in fig-4.3b

Figure 4.7: Natural language explanations for the queries in figure-4.3

ness falls under what is called as functionally-grounded evaluation in the explanability literature. Functionally grounded evaluation is done without humans-in-the-loop and with proxy metrics. On the particular technique of building neighborhoods around the input query and comparing explanations and the opaque model in the neighborhood, there is actually a closely related work that evaluates the robustness of the explanations in neighborhoods that are also constructed in a very similar manner [10]. Similarly there have been other efforts to quantify the faithfulness of gradient based explanability methods by performing sanity checks on the generated explanations [5]. This paper also argues that evaluating explanations purely on visual inspection can be misleading.

We would also like to mention that the techniques we have considered in this paper are known as feature-importance based approaches and Saliency Map based techniques (essentially gradient based approaches popular in Computer Vision tasks). LIME and Shapley values are traditionally considered in the feature importance based techniques but we have provided a gradient based perspective to them in this paper connecting them with Saliency map based techniques. Other forms of local explanation methods are Rule based, Prototypes/Examples based and Counterfactual based.

4.6 Summary

In this chapter, we divide the difficult task of evaluating explanations into two components. Faithfulness to the opaque model, the first of the two components, admits quantitative evaluation. We propose the quantitative evaluation in the form of an approximation error between the local explanations and opaque model. We show that the approximation RMSE can be optimized leading to a new explanation technique that we refer to as COFFEE. Using the same approximation error allows us to compare the faithfulness of explanations from popular models like LIME and Shapley values to explain two recent and highly accurate recommendation models. We find that COFFEE provides the most faithful explanations to the opaque model followed by Shapley values and then LIME. For the second component in evaluation of explanations, ease of human understanding, we show that COFFEE can be used to obtain explanations in alternative feature representations. In the future, we aim to specialize COFFEE for other applications besides recommendation models as the approach is general and can be applied to all the areas where LIME and Shapley values are currently being used.

Chapter 5

Humans and Computers Working Together in Decision Making[1]

This chapter is based on "Selectively Contextual Bandits." It shows how the third mode of human-machine collaboration, humans and computers working together in decision making, maintains utility while reducing harm of machine learning models in the task of image personalization.

Personalization is an integral part of most web-service applications and determines which experience to display to each member. A popular algorithmic framework used in industrial personalization systems are contextual bandits, which seek to learn a personalized treatment assignment policy in the presence of treatment effects that vary with the observed contextual features of the members. In order to keep the optimization task tractable, such systems can myopically make independent personalization decisions that can conspire to create a suboptimal experience in the aggregate of the member's interaction with the web-service. We design a new family of online learning algorithms that benefit from personalization while optimizing the aggregate impact

[1] This chapter was originally published to arXiv with the following citation: Claudia V. Roberts, Maria Dimakopoulou, Qifeng Qiao, Ashok Chandrashekhar, and Tony Jebara. "Selectively Contextual Bandits."

of the many independent decisions. Our approach selectively interpolates between any contextual bandit algorithm and any context-free multi-armed bandit algorithm and leverages the contextual information for a treatment decision only if this information promises significant gains over a decision that does not take it into account. Apart from helping users of personalization systems feel less targeted, simplifying the treatment assignment policy by making it selectively reliant on the context can help improve the rate of learning. We evaluate our approach on several datasets including a video subscription web-service and show the benefits of such a hybrid policy.

5.1 Overview

In web services, users are often faced with a task of selecting an item from a large catalog. Examples include listening to a song from a catalog of 11 million on a music service or watching a video from 500 million user-uploaded videos on a video service. Equally challenging is the decision faced by the web service, the one hosting and presenting the choices to its users. Given limited screen real-estate, which of the 11 million songs or 500 million videos should it present to its users? Personalization has served as the de facto solution to this problem. Web services use proactive and reactive personalized recommendations to guide users to items that are relevant as well as help them discover new items they will also enjoy. From a mix of implicitly learned and explicitly collected features of each user and item, the services prune the list of available options that they present to each user and provide a personalized experience whilst doing so. While there are often competing objectives when deciding the optimal item or set of items to present to a user, overall, the goal is to satisfy the user in order to increase engagement with the service and retain the user over time.

Human preferences, however, are more complicated and nuanced than any one model can capture perfectly. When there are only 10 choices available, for example, one is unable to provide a truly bespoke experience. The personalization algorithm in this case is forced to generalize and put a user in one of, say, 10 buckets. Furthermore, the personalization systems are operating on limited user information due to the personalization privacy paradox [109, 107, 4, 11], the European Union's General Data Protection Regulation, and various privacy and data protection laws. Often times, these systems operate on incomplete information such as impression data and past behavior as proxies for personal taste preferences. Personalization systems are not truly personalizing per se, they are simply attempting to learn a model for what a user might find generally appealing based on information the user might have explicitly provided via user questionnaires, past behavior, and the behavior of similar users. In other words, at some point, these personalization systems must generalize in order to make the problem more tractable, but in the process, they may conspire to create a sub-optimal user experience for individuals and groups of users.

To illustrate the aforementioned point, imagine a context-aware event recommender that recommends a user to attend an event where the only thing that person has in common with any of the other attendees is their religious affiliation, causing that user to feel one-dimensional and reducing the perceived utility of the event recommender service. Web services that provide personalized experiences aim to ensure that they are honing in on meaningful features of their user base and not simple generalizations of their users. This is especially important and challenging given the growing awareness and concern about algorithmic bias and algorithms' potential to amplify harmful societal stereotypes via information retrieval systems [18, 110, 65, 45].

For this purpose, recommendation engines in industry rely on *multi-armed bandit* algorithms to learn how to optimally recommend items to the users. In particular *contextual multi-armed bandits*, which attempt to learn the optimal personalized recommendation for each user given user information, i.e., the context, are widely used in practice. While personalization in web services makes the catalog more accessible to a user by reducing the burden of choice, it has the potential to isolate the user and unintentionally create "filter bubbles". This may happen even if the users are modeled by the service only using implicit behavioral data, since the feedback loops of online learning can cause an increased focus on narrower interests by the content publisher. This has the potential of hindering a user from participating in social conversation within their network on specific content, as the content that is served to different individuals in the network may differ drastically.

Additionally, as the user context gets more detailed and higher-dimensional, the model estimation of a contextual bandit becomes more challenging and the regret bounds may take longer to converge. [23] have shown that all else equal, using assignment policies that are simpler (in terms of how they vary with contextual variables) in the early learning phases of the algorithm can improve the rate of learning and decrease regret. Finally, simpler, unpersonalized assignment rules may have other advantages as well; for example, [51] highlight the advantages of simplicity for interpretability in health applications of contextual bandits. On the other hand, if all users had an identical exposure to the catalog, then there is increased potential for social engagement on common topics. However, given the typical sizes of the catalogs, the user experience would be significantly worse as it would be harder to find content that appeals to an individual user personally.

We design a new family of bandit algorithms that interpolate between unpersonalized and personalized recommendations. This new family of algorithms aims to mitigate the aforementioned downsides of over-personalization. In particular, we investigate a class of online algorithms known as contextual bandits and their application to personalization. While the additional data from context is valuable, when multiple decisions are being taken (sequentially or jointly) based on contextual bandits, it is possible that sub-optimal model estimation may result in mis-calibrated outputs, for instance, a homogeneous user experience. Furthermore, as context gets more detailed and higher-dimensional, some regret bounds take longer to converge. However, in most problem domains, context is better than the lack thereof. This new family of algorithms, called *selectively contextual bandits* (SCB), chooses between a contextual bandit decision and a non-contextual bandit decision in every iteration. The context is only used when the contextual decision yield a predicted reward lift higher than a parameter δ. The δ parameter is annealed at a rate that depends on the regret bound of the upper confidence bound (UCB) such that we optimize the regret bound to remove dependence on the dimensionality of the context. Alternatively, we can regularize the estimator of Lin-UCB towards a non-contextual setting. For instance, if we can show that having a δ unpersonalized (non-contextual policy) can incur at most $\mathcal{O}(1)$ regret per time step over an optimal contextual bandit. However, if we set δ to shrink at a certain rate, for instance, $\delta = 1/t$ then we may still get a logarithmic regret.

We evaluate our results on several contextual bandit data-sets such as classification based public datasets as well as a large-scale proprietary dataset. In the industry dataset setting, we show that it is possible to decrease the amount of personalization without hurting regret style metrics and can sometimes even improve upon them. In the classification setting, we show that it is possible to achieve regret bounds equivalent to fully contextual baselines while reducing the number of contextual treatments in favor of non-contextual treatments.

5.2 Preliminaries

5.2.1 Problem Formulation

The problem of image personalization can be formulated as a stochastic contextual bandit problem. In the stochastic contextual bandit setting (see [16] for a survey), there is a finite set of arms $\mathcal{A} = \{1, \ldots, K\}$. At time t, the environment produces

$(x_t, r_t(1), \ldots, r_t(K)) \sim \mathcal{D}$, where x is a d-dimensional context and $r_t(a)$ is the reward associated with each arm $a \in \mathcal{A}$.

When the recommender policy selects arm a_t, the observables are $(x_t, a_t, r_t(a_t))$. In particular, there is partial observability and only the reward $r_t(a_t)$ for the chosen arm a_t is observed. At each time t, the optimal assignment is the arm with the maximum expected reward and is denoted as $a_t^* = \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}[r_t(a)|x_t]$.

The goal of the policy is to find an assignment rule that sequentially assigns an arm to minimize the cumulative expected regret over horizon T

$$\text{Regret}(T) = \sum_{t=1}^{T} \mathbb{E}[r(a_t^*) - r(a_t)]$$

where the assignment rule is a function of the previous observations $(x_\tau, a_\tau, r_\tau(a_\tau))$ for $\tau = 1, \ldots, t-1$ and of the new context x_t.

5.2.2 Exploration vs. Exploitation

Therefore, the decision-maker has to balance exploring arms for which there is limited knowledge in order to learn and exploiting the accumulated knowledge in order to attain higher rewards.

Two established approaches for balancing the exploration vs. exploitation trade-off in stochastic contextual bandits are the linear upper confidence bound (LinUCB) algorithm [53] and the linear Thompson sampling (LinTS) algorithm [6] as well as their generalized linear (particularly logistic) counterparts [19, 55]. These algorithms postulate that the expected reward of arm a conditional on the context x can be modeled as a linear or a generalized linear function of the context with unknown parameters w_a. Then, at each time t, they use the historical observations $\{(x_\tau, a_\tau, r_\tau(a_\tau))\}_{\tau=1}^{t-1}$ and regularized linear or logistic regression to form an upper confidence bound or posterior (exact or approximate) on the unknown parameters θ_a of each arm $a \in \mathcal{A}$.

Finally, the upper confidence bound or the posterior is used to balance exploration vs. exploitation when deciding the arm a_t for the new context x_t.

A simple and popular heuristic for bandit problems is the ϵ-greedy exploration strategy [92]. According to this strategy, at every time t the decision-maker computes point estimates $\hat{w}_a$ for each arm $a \in \mathcal{A}$ based on the historical observations $\{(x_\tau, a_\tau, r_\tau(a_\tau))\}_{\tau=1}^{t-1}$ and uses these point estimates to find the arm with the highest predicted expected reward for context x_t. Then, the decision-maker selects the best predicted arm with probability $1-\epsilon$ and with probability ϵ selects an arm from $\mathcal{A}$ uniformly at random.

Both Thompson sampling and UCB for contextual bandits have strong regret bound guarantees, however Thompson sampling tends to perform much better in practice [19, 80, 24]. On the other hand, ϵ-greedy has sub-optimal guarantees compared to both Thompson sampling and UCB, but is popular in practice due to its simplicity and generally good performance.

5.3 Related Work

We focus on algorithms for the stochastic contextual bandit problem with binary rewards, but all the presented algorithms can be extended to real number rewards. We first present two well-known baselines from the literature; a contextual bandit algorithm that models the expected reward of each arm conditional on the context as a logistic function and a non-contextual bandit algorithm that does not take into account the context during the decision making. Subsequently, we present our approach, the selectively contextual bandit, which interpolates between the contextual and the non-contextual bandit depending on the predicted benefit from taking the contextual information into account in each time period. All three algorithms can be paired with any of the exploration schemes outlined in Section 5.2.2. Due to space limitations, we present the Thompson sampling version of the algorithms, which can be readily adapted to the UCB and ϵ-greedy versions.

5.3.1 K-Armed Bernoulli Bandit

In the non-contextual formulation, the decision-maker does not take into account the context of every time period but rather tries to learn the unpersonalized, globally optimal arm while balancing the exploration vs. exploitation trade-off. One straightforward approach is to model this problem as a K-armed Bernoulli bandit with independent arms $\mathcal{A}$ [96]. In this formulation, the reward of arm a follows a Bernoulli distribution with mean θ_a. It is standard to model the mean reward of arm a using a Beta distribution with parameters α_a and β_a, since it is the conjugate distribution of the binomial distribution. At every time t, the agent draws a sample mean reward $\hat{\theta}_a \sim \text{Beta}\left(\alpha_a, \beta_a\right)$ for each arm $a \in \mathcal{A}$ and selects arm $a_t = \text{argmax}_{a \in \mathcal{A}} \hat{\theta}_a$. Based on the observed reward $r_t(a_t)$, the decision-maker updates the posterior distribution on θ_a. Algorithm 1 presents the approach.

Algorithm 1 Non-Contextual K-Armed Bernoulli Bandit

```
Require: Initial $\alpha_a$ and $\beta_a$ for all $a \in A$ (default value: 1)
for $t = 1, \ldots, T$ do
    for each arm $a \in A$ do
        Sample $\hat{\theta}_a \sim \text{Beta}\left(\alpha_a, \beta_a\right)$
    end for
    Select arm $a_t = \text{argmax}_{a \in A} \hat{\theta}_a$
    Observe reward $r_t(a_t)$, where $r_t \sim \mathcal{D}(\cdot|x_t)$
    if $r_t(a_t) = 1$ then
        $\alpha_{a_t} = \alpha_{a_t} + 1$
    else
        $\beta_{a_t} = \beta_{a_t} + 1$
    end if
end for
```

5.3.2 Generalized Linear Bandit

Linear bandits [53, 6] and generalized linear bandits (particularly logistic) [19, 55] are widely used in web services for the personalization of news recommendation, advertising and search. Generalized linear bandits (logistic regression in particular) have demonstrated stronger performance than linear bandits in many applications where rewards are binary. In this section, we model the stochastic contextual bandit problem as a generalized linear bandit, as in [19].

The decision-maker models the expected reward of arm a, $\mu_a = \mathbb{E}[r(a)|x]$, as a logistic function of context x with parameters $\theta_a \in \mathbb{R}^d$, $\mu_a = \mathbb{P}(r(a) = 1|x) = \sigma(\theta_a^\top x)$ where $\sigma(z) \equiv \frac{1}{1+\exp(-z)}$ is the sigmoid function. The posterior distribution on the parameters θ_a of each arm $a \in \mathcal{A}$ is approximated by a multivariate Gaussian distribution updated via the Laplace approximation. Specifically, the decision-maker starts with a multivariate Gaussian prior each θ_a with mean $\boldsymbol{\mu}_0 = \mathbf{0} \in \mathbb{R}^{\ell m}$ and covariance matrix $\boldsymbol{\Sigma}_0 = \lambda \cdot \mathbb{I}_{\ell m}$, where $\mathbb{I}_{\ell m}$ is the $\ell m \times \ell m$ identity matrix and λ is a regularization parameter.

Algorithm 2 Generalized Linear Bandit

```
Require: Parameters of weight prior $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$
Draw weight sample $\hat{\mathbf{w}} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
for $t = 1, \ldots, T$ do
    for each arm $\mathbf{a} \in A$ do
        Compute $\hat{\theta}(\mathbf{a}) = \frac{1}{1+\exp(-\hat{\mathbf{w}}^\top \mathbf{x}_\mathbf{a})}$
    end for
    Select arm $\mathbf{a}_t = \text{argmax}_\mathbf{a} \hat{\theta}(\mathbf{a})$
    Observe reward $r_t \sim \mathcal{D}(r|\mathbf{a}_t)$
    Update weight posterior parameters $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$
    Draw a new weight sample $\hat{\mathbf{w}} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$
end for
```

As in [19], the posterior updating at time t is as follows. Before the observation at time t

$$\begin{aligned}\log(\mathbb{P}(\mathbf{w})) =& -\frac{1}{2}k\log(2\pi) - \frac{1}{2}\log(\mathbf{\Sigma}_{t-1}) - \\ &- \frac{1}{2}(\mathbf{w}-\boldsymbol{\mu}_{t-1})^{\top}\mathbf{\Sigma}_{t-1}^{-1}(\mathbf{w}-\boldsymbol{\mu}_{t-1}).\end{aligned}$$

The log-likelihood of the observations at time t is

$$\log(\mathbb{P}(r_t|\mathbf{x}_{\mathbf{s}_t},\mathbf{w})) = r_t\log\left(\sigma(\mathbf{w}^{\top}\mathbf{x}_{\mathbf{s}_t})\right) + (1-r_t)\log\left(1-\sigma(\mathbf{w}^{\top}\mathbf{x}_{\mathbf{s}_t})\right).$$

The log-posterior of $\mathbf{w}$ at time t is

$$\begin{aligned}\log(\mathbb{P}(\mathbf{w}|\mathbf{x}_{\mathbf{a}_t},r_t)) \propto& -\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu}_{t-1})^{\top}\mathbf{\Sigma}_{t-1}^{-1}(\mathbf{w}-\boldsymbol{\mu}_{t-1}) + \\ &+ r_t\log\left(\sigma(\mathbf{w}^{\top}\mathbf{x}_{\mathbf{a}_t})\right) + (1-r_t)\log\left(1-\sigma(\mathbf{w}^{\top}\mathbf{x}_{\mathbf{a}_t})\right)\end{aligned}$$

The posterior mean of $\mathbf{w}$ is the maximum a posteriori estimate $\boldsymbol{\mu}_t = \mathbf{w}_{\text{MAP}} = \operatorname{argmax}_{\mathbf{w}}\log(\mathbb{P}(\mathbf{w}|\mathbf{X},\mathbf{r}))$ and the posterior covariance matrix of $\mathbf{w}$ is $\mathbf{\Sigma}_t^{-1} = \mathbf{\Sigma}_{t-1}^{-1} + \sigma(\mathbf{w}_{\text{MAP}}^{\top}\mathbf{x}_{\mathbf{a}_t})(1-\sigma(\mathbf{w}_{\text{MAP}}^{\top}\mathbf{x}_{\mathbf{t}}))\mathbf{x}_{\mathbf{a}_t}\mathbf{x}_{\mathbf{a}_t}^{\top}$. To choose the next arm, the agent draws a weight sample $\hat{\mathbf{w}} \sim \mathcal{N}(\boldsymbol{\mu}_t, \mathbf{\Sigma}_t)$ and forms an estimate of the expected reward $\hat{\theta}(\mathbf{a}) = \sigma(\hat{\mathbf{w}}^{\top}\mathbf{x}_{\mathbf{a}})$ of each arm $\mathbf{a} \in \mathcal{K}$ based on this weight sample. Then, the agent plays arm $\mathbf{a}_t = \operatorname{argmax}_{\mathbf{a}}\hat{\theta}(\mathbf{a})$. Algorithm 2 outlines the approach.

5.4 Selectively Contextual Bandit

Given the constituent policies - one contextual and the other non-contextual, we now provide the details of a hybrid policy that selectively switches between the two policies. At each time step, the two policies are used to determine their optimal arm assignments. If the arms selected from the two policies are different, rewards for the two arms using the contextual policy are estimated. The estimated rewards are then compared to determine if the predicted reward from the contextual policy for the arm selected by the contextual policy is significantly better than the arm selected by the non-contextual policy. If so, SCB selects the contextual winner, if not the non-contextual winner is used. The policies are then updated with the observed reward once the SCB makes its final selection. The overall algorithm is sketched out in Algorithm 3.

Algorithm 3 Selectively contextual bandits

```
Require: $\pi$ : Non-contextual policy, $\pi_c(x)$ : Contextual policy
for $t = 1, \ldots, T$ do
    Select arm $a_{nc} = \pi$
    Select arm $a_c = \pi_c(x)$
    Predict reward $r_{nc}(a_{nc})$ and $r_c(a_c)$
    if $\delta(r_c(a_c), r_{nc}(a_{nc})) > \lambda$ then
        $a_{scb} = a_c$
    else
        $a_{scb} = a_{nc}$
    end if
    Update $\pi$ and $\pi_c(x)$
end for
```

In our scheme, we evaluate two different ways of calculating the non-contextual winner: mean and beta-Bernoulli. Further, we also consider two different formulations of the delta operation in the algorithm: ratio or relative difference (please find the details in Section 5.5.1). Finally, we allow for shrinking or annealing the δ threshold by a constant decay rate on a specified time schedule. This is implemented by shrinking the δ threshold by a specified constant rate at specified epochs.

5.5 Experiments

We hypothesize that a fully personalized policy should always perform better in regret analysis comparisons. But there exist real-world use cases where a hybrid policy may be desired. For example, a hybrid policy may have benefits such as avoiding filter bubbles and enabling users to participate in the zeitgeist or enabling greater overlap in shared experiences in the ever increasing personalized web. No public datasets exist to verify this claim, however. So we restrict our experiments to show that the regret of a hybrid policy is comparable to a purely contextual one. We evaluate our approach in a classification setting using public datasets, and we present and discuss our results in this section.

5.5.1 Experiments on Public Datasets

Multiclass Classification with Multi-Armed Bandits. When experimenting with and comparing different contextual bandit algorithms, it is common to transform multiclass classification tasks into multi-armed bandit formulations [28]. We make the assumption that the observations are sampled from a fixed distribution and are independent and identically distributed. In both the non-contextual and contextual bandit setting, the number of classes corresponds to the number of arms. In the contextual setting, the features of each data sample correspond to that sample's context. Accompanying each data sample is the ground truth class label. In a multiclass classification problem, the task is to learn a model that correctly assigns the correct class label to each data sample in a test set. Correspondingly, in the adaptation to a bandit problem, the goal is to learn an assignment policy that assigns the optimal (correct) arm to each sample. In the multiclass classification setting, we are attempting to learn a model that minimizes the classification error, which corresponds to the policy's expected regret in the multi-armed bandit setting. In our implementation of the various non-contextual bandits, the assignment policy opts not to use the sample's features (or context) during arm assignment or when updating the posterior distribution of each arm. In each non-contextual bandit's contextual counterpart, we do leverage the additional information of the sample's features to inform arm assignment and when updating the posterior distributions. In both the non-contextual and contextual settings, after each time time-step t, an arm is assigned to sample x_t. If the arm assignment is correct, the agent incurs a reward of one. If the arm assignment is incorrect, then the agent incurs a regret of 1. When comparing the performance of various contextual bandit algorithms in this multiclass classification problem setting, it is common to perform regret analysis and visualize the regret graphs over the history of observations using the normalized cumulative regret.

Experimental Set-up. We use the Open Media Library (OpenML) [101] to collect 20 publicly available classification datasets. The datasets we use span various domains such as healthcare, biology, ecology, and computer vision and vary with regards to their attributes i.e., number of observations, classes, and features. As part of preprocessing, the categorical feature columns are one-hot encoded. Before each run, we randomly shuffle the dataset. We run our suite of bandit algorithms on each of the 20 datasets for different model hyperparameters. For the SCB bandits, we vary the various input parameters including the delta threshold value, the delta shrinkage rate, and delta shrinkage schedule. These SCB input parameters control the amount of non-contextual decisions that are made in favor of contextual ones, with the option to anneal δ by a constant rate at various time-steps in the horizon.

SCB Input Parameter Selection. In this paragraph we offer a more in-depth explanation on how we pick the initial delta rates as well as how we selected the subsequent annealing rate and annealing schedule. But first, we provide some intuition for why delta rate annealing may be desired or even necessary in some cases. In the earlier timesteps of a contextual model that is learning the optimal arm assignment policy for a particular dataset, the model has not yet learned a good policy because it has not seen enough samples. Thus, we choose a higher δ value in this lower sample regime, making noncontextual decisions more frequently. As time goes on, the contextual bandit begins to learn a better arm assignment policy so we shrink the rate accordingly to account for this higher degree of confidence. During our evaluation of our SCB models on the OpenML datasets, we began by selecting an initial delta rate of 1.0 for SCB models using ratios for expected reward comparisons and 0.0 for SCB models using relative differences for expected reward comparisons. This was for sanity checking that SCB policies with thresholds meant to select the contextual winner every time matched the fully contextual baseline policy. We then chose reasonably high thresholds, e.g. 1.5 for SCB policies based on ratio comparisons and 0.5 for those based on relative difference comparisons, without an annealing schedule to observe the commutative regret over the horizon using a policy that selects a large number of noncontextual decisions in favor of contextual ones. As expected, the fully contextual bandit always performed better than our SCB models under these high initial delta rates. We then steadily decreased this initial decay rate until finding a starting rate that allowed our policy to roughly match the regret bounds of the fully contextual policy. Having found this initial delta rate, we then applied a shrinkage scheduler that was not aggressive, for doing so also decreases the number of noncontextual decisions that are made. Future work includes automatically finding the SCB parameters that allow the maximal noncontextual decisions to be made while staying on par or improving upon the fully contextual policy.

Compared Models

The following lists the multi-armed bandit algorithms we evaluated for performance comparison. It includes different flavors of SCB bandits and baseline models. If the model name ends in "Ratio," the ratio of the contextual to noncontextual expected reward is compared against the SCB delta threshold at each time-step t to decide between differing arm assignments. If the model name ends in "Diff," then the relative difference of the contextual and noncontextual expected reward is compared against the SCB delta threshold. $\epsilon = .2$ for all ϵ-greedy bandits evaluated.

- **IndependentBernoulliArmsEGAgent**[96] non-contextual beta-Bernoulli bandit using ϵ-greedy as the explore/exploit strategy.
- **LogisticRegressionEGAgent**[19, 55] contextual bandit that models the expected reward of each arm as a logistic function using ϵ-greedy as the explore/exploit strategy.
- **SCBEGAgent_Ratio** SCB agent that interpolates between treatment decisions made by IndependentBernoulliArmsEGAgent and LogisticRegressionEGAgent, using ratio of expected rewards for comparisons against SCB δ parameter.
- **meanSCBEGAgent_Ratio** SCB agent with LogisticRegressionEGAgent as its base model, using ratio of expected rewards for comparisons against SCB δ parameter. The noncontextual winning arm is determined to be the arm with the maximum average expected reward taken across all contexts in the history.
- **SCBEGAgent_Diff** SCB agent that interpolates between treatment decisions made by IndependentBernoulliArmsEGAgent and LogisticRegressionEGAgent, using the relative difference of expected rewards for comparison against SCB δ parameter.
- **meanSCBEGAgent_Diff** SCB agent with LogisticRegressionEGAgent as its base model, using relative differences during comparisons against SCB δ parameter. The noncontextual winning arm is determined to be the arm with the maximum average expected reward taken across all contexts in the history.
- **IndependentBernoulliArmsTSAgent** non-contextual beta-Bernoulli bandit using Thompson Sampling as explore/exploit strategy.

- **LogisticRegressionTSAgent**[6] contextual bandit that models the expected reward of each arm as a logistic function using Thompson Sampling as the explore/exploit strategy.

- **SCBTSAgent_Ratio** SCB agent that interpolates between treatment decisions made by IndependentBernoulliArmsTSAgent and LogisticRegressionTSAgent, using ratio of expected rewards for comparisons against SCB δ parameter.

- **meanSCBTSAgent_Ratio** SCB agent with LogisticRegressionTSAgent as its base model, using ratio of expected rewards for comparisons against SCB δ parameter. The noncontextual winning arm is determined to be the arm with the maximum average expected reward taken across all contexts in the history.

- **SCBTSAgent_Diff** SCB agent that interpolates between treatment decisions made by IndependentBernoulliArmsTSAgent and LogisticRegressionTSAgent, using the relative difference of expected rewards for comparison against SCB δ parameter.

- **meanSCBTSAgent_Diff** SCB agent with LogisticRegressionTSAgent as its base model, using the relative difference of expected rewards for comparison against SCB δ parameter. The noncontextual winning arm is determined to be the arm with the maximum average expected reward taken across all contexts in the history.

- **IndependentBernoulliArmsUCBAgent**[53] non-contextual beta-Bernoulli bandit using UCB as explore/exploit strategy.

- **LogisticRegressionUCBAgent** contextual bandit that models the expected reward of each arm as a logistic function using UCB as the explore/exploit strategy.

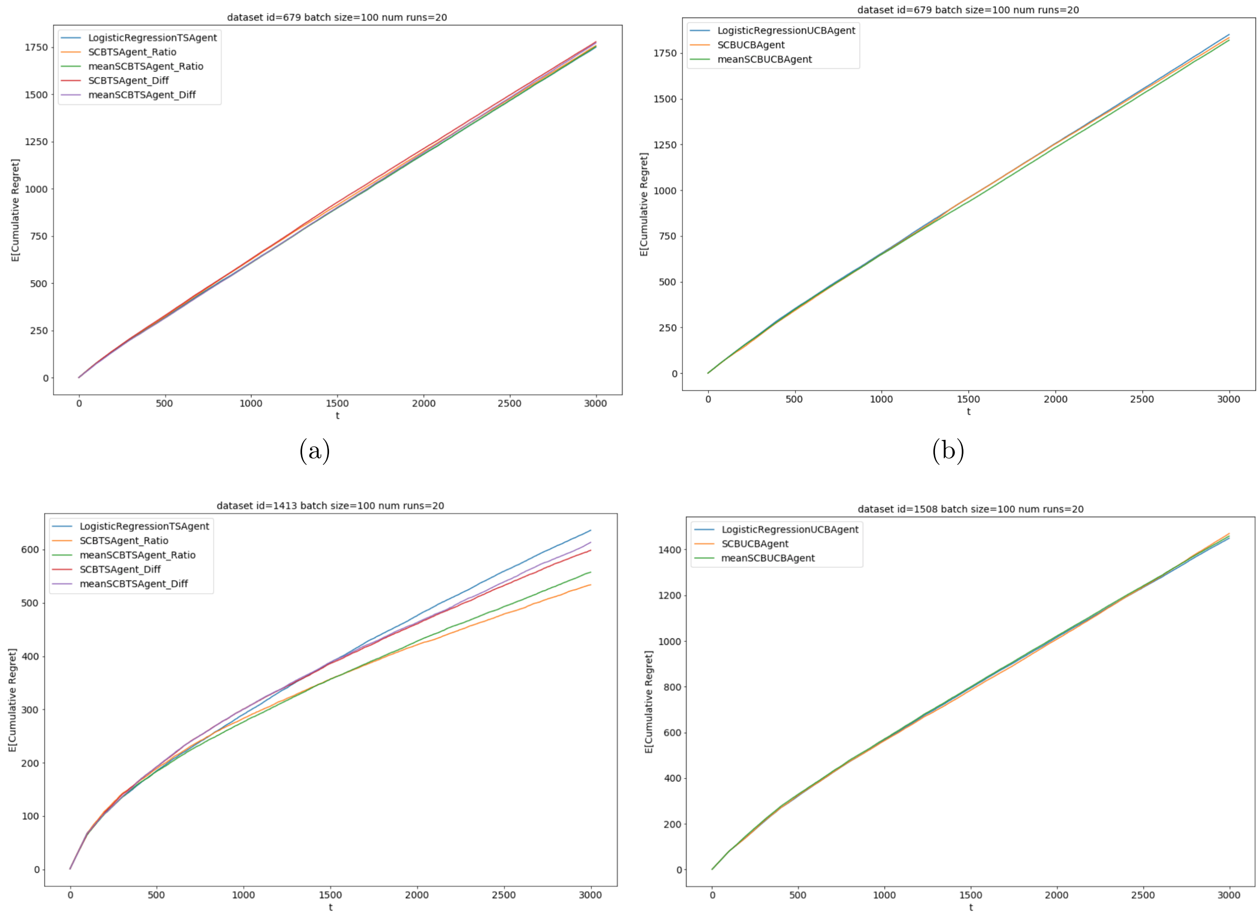
- **SCBUCBAgent** SCB agent that interpolates between treatment decisions made by IndependentBernoulliArmsUCBAgent and LogisticRegressionUCBAgent, using the relative difference of expected rewards for comparison against SCB δ parameter.

- **meanSCBUCBAgent** SCB agent with LogisticRegressionUCBAgent as its base model, using the relative difference of expected rewards for comparison against SCB δ parameter. The noncontextual winning arm is determined to be the arm with the maximum average expected reward taken across all contexts in the history.

Results

Figure 5.1 shows a regret analysis comparison between the various SCB and baseline algorithms described in Section 5.5.1 on three different OpenML public multiclass classification datasets. Due to space constraints, we have not included the full set of graphs for each of the 20 datasets on each of the 16 models. The x coordinate is the timestep and the y coordinate is the normalized cumulative regret $\frac{1}{n}\sum_{t=1}^{n}(1-r_t(a_t))$. t is the timestep, the arm selected by the bandit at time t is $a_t \in \{1, 2, ...K\}$, where K is the number of arms or classes in the dataset with n observations and with reward function $r_t(a_t) = 1\{a_t = c_t\}$ where c_t is the true class of context x_t. The lower the cumulative regret the better. We run each agent 20 times over each dataset (the dataset is reshuffled at the beginning of each run). We set the horizon to 3000 observations, updating the prior distributions and the logistic models after every batch of 100 timesteps.

Figure 5.1a shows a comparison of the various Thompson Sampling bandits we evaluated on OpenML dataset id 679, a dataset of sleep state measurements. We observe that all five bandits display the same regret curves. However, while the baseline model, LogisticRegressionTSAgent, made contextual decisions at every timestep t,



(a)

(b)

(c)

(d)

Figure 5.1: Regret analysis comparison between SCB and baseline algorithms on OpenML public multiclass classification datasets.

the SCBTSAgent_Ratio bandit chose noncontextual decisions over contextual ones 7.2% of the time averaged across the 20 runs. Similarly, the meanSCBTSAgent_Ratio bandit chose noncontextual decisions over contextual ones 10.4% of the time, SCBTSAgent_Diff 12.1% of the time, and meanSCBTSAgent_Diff 14.2% of the time. Figure 5.1b shows a comparison of the various UCB bandits we evaluated on the same dataset. Again, we observe that all three bandits display the same regret curves. The baseline model, LogisticRegression UCBAgent, made contextual decisions at every timestep, SCBUCBAgent chose the noncontextual decision over the contextual decision a mean percentage of 33.4% of the times, and meanSCBUCBAgent 14.7% of the times.

Figure 5.1c shows a comparison of the various Thompson Sampling bandits we evaluated on OpenML dataset id 1413, the Iris dataset. We observe that the four SCB bandits outperform the baseline LogisticRegressionTSAgent. However, it is interesting to note that the number of noncontextual decisions made in favor of contextual ones was minimal. The SCBTSAgent_Ratio bandit chose noncontextual decisions over contextual ones only .43% of the time averaged across the 20 runs. Similarly, the meanSCBTSAgent_Ratio bandit chose noncontextual decisions over contextual ones .43% of the time, SCBTSAgent_Diff made a few more noncontextual decisions at 3.9%, and meanSCBTSAgent_Diff at 1.4%. Figure 5.1d shows a comparison of the various UCB bandits we evaluated on the OpenML dataset id 1508, a user knowledge dataset. Here we observe again all three models performing on par with each other with the SCBUCBAgent choosing noncontextual decision over the contextual decision a mean percentage of 9.6% and meanSCBUCBAgent 12.1%.

The results from the experiments run on publicly available multiclass classification datasets showed us that in many cases, there exists a hybrid policy that reduces dependence on context whilst achieving regret bounds that are on par with fully contextual baseline algorithms. In a few cases, we found a policy that made it possible to outperform a fully contextual policy. Likewise, we also observed a few cases where we were unable to find a delta value that reduced the number of contextual decisions without strictly hurting performance. Overall, we believe that SCB is an algorithm that practitioners can employ if they wish to find policies that reduce the number of the contextual decisions (or conversely, increase the number of common treatments that are assigned) without significantly impacting performance.

5.5.2 Experiments on proprietary Dataset

In an online video subscription setting, promotional title artwork has undergone a paradigm shift. The burden has shifted from attempting to appeal to as many people as possible with two or three posters to attempting to appeal to a single user in a single online video viewing session. The goal is no longer solely about capturing public joy but about capturing individual attention. Acknowledging that any given title may cover a range of thematic themes with an ensemble of cast members that may each appeal to a particular user in a different way, personalized artwork selection allows online video services to hone in on the varying taste profiles of their member-base. Artwork personalization via online learning with explore exploit and contextual multi-armed bandits is important to the member experience in a video subscription service with thousands of titles and shows available to watch. The image personalization system algorithmically selects 1 of N possible images from the title image suite to display to a user at a specific point in time or viewing session. The goal is to emphasize different themes through various artwork according to some context (user viewing history, country) in order to capture individual preferences for cast members, genres, artistic themes, etc.

Evaluation Metrics. We define the Click through rate (CTR) of an item as the fraction of users who engaged with an item after being presented with it. A controlled randomized uniform exploration of the candidate items is done by a logging policy which provides us with a dataset to evaluate the proposed SCB algorithm. We compute off-policy replay metrics by following the method described in [54] to evaluate and compare the unbiased offline performance of the various online learning policies we experimented with. This method allows us to answer counterfactual questions based on the logged exploration data. In other words, we can compare offline what would have happened in historical sessions under different scenarios if the recommender system used different algorithms in an unbiased way.

Experimental Set-up. We compared SCB against a multi-armed contextual bandit policy and a non contextual multi-arm bandit policy. The contextual information of a user is represented as a feature vector provided as input to the model for predicting the probability of reward for each item i. Features primarily encapsulate the user's past engagement behavior. We tested one class of SCB models, based on the same underlying logistic regression model used in production and using the ratio of expected rewards for comparisons against the SCB δ parameter. If the ratio of the expected reward of the contextual selected arm (or personalized winning image) to the expected reward of the noncontextual selected arm (or unpersonalized winning image) is not above some threshold δ, then we show the unpersonalized winning image. We calculate the policy level take-rate and repeat the experiment for various SCB δ threshold values. As we sweep the delta from 1 to 10, we move from a fully personalized image selection experience to a fully unpersonalized one, with all members being impressed with the same images for the same titles. We do not anneal the delta rate over the history of contexts, i.e. we keep the delta rate constant. We ran experiments on different asset types and across data streams from different months and days.

Compared Policies

We considered three different policies for selecting the noncontextual winning image.

- **SCB Global Majority Vote** The noncontextual selected image for title t is calculated as the global majority vote image for t across all contexts. Within the context of user u and title t, if the ratio between the expected reward of the contextual selected image to the expected reward of the noncontextual selected image is not greater than SCB input parameter δ, then show user u the global majority vote image for t. Otherwise, show the member the personalized winning image.

- **SCB Country-level Majority Vote** The noncontextual selected image for title t is calculated as the country-level majority vote image for t across all contexts with country location c. Within the context of member m with country location c and title t, if the ratio between the expected reward of the contextual selected image to the expected reward of the noncontextual selected image is not greater than SCB input parameter δ, then show user u the country-level majority vote image for t. Otherwise, show the member the personalized winning image.
- **SCB Marketing Default** The noncontextual selected image for title t is set to the marketing default image. Within the context of user u and title t, if the ratio between the expected reward of the contextual selected image to the expected reward of the noncontextual selected image is not greater than SCB input parameter δ, then show user u the marketing default image for t. Otherwise, show the member the personalized winning image.

Results

Figure 5.2 shows a comparison of the three different policies tested for selecting the noncontextual arm (unpersonalized winning image) on the large-scale proprietary image personalization dataset. Exact details and values are omitted to protect business-sensitive intellectual property. The x coordinate is the SCB threshold value. As we move from left to right, increasing the threshold, we go from a fully personalized experience (threshold of 1) to a fully unpersonalized experienced. That is, we move from a policy that selects the contextual decision every time to one that selects the

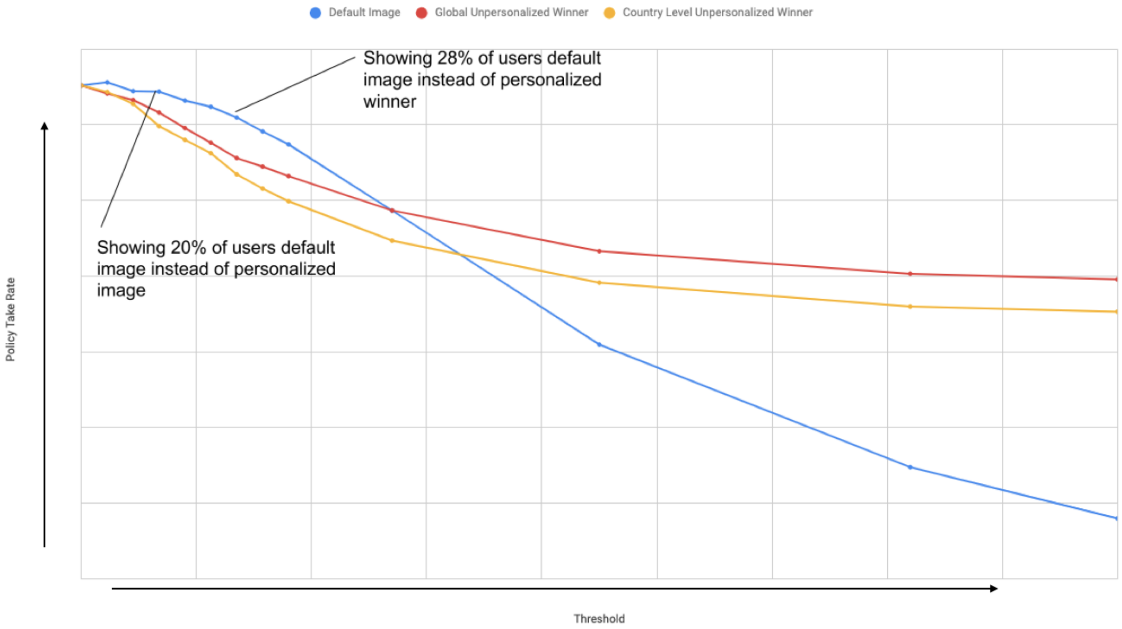


Figure 5.2: Comparison of three different policies for selecting the noncontextual arm in an industry image personalization system. A policy that fallbacks on the default marketing image during non-contextual SCB decisions performs best until a certain threshold.

noncontextual decision every time. The y coordinate is the offline policy take-rate. The higher the value the better. Each of the three lines corresponds to each of the three policies described in Section 5.5.2.

The first important observation is that with the SCB Marketing Default policy in particular we are able to increase the number of users who are impressed with the noncontextual decision (in this case, the default image) without a significant negative impact on the overall take-rate. As the figure shows, we are able to show 20% of the users the noncontextual decision, i.e. default image, instead of the contextual decision, i.e. personalized image, while at the same time achieving a take-rate that is on par with the production policy take-rate that always opts to go with the contextual decision. The second important observation is that at a threshold value that converts 10% of the contextual decisions to noncontextual ones, we observe an SCB Marketing Default policy take-rate that slightly outperforms the baseline take-rate (the fully contextual policy used in production). In this particular setting, we saw SCB perform best when using the marketing default image as the noncontextual fallback decision. Because as a we increased the SCB threshold value δ, the policy take-rate stayed relatively flat before sharply decreasing. It is worth noting that a policy that shows all users the default image does worse than a fully personalized experience and does worse than a policy that shows all users the global majority vote or country-level majority vote image. Intuitively, this aligns with original motivation to implement image personalization. We conducted the same analysis for different image asset types and across data streams from five different days and saw similar trends. Overall, our offline empirical results on this industry dataset suggest that there exists an SCB policy that reduces dependence on context that can achieve replay take-rates that are on par with or even outperform those achieved by a fully contextual policy.

5.6 Summary

In this chapter we propose a new family of multi-armed contextual bandits called selectively contextual bandits (SCB) that selectively interpolate between contextual and non-contextual treatment decisions. Using publicly available datasets corresponding to 20 different classification tasks, we have empirically demonstrated that it is possible to increase the number of non-contextual decisions from the policy while achieving similar regret metrics as a fully-contextual policy. In fact, we observe that in some cases, it is possible to even slightly outperform a fully-contextual policy. Further, we demonstrate that SCB is flexible, accommodating different explore/exploit algorithms and allowing the ability to control the amount of contextual decisions that are made using a scheduler to anneal the SCB threshold over the history of time-steps. We hypothesize that an SCB policy is beneficial in creating rich personalized treatments while also increasing the number of shared experiences across users, potentially leading to social participation through network effects.

Chapter 6

Landscape of Human-Machine Collaboration in the Fragile Families Challenge and Recommender Systems

In this chapter, we delineate the various ways in which humans and machines collaborate in the challenging real-world applications of the Fragile Families Challenge and recommender systems.

In this thesis, we cover three specific modes of human-machine collaboration in the aforementioned two contexts. However, upon doing a literature review of the full set of 17 research papers submitted to the Fragile Families Challenge and a literature review of existing recommender system research papers, we are able to draw a more complete picture of the various ways in which humans and machines collaborate in these settings.

6.1 But First, What Does Not Classify as Human-Machine Collaboration?

So what doesn't classify as human-machine collaboration? First, let's not lose sight that humans created machines and they created the algorithms that power machine learning, so everything can be called human-human collaboration. However, we do not want to get into the metaphorical philosophical weeds and we still want to develop a framework that is useful for understanding the deeply complicated landscape of machine learning and its uses in our world. Human-machine collaboration typically does not occur in the early stages of the machine learning pipeline, e.g. data collection, data preparation, data wrangling. While humans are indeed cleaning up the data, labeling data, removing duplicates, imputing missing data, etc, it does not represent a core collaboration between human and machines as we've described in this thesis. We see examples of this type of non-human-machine collaboration in [9] and [37]. In these examples, taken from the Special Collection of FFC, the authors use their intuition and volunteers to manually impute missing data and to manually select additional variables related to the ones automatically selected by machine learning models. Using a car analogy, these types of steps are similar to tuning a car, changing its oil, filling it with gas so that the car can do its basic job. Human-machine collaboration is like changing out the tires to heavy-duty terrain tires because the human has the foresight of knowing the type of rocky terrain and harsh elements the car will be heading into.

6.2 Modes of Collaboration Found in the Fragile Families Challenge

17 research papers were accepted and published to the Socius open journal Special Collection of the Fragile Families Challenge (FFC). Of these, seven (including the

paper presented in Chapter 2) demonstrated humans and machines collaborating to take on the important and challenging social science task of predicting child outcomes (as described in Section 2.1). In this section, we look at the others modes of human-machine collaboration employed in the FFC.

[84] describes the Fragile Families Challenge in depth. This paper describes how social scientists created a scientific mass collaboration to measure and understand how predictable child life trajectories are. This paper is noteworthy because it shows how a large group of humans, most of whom were machine learning technologists of varying expertise levels, came together to compete against each other to earn the top prize in a prediction problem. This shows machines and humans collaborating together at scale via a competition in order to make progress in a difficult prediction task.

Like [79], there were several research papers that demonstrated humans and machines collaborating by humans sharing their varying levels of social science domain expertise with the machines, most typically during the feature selection phase of the machine learning pipeline. [7] showed humans and machines collaborating in this way via a non-expert reading of prior social science empirical research in order to do manual variable selection. [32] used a survey to gather domain expertise. The authors surveyed a scholarly community of social scientists as well as an anonymous community of laypeople to elicit their beliefs about which variables in the FFCWS data set would best predict each of the six outcomes. The author of [64] is a social science professor; he applied his expert knowledge of the six outcomes as the main approach to selecting variables during feature selection. [73] demonstrated how a group of social scientists read 25 papers on existing research using the FFCWS and who used this knowledge to guide their manual variable selection. Finally, while [78] employed a mostly automatic feature selection process, at one point the researchers also manually added some features based on their reading of social science literature.

Overall, two different modes of human-machine collaboration were seen in the Special Collection of the FFC: (*i*) humans adding their domain knowledge during the machine learning feature selection process and (*ii*) humans creating a large-scale machine-learning competition to gain collective knowledge about the predictability of social science life outcomes.

6.3 Modes of Collaboration Found in Recommender Systems

As described at length in Chapter 3, Chapter 4, and Chapter 5, recommender or personalization systems are vital in the modern big data era for providing relevant content and products to users with limited time, attention, and screen real estate. In Chapter 3 and Chapter 4 we saw examples of humans and machines collaborating by computers providing explanations of their workings via algorithmic explanations. In Chapter 5 we saw humans and machines collaborating but working in tandem in decision making to provide a better personzalition experience for users. In this section, we looking at existing recommender system literature to highlight other ways in which humans and machines collaborate.

The cold-start problem in recommender systems describes the problem of providing a product recommendation to a new user to the system [56], [36]. It is challenging to provide relevant content to a user with zero interaction history on the platform. For this reason, we see humans and machines interacting typically explicitly during the user's first experience with a recommender platform with the machines eliciting preference feedback from the users. [75] propose asking user's initial interview questions to learn some of their preferences. [114] propose building upon this strategy by constructing a decision tree of initial interview questions so that the recommender adapts the questions based on how the user responded to the prior questions.

During the course of a user's interaction on a recommender system driven platform, there are various opportunities for human-machine collaboration in order to improve the utility of the recommender system. [113] describe a system where it continuously collects and acts upon interactive feedback it receives from the user as she engages with the recommended content. Conversational and question-based recommender systems as described in [117] and [112] are systems where humans and machines are collaborating via machines asking humans automatically constructed and algorithmically chosen questions in order to learn a user's belief and preferences. [111] go a step further by proposing a visual dialog augmented cascade model where users are recommended items with visuals ans where users give their explicit feedback by describing their desired preferences about the items using natural language. In critiquing-based systems, another form of conversational recommender systems, users give explicity feedback to recommendation in the format of "show me more like this but ..." [82]. In these type of systems, users can either accept a product recommendation or provide constructive criticism by critiquing specific attributes of the product in order to get more utile recommendations.

Overall, two different broad modes of human-machine collaboration were seen in a literature review of existing research on recommender systems: (*i*) machines querying users for feedback on active product recommendations and (*ii*) machines querying users for preference and demographic data about themselves.

6.4 Characterizing the Topology of Human-Machine Collaboration in FFC and RecSys

In this section, we characterize the attributes that describe the different modes of human-machine collaboration we have seen and discussed in this dissertation. These include:

- (A) Humans providing domain knowledge and expertise to machines
- (B) Machines providing algorithmic explanations to humans
- (C) Humans and machines working together in predictive decision making
- (D) Humans and machines working at scale via mass collaboration
- (E) Machines requesting real-time feedback on recommendations
- (F) Machines conducting human data collection during cold start problems

The attributes that characterize these different modes of human-machine collaboration are:

- **Stakeholder**: Who are the stakeholders? This could be a machine learning engineer, the end user, the platform, the content creator, domain expert
- **Collaboration Beneficiary**: The stakeholder is putting in their expertise but who is getting utility? Who benefits from the human-machine collaboration?
- **Stage of the Machine Learning Pipeline**: Where in the machine-learning pipeline does the human-machine collaboration take place? Stages include data collection, data preparation, data segregation, model training, model evaluation, model deployment, model scoring, performance monitoring.

Table 6.1: Using these attributes defined, we can characterize the various modes of human-machine collaboration by placing them in a matrix.

Collab Mode	**Stakeholder**	**Beneficiary**	**ML Stage**	*Refs*
A	engineer/expert	platform	data segregation	[79, 7, 32, 64, 73, 78]
B	engineer	engineer/end user/content creator	eval/deploy/perf	[39, 35, 100, 99]
C	engineer	end user	deploy/scoring	[109, 107, 4, 11]
D	engineer	engineer	all	[84]
E	end user	end user/platform/engineer	deploy	[113, 117, 112, 111, 82]
F	end user	end user/platform/engineer	deploy	[56, 36, 75, 114]

Chapter 7

Conclusion

To conclude, we ask ourselves, why are humans so keen to collaborate with machines in the automatic machine-learning backed processing of real-world big data? Because the ultimate goal is for the few to be able to read the minds of the many. If companies and governments can read people's minds, then these entities can accurately predict people's behavior and preferences in the future, thus reaching their ultimate capitalistic and tyrannical dreams.

So, stay random my friends. Don't let people get too comfortable thinking they have you figured out. This might be our last defense in this world with no privacy.

Bibliography

[1] The home of data science and machine learning. https://www.kaggle.com. [Online; accessed 09-October-2017].

[2] The movie database. https://www.themoviedb.org. Accessed: 2021-05-04.

[3] Research methods knowledge base. https://www.socialresearchmethods.net/kb/survey.php. [Online; accessed 09-October-2017].

[4] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. Privacy and human behavior in the age of information. *Science*, 347:509–514, 2015.

[5] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf.

[6] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135, 2013.

[7] Caitlin E. Ahearn and Jennie E. Brand. Predicting layoff among fragile families. *Socius*, 5:2378023118809757, 2019. PMID: 34553043. arXiv:https://doi.org/10.1177/2378023118809757, doi:10.1177/2378023118809757.

[8] Sajad Ahmadian, Nima Joorabloo, Mahdi Jalili, and Milad Ahmadian. Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach. *Expert Systems with Applications*, 187:115849, 2022. URL: https://www.sciencedirect.com/science/article/pii/S0957417421012100, doi:https://doi.org/10.1016/j.eswa.2021.115849.

[9] Drew M. Altschul. Leveraging multiple machine-learning techniques to predict major life outcomes from a small set of psychological and socioeconomic variables: A combined bottom-up/top-down approach. *Socius*, 5:2378023118819943, 2019. arXiv:https://doi.org/10.1177/2378023118819943, doi:10.1177/2378023118819943.

[10] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf.

[11] Naveen Awad and M. Krishnan. The personalization privacy paradox: An empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS Quarterly*, 30:13–28, 03 2006. doi:10.2307/25148715.

[12] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, August 2010.

[13] Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for deep neural network architectures. In Kuinam J. Kim and Nikolai Joukov, editors, *Information Science and Applications (ICISA) 2016*, pages 913–922, Singapore, 2016. Springer Singapore.

[14] Christophper M. Bishop. *Pattern Recognition and Machine Learning.* 2006. Chapter 3, Section 2.

[15] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statist. Sci.*, 16(3):199–231, 08 2001. URL: http://dx.doi.org/10.1214/ss/1009213726, doi:10.1214/ss/1009213726.

[16] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[17] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, July 2011. doi:10.1109/TIT.2011.2146090.

[18] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 4 2017. doi:10.1126/science.aal4230.

[19] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[20] Yibo Chen, Chanle Wu, Ming Xie, and Xiaojun Guo. Solving the sparsity problem in recommender systems using association retrieval. *JCP*, 6:1896–1902, 08 2011. doi:10.4304/jcp.6.9.1896-1902.

[21] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, page 1244–1253, New York, NY, USA, 2018. Association for Computing Machinery. `doi:10.1145/3219819.3220063`.

[22] Princeton University Library Data and Statistical Services. Interpreting regression output. `https://dss.princeton.edu/online_help/analysis/interpreting_regression.htm`. [Online; accessed 26-March-2018].

[23] Maria Dimakopoulou, Zhengyuan Zhou, Susan Athey, and Guido Imbens. Estimation considerations in contextual bandits. *arXiv preprint arXiv:1711.07077v4*, 2018.

[24] Maria Dimakopoulou, Zhengyuan Zhou, Susan Athey, and Guido Imbens. Balanced linear contextual bandits. *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[25] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*, 2017.

[26] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex J. Smola, and Vladimir Vapnik. Support vector regression machines. pages 155–161. MIT Press, 1997. URL: `http://papers.nips.cc/paper/1238-support-vector-regression-machines.pdf`.

[27] Mengnan Du, Ninghao Liu, Fan Yang, Shuiwang Ji, and Xia Hu. On attribution of recurrent neural network predictions via additive decomposition. In *The World Wide Web Conference*, WWW '19, page 383–393, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3308558.3313545`.

[28] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *CoRR*, abs/1103.4601, 2011. URL: `http://arxiv.org/abs/1103.4601`, `arXiv:1103.4601`.

[29] Jamie Duell, Xiuyi Fan, Bruce Burnett, Gert Aarts, and Shang-Ming Zhou. A comparison of explanations given by explainable artificial intelligence methods on analysing electronic health records. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4, 2021. `doi:10.1109/BHI50953.2021.9508618`.

[30] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[31] Radwa El Shawi, Youssef Sherif, Mouaz Al-Mallah, and Sherif Sakr. Interpretability in healthcare a comparative study of local machine learning interpretability techniques. In *2019 IEEE 32nd International Symposium on*

Computer-Based Medical Systems (CBMS), pages 275–280, 2019. doi:10.1109/CBMS.2019.00065.

[32] Anna Filippova, Connor Gilroy, Ridhi Kashyap, Antje Kirchner, Allison C. Morgan, Kivan Polimis, Adaner Usmani, and Tong Wang. Humans in the loop: Incorporating expert and crowd-sourced knowledge for predictions using survey data. *Socius*, 5:2378023118820157, 2019. PMID: 33981842. arXiv:https://doi.org/10.1177/2378023118820157, doi:10.1177/2378023118820157.

[33] Damien Garreau and Ulrike von Luxburg. Explaining the explainer: A first theoretical analysis of lime. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296. PMLR, 26–28 Aug 2020. URL: http://proceedings.mlr.press/v108/garreau20a.html.

[34] Rob Geada, Tommaso Teofili, Rui Vieira, Rebecca Whitworth, and Daniele Zonca. Trustyai explainability toolkit. *CoRR*, abs/2104.12717, 2021. URL: https://arxiv.org/abs/2104.12717, arXiv:2104.12717.

[35] Sofia Gkika and George Lekakos. The persuasive role of explanations in recommender systems. *CEUR Workshop Proceedings*, 1153:59–68, 01 2014.

[36] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. On bootstrapping recommender systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1805–1808, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1871437.1871734.

[37] Brian J. Goode, Debanjan Datta, and Naren Ramakrishnan. Imputing data for the fragile families challenge: Identifying similar survey questions with semiautomated methods. *Socius*, 5:2378023118822647, 2019. arXiv:https://doi.org/10.1177/2378023118822647, doi:10.1177/2378023118822647.

[38] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. doi:10.1145/2827872.

[39] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, page 241–250, New York, NY, USA, 2000. Association for Computing Machinery. doi:10.1145/358916.358995.

[40] Bernease Herman. The promise and peril of human evaluation for model interpretability. *ArXiv*, abs/1711.07414, 2017.

[41] Jianping Hua, Zixiang Xiong, James Lowey, Edward Suh, and Edward R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, April 2005. URL: http://dx.doi.org/10.1093/bioinformatics/bti171, doi:10.1093/bioinformatics/bti171.

[42] Nouhaila Idrissi and Zellou Ahmed. A systematic literature review of sparsity issues in recommender systems. *Social Network Analysis and Mining*, 10, 12 2020. doi:10.1007/s13278-020-0626-2.

[43] Sérgio Jesus, Catarina Belém, Vladimir Balayan, João Bento, Pedro Saleiro, Pedro Bizarro, and João Gama. How can i choose an explainer? an application-grounded evaluation of post-hoc explanations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 805–815, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3442188.3445941.

[44] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4211–4222. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper/2020/file/2c29d89cc56cdb191c60db2f0bae796b-Paper.pdf.

[45] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3819–3828, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2702123.2702520.

[46] Kate Kelley, Belinda Clark, Vivienne Brown, and John Sitzia. Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, 15(3):261–266, 2003. URL: + http://dx.doi.org/10.1093/intqhc/mzg031, arXiv:/oup/backfile/content_public/journal/intqhc/15/3/10.1093/intqhc/mzg031/2/mzg031.pdf, doi:10.1093/intqhc/mzg031.

[47] G.D. Kuh, J.L. Kinzie, J.A. Buckley, B.K. Bridges, and J.C. Hayek. *What Matters to Student Success: A Review of the Literature*. National Postsecondary Education Cooperative, 2006. URL: https://books.google.com/books?id=wfJprgEACAAJ.

[48] Akshi Kumar and Abhilasha Sharma. Alleviating sparsity and scalability issues in collaborative filtering based recommender systems. In Suresh Chandra Satapathy, Siba K. Udgata, and Bhabendra Narayan Biswal, editors, *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, pages 103–112, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[49] Vivian Lai, Jon Z. Cai, and Chenhao Tan. Many Faces of Feature Importance: Comparing Built-in and Post-hoc Feature Importance in Text Classification. *arXiv e-prints*, page arXiv:1910.08534, October 2019. arXiv:1910.08534.

[50] Himabindu Lakkaraju and Osbert Bastani. "how do i fool you?": Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 79–85, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3375627.3375833.

[51] Huitian Lei, Ambuj Tewari, and Susan A Murphy. An actor-critic contextual bandit algorithm for personalized mobile health interventions. *arXiv preprint arXiv:1706.09090*, 2017.

[52] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, USA, 2nd edition, 2014.

[53] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[54] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 297–306, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1935826.1935878, doi:10.1145/1935826.1935878.

[55] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2071–2080. JMLR. org, 2017.

[56] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. Seamlessly unifying attributes and items: Conversational recommendation for cold-start users. *ACM Trans. Inf. Syst.*, 39(4), aug 2021. doi:10.1145/3446427.

[57] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 689–698, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. doi:10.1145/3178876.3186150.

[58] Zhong Qiu Lin, Mohammad Javad Shafiee, Stanislav Bochkarev, Michael St. Jules, Xiao Yu Wang, and Alexander Wong. Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. *ArXiv*, abs/1910.07387, 2019.

[59] Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, sep 2018. doi:10.1145/3233231.

[60] Matthew Lombard, Jennifer Snyder-Duch, and Cheryl Campanella Bracken. Content analysis in mass communicationassessment and reporting of intercoder reliability. *Human Communication Research*, 28(4):587–604, 2002. URL: http://dx.doi.org/10.1111/j.1468-2958.2002.tb00826.x, arXiv:/oup/backfile/content_public/journal/hcr/28/4/10.1111_j.1468-2958.2002.tb00826.x/2/jhumcom0587.pdf, doi:10.1111/j.1468-2958.2002.tb00826.x.

[61] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[62] Xin Man and Ernest P. Chan. The best way to select features? comparing mda, lime, and shap. 2020.

[63] Daniel A. McFarland, Kevin Lewis, and Amir Goldberg. Sociology in the era of big data: The ascent of forensic social science. *The American Sociologist*, 47(1):12–35, Mar 2016. doi:10.1007/s12108-015-9291-8.

[64] Stephen McKay. When 4 10,000: The power of social science knowledge in predictive performance. *Socius*, 5:2378023118811774, 2019. arXiv:https://doi.org/10.1177/2378023118811774, doi:10.1177/2378023118811774.

[65] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. Auditing search engines for differential satisfaction across demographics. In *Proceedings of the 26th International Conference on World Wide Web (Industry Track)*, April 2017. URL: https://www.microsoft.com/en-us/research/publication/auditing-search-engines-for-differential-satisfaction-across-demographics/.

[66] Andreas Messalas, Christos Makris, and Yannis Kanellopoulos. Model-agnostic interpretability with shapley values. 07 2019. doi:10.1109/IISA.2019.8900669.

[67] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[68] Christoph Molnar. *Limitations of Interpretable Machine Learning Methods*. 2020. https://github.com/compstat-lmu/iml_methods_limitations.

[69] Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, April 2021. URL: https://www.microsoft.com/en-us/research/publication/towards-unifying-feature-attribution-and-counterfactual-explanations-different-means-to-the-same-end/.

[70] Caio Nóbrega and Leandro Marinho. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 1671–1678, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3297280.3297443`.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[72] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[73] Louis Raes. Predicting gpa at age 15 in the fragile families and child wellbeing study. *Socius*, 5:2378023118824803, 2019. `arXiv:https://doi.org/10.1177/2378023118824803`, `doi:10.1177/2378023118824803`.

[74] Yanou Ramon, David Martens, Foster J. Provost, and Theodoros Evgeniou. Counterfactual explanation algorithms for behavioral and textual data. *CoRR*, abs/1912.01819, 2019. URL: `http://arxiv.org/abs/1912.01819`, `arXiv:1912.01819`.

[75] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, and John Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, page 127–134, New York, NY, USA, 2002. Association for Computing Machinery. `doi:10.1145/502716.502737`.

[76] Nancy E. Reichman, Julien O. Teitler, Irwin Garfinkel, and Sara S. McLanahan. Fragile families: sample and design. *Children and Youth Services Review*, 23(4):303 – 326, 2001. `doi:https://doi.org/10.1016/S0190-7409(01)00141-4`.

[77] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. `doi:10.1145/2939672.2939778`.

[78] Daniel E. Rigobon, Eaman Jahani, Yoshihiko Suhara, Khaled AlGhoneim, Abdulaziz Alghunaim, Alex "Sandy" Pentland, and Abdullah Almaatouq. Winning models for grade point average, grit, and layoff in the fragile families challenge. *Socius*, 5:2378023118820418, 2019. `arXiv:https://doi.org/10.1177/2378023118820418`, `doi:10.1177/2378023118820418`.

[79] Claudia V. Roberts. Friend request pending: A comparative assessment of engineering- and social science–inspired approaches to analyzing complex birth

cohort survey data. *Socius*, 5:2378023118820431, 2019. arXiv:https://doi.org/10.1177/2378023118820431, doi:10.1177/2378023118820431.

[80] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

[81] Ashish Sahu and Pragya Dwivedi. User profile as a bridge in cross-domain recommender systems. *Applied Intelligence*, 01 2019.

[82] Yasser Salem, Jun Hong, and Weiru Liu. History-guided conversational recommendation. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, page 999–1004, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2567948.2578844.

[83] Matthew Salganik, Ian Lundberg, Alex Kindel, and McLanahan Sara. Introduction to the special issue on the fragile families challenge.

[84] Matthew J. Salganik, Ian Lundberg, Alexander T. Kindel, and Sara McLanahan. Introduction to the special collection on the fragile families challenge. *Socius*, 5:2378023119871580, 2019. arXiv:https://doi.org/10.1177/2378023119871580, doi:10.1177/2378023119871580.

[85] Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/3ab6be46e1d6b21d59a3c3a0b9d0f6ef-Paper.pdf.

[86] Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi:10.7249/P0295.

[87] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org, 2017.

[88] Jacob Sippy, Gagan Bansal, and Daniel S. Weld. Data staining: A method for comparing faithfulness of explainers. 2020.

[89] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)*, 2020. URL: https://arxiv.org/pdf/1911.02508.pdf.

[90] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, WWW '19, page 3251–3257, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3308558.3313710.

[91] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[92] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[93] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 303–310, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3278721.3278725.

[94] Maartje ter Hoeve, Anne Schuth, Daan Odijk, and M. de Rijke. Faithfully explaining rankings in a news recommender system. *ArXiv*, abs/1805.05447, 2018.

[95] The Fragile Families and Child Wellbeing Study. Fragile families and child wellbeing study. http://www.fragilefamilies.princeton.edu/about.

[96] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[97] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[98] A. N. Tikhonov. On the solution of incorrectly formulated problems and the regularization method. *Dokl Akad Nauk SSSR*, 151:501–504, 1963.

[99] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In G. Uchyigit, editor, *Data Engineering Workshop*, pages 801–810. IEEE Computer Society, December 2007. IEEE 23rd International Conference on Data Engineering (ICDE 2007) ; Conference date: 16-04-2007 Through 20-04-2007. doi:10.1109/ICDEW.2007.4401070.

[100] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction*, 22, 10 2012. doi:10.1007/s11257-011-9117-5.

[101] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. URL: http://doi.acm.org/10.1145/2641190.2641198, doi:10.1145/2641190.2641198.

[102] Mythreyi Velmurugan, Chun Ouyang, Catarina Moreira, and Renuka Sindhgatta. Evaluating Explainable Methods for Predictive Process Analytics: A Functionally-Grounded Approach. *arXiv e-prints*, page arXiv:2012.04218, December 2020. arXiv:2012.04218.

[103] Jesse Vig, Shilad Sen, and John Riedl. The tag genome: Encoding community knowledge to support novel interaction. *ACM Trans. Interact. Intell. Syst.*, 2(3), September 2012. doi:10.1145/2362394.2362395.

[104] Giorgio Visani, Enrico Bagli, Federico Chesani, Alessandro Poluzzi, and Davide Capuzzo. Statistical stability indices for LIME: obtaining reliable explanations for machine learning models. *CoRR*, abs/2001.11757, 2020. URL: https://arxiv.org/abs/2001.11757, arXiv:2001.11757.

[105] P.S. Visser, Jon A. Krosnick, and Paul J. Lavraka. Survey research. *Handbook of research methods in social and personality psychology*, pages 223–252, 2000. [Online; accessed 09-October-2017].

[106] Domen Vres and Marko Robnik-Sikonja. Better sampling in explanation methods can prevent dieselgate-like deception. *CoRR*, abs/2101.11702, 2021. URL: https://arxiv.org/abs/2101.11702, arXiv:2101.11702.

[107] Tiffany Barnett White, Debra L. Zahay, Helge Thorbjornsen, and Sharon Shavitt. Getting too personal: Reactance to highly personalized email solicitations. *Marketing Letters*, 19(1):39–50, 2008. URL: http://www.jstor.org/stable/41217894.

[108] Daniel Wright. Making friends with your data: Improving how statistics are conducted and reported. *The British journal of educational psychology*, 73:123–36, 04 2003.

[109] Heng Xu, Robert Luo, John Carroll, and Mary Beth Rosson. The personalization privacy paradox: An exploratory study of decision making process for location-aware marketing. *Decision Support Systems*, 51:42–52, 04 2011. doi:10.1016/j.dss.2010.11.017.

[110] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2921–2930. Curran Associates, Inc., 2017. URL: http://papers.nips.cc/paper/6885-beyond-parity-fairness-objectives-for-collaborative-filtering.pdf.

[111] Tong Yu, Yilin Shen, and Hongxia Jin. A visual dialog augmented interactive recommender system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 157–165, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3292500.3330991.

[112] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and*

Knowledge Management, CIKM '18, page 177–186, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3269206.3271776.

[113] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information Knowledge Management*, CIKM '13, page 1411–1420, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2505515.2505690.

[114] Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 315–324, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2009916.2009961.

[115] Zhengze Zhou, Giles Hooker, and Fei Wang. *S-LIME: Stabilized-LIME for Model Explanation*, page 2429–2438. Association for Computing Machinery, New York, NY, USA, 2021. URL: https://doi.org/10.1145/3447548.3467274.

[116] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[117] Jie Zou, Yifan Chen, and Evangelos Kanoulas. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 881–890, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3397271.3401180.

[118] Rebecca Zwick and Jennifer Greif Green. New perspectives on the correlation of sat scores, high school grades, and socioeconomic factors. *Journal of Educational Measurement*, 44(1):23–45, 2007. URL: http://www.jstor.org/stable/20461841.