

AdaFair: Cumulative Fairness Adaptive Boosting

V. Ioifidis¹ E. Ntoutsi¹

¹University of Leibniz
L3S Research Center
Hannover, Germany

November 11, 2019

Discrimination

- Discrimination is treatment or consideration of, or making a distinction towards, a person based on a protected attribute to which the person is perceived to belong.
- Protected attributes are considered to be: age, disability, race, religion, sex, sexual orientation, etc.



Image source: <https://www.hr magazine.co.uk/article-details/discrimination-costs-the-uk-127-billion-a-year>

Recent discrimination discoveries in machine learning applications

Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments

Alexandra Chouldechova*

DE GRUYTER OPEN

Proceedings on Privacy Enhancing Technologies 2018, 1(1):10–112

Amil Datta*, Michael Carl Tschantz, and Anupam Datta

Automated Experiments on Ad Privacy Settings

A Tale of Opacity, Choice, and Discrimination

101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi¹, Kai-Wei Chang², James Zou³, Venkatesh Saligrama⁴, Adam Kalai⁵

¹Amazon University, 8 Saint Mary's Street, Boston, MA
²Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
³tschantz@cs.toronto.edu, kai@cs.toronto.edu, jameszou@cs.toronto.edu, venkatesh.saligrama@toronto.edu

Abstract

The blind application of machine learning runs the risk of amplifying biases present in data. Such a danger is facing us with word embeddings, a popular framework to represent text data as vectors which has been used in many machine learning and natural language processing tasks. We show that even word embeddings trained on Google News articles exhibit *female/male gender stereotypes* to a disturbing extent. This causes concerns because their widespread use, as we discuss, often tends to amplify these biases. Concretely, gender bias in first shown to be captured by directions in the word embedding. Second, gender neutral words are shown to be linearly separable from gender disambiguation words in the word embedding. Using these properties, we provide a methodology for modifying an embedding to remove gender stereotypes, such as the association between the words *receptionist* and *female*, while maintaining desired associations such as between the words *queen* and *female*. Using crowd-worker evaluation as well as standard benchmarks, we empirically demonstrate that our algorithm significantly reduces gender bias in embeddings while preserving the useful properties such as the ability to cluster related concepts and to solve analogy tasks. The resulting embeddings can be used in applications without amplifying gender bias.

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

1 Introduction

Research on word embeddings has shown significant interest in machine learning and natural language processing. There have been hundreds of papers written about word embeddings and their applications, from Web search [22] to parsing [Carreras and Hays, 11]. However, none of these papers have recognized how historically sexist the embeddings are and hence risk introducing biases of various types into real-world systems.

A word embedding, trained on word co-occurrence in text corpora, represents each word (or common phrase) as a d-dimensional word vector $w \in \mathbb{R}^d$. It serves as a dictionary of sorts for computer programs that would like to use word meaning. First, words with similar semantic meanings tend to have vectors that are close together. Second, the vector differences between words in embeddings have been shown to represent relationships between words [27, 21]. For example given an analogy puzzle, “man is to king as woman is to ?” (denoted as *man:king :: woman:?*), simple arithmetic of the embedding vectors finds that *queen* is the best answer because *king - man ≈ king - queen*. Similarly, *man:king ≈ queen:prince* is returned for *Paris:France :: Tokyo:?*. It is surprising that a simple vector arithmetic can simultaneously capture a variety of relationships. It has also existed practitioners because such a text could be used for search applications involving natural language. Indeed, they are being studied and used in a variety of downstream applications (e.g., document retrieval [23], sentiment analysis [14], and question retrieval [17]).

However, the embeddings also present sexism implicit in text. For instance, it is also the case that *girl - woman ≈ cousin - grandmother*.

Our new work, the same system that solved the above reasonable analogies will offensively answer “man is to computer programmer as woman is to a” with “homemaker”. Similarly, it will output that a

NIPS Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Discrimination in Online Ad Delivery

Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness

Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*

Joy Buolamwini

MIT Media Lab 75 Amherst St. Cambridge, MA 02139

joyab@mit.edu

Abstract
The blind application of machine learning runs the risk of amplifying biases present in data. Such a danger is facing us with word embeddings, a popular framework to represent text data as vectors which has been used in many machine learning and natural language processing tasks. We show that even word embeddings trained on Google News articles exhibit female/male gender stereotypes to a disturbing extent. This causes concerns because their widespread use, as we discuss, often tends to amplify these biases. Concretely, gender bias in first shown to be captured by directions in the word embedding. Second, gender neutral words are shown to be linearly separable from gender disambiguation words in the word embedding. Using these properties, we provide a methodology for modifying an embedding to remove gender stereotypes, such as the association between the words receptionist and female, while maintaining desired associations such as between the words queen and female. Using crowd-worker evaluation as well as standard benchmarks, we empirically demonstrate that our algorithm significantly reduces gender bias in embeddings while preserving the useful properties such as the ability to cluster related concepts and to solve analogy tasks. The resulting embeddings can be used in applications without amplifying gender bias.



Gender differences and bias in open source: pull request acceptance of women versus men

Josh Tenenbaum¹, Andrew Kofink², Justin Middleton³, Clarissa Rainer⁴, Emerson Murphy-Hill⁵, Chris Parnis⁶ and Jon Stallings⁷

¹Department of Computer Science, California Polytechnic State University—San Luis Obispo, San Luis Obispo, CA, United States
²Department of Computer Science, North Carolina State University, Raleigh, NC, United States
³Department of Statistics, North Carolina State University, Raleigh, NC, United States

ABSTRACT

Biases against women in the workplace have been documented in a variety of studies. This paper presents a large scale study on gender bias, where we compare acceptance rates of contributions from men versus women in an open source software community. Surprisingly, our results show that women's contributions tend to be accepted more often than men's. However, for contributors who are outside to a project and their gender is identifiable, men's acceptance rates are higher. Our results suggest that although women on GitHub may be more competent overall, bias against them exists nonetheless.

Subjects: Human-Computer Interaction, Social Computing, Programming Languages, Software Engineering
Keywords: Gender Bias, Open source, Software development, Software engineering

INTRODUCTION

In 2011, a software developer named Rachel Nabors wrote about her experiences trying to fix bugs in open source software [http://www.rachelnabors.com/2011/04/08/github-and-pull-requests-and-overall/]. Nabors was surprised that all of her contributions were rejected by the project owners. A reader suggested that she was being discriminated against because of her gender.

Research suggests that, indeed, gender bias pervades open source. In Nafar's interviews with women in open source, she found that “sexist behavior is, as constant as it is extreme” [Nafar, 2011]. In Vasilakis and colleagues' study of Stack Overflow, a question and answer community for programmers, they found “a relatively ‘unfriendly’ community where women disagree women, although their activity levels are comparable to men's” [Vasilakis, Capiluppi & Serebrenik, 2016]. These studies are especially troubling in light of research which suggests that diverse software development teams are more productive than homogeneous teams [Vasilakis et al., 2015]. Nevertheless, in a 2011 survey of the more than 2000 open source developers who indicated a gender, only 11.2% were women [Aryna-Datta, Rubin & Datta, 2014].

Submitted 18 August 2016

Assigned 18 March 2017

Published 1 May 2017

Corresponding author

Emerson Murphy-Hill

emurphy@cs.unc.edu

Academic editor

Andrew Kofink

Additional Information and

Declarations can be found on

page 27

DOI 10.7717/peerj-1111

© Copyright

2017 Tenenbaum et al.

Distributed under

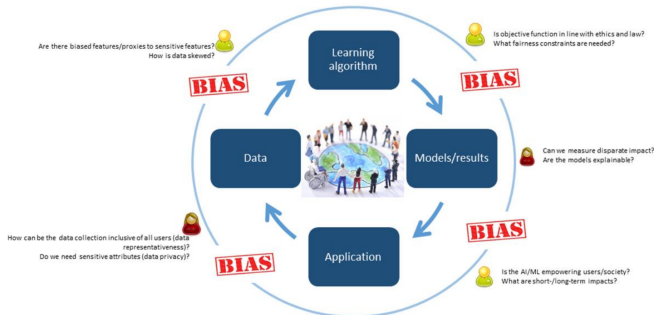
Creative Commons CC-BY 4.0

OPEN ACCESS



Why Machine Learning is Unfair?

- Data might encode existing biases e.g., bias caused by humans, features of minorities contain more noise.
- Data collection feedback loops.
- Different data distributions for different groups e.g., lack of observed examples.
- Proxies to protected attributes e.g., marital status *wife* or *husband* can reveal the gender.



The "trap" of *Equalized Odds*

Example

- Positive class \ll Negative class e.g.,
 $|s^+| + |\bar{s}^+| = 5\%$, $|s^-| + |\bar{s}^-| = 95\%$
- Model classifies everything as negative.
- Accuracy is still high (95%) and model is "fair" i.e.,
 $\delta FNR = 0, \delta FPR = 0$

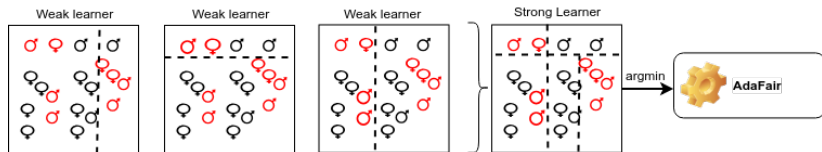
Goal

Find a mapping function $f(\cdot)$ that minimizes Eq.Odds while performing well for both classes.

$$BER = 1 - \frac{1}{2} \cdot \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) = 1 - \frac{1}{2} \cdot (TPR + TNR)$$

AdaFair: Overview

- Fairness-aware boosting approach that deals with class-imbalance and unfair outcomes.
- Changes data distribution at each round based on the notion of cumulative fairness.
- After the training phase, the best sequence of weak learners which achieve high performance and fairness is selected.



Cumulative Fairness

- Let $j : 1 \dots T$ be the current boosting round, T is user defined
- Let $H_{1:j} = \sum_{i=1}^j a_i h_i(x)$ be the ensemble model up to current round j .
- The cumulative fairness of the ensemble up to round j , is defined based on the parity in the predictions of weak learners $h_1() \dots h_j()$ between protected and non-protected groups

Cumulative Fairness

$$\delta FNR^{1:j} = \frac{\sum_{i=1}^{|\bar{s}_+|} 1 \cdot \mathbb{I}[\sum_{k=1}^j a_k h_k(x_i^{\bar{s}_+}) \neq y_i]}{|\bar{s}_+|} - \frac{\sum_{i=1}^{|s_+|} 1 \cdot \mathbb{I}[\sum_{k=1}^j a_k h_k(x_i^{s_+}) \neq y_i]}{|s_+|}$$
$$\delta FPR^{1:j} = \frac{\sum_{i=1}^{|\bar{s}_-|} 1 \cdot \mathbb{I}[\sum_{k=1}^j a_k h_k(x_i^{\bar{s}_-}) \neq y_i]}{|\bar{s}_-|} - \frac{\sum_{i=1}^{|s_-|} 1 \cdot \mathbb{I}[\sum_{k=1}^j a_k h_k(x_i^{s_-}) \neq y_i]}{|s_-|}$$

Fairness weights

- Vanilla AdaBoost already boosts misclassified instances for the next round.
- Our weighting explicitly targets fairness by extra boosting discriminated groups for the next round.
- Instances $x_i \in D$ which belong to a group that is discriminated receive a fairness-related weight u_i

Weight calculation

$$u_i = \begin{cases} |\delta FNR^{1:j}|, & \text{if } \mathbb{I}((y_i \neq h_j(x_i)) \wedge |\delta FNR^{1:j}| > \epsilon), x_i \in s_+, \delta FNR^{1:j} > 0 \\ |\delta FNR^{1:j}|, & \text{if } \mathbb{I}((y_i \neq h_j(x_i)) \wedge |\delta FNR^{1:j}| > \epsilon), x_i \in \bar{s}_+, \delta FNR^{1:j} < 0 \\ |\delta FPR^{1:j}|, & \text{if } \mathbb{I}((y_i \neq h_j(x_i)) \wedge |\delta FPR^{1:j}| > \epsilon), x_i \in s_-, \delta FPR^{1:j} > 0 \\ |\delta FPR^{1:j}|, & \text{if } \mathbb{I}((y_i \neq h_j(x_i)) \wedge |\delta FPR^{1:j}| > \epsilon), x_i \in \bar{s}_-, \delta FPR^{1:j} < 0 \\ 0, & \text{otherwise} \end{cases}$$

AdaFair's pseudocode

Input: $D = (x_i, y_i)_1^N, T, \epsilon$

Output: Ensemble H

- 1 Initialize $w_i = 1/N$ and $u_i = 0$, for $i = 1, 2, \dots, N$
- 2 For $j = 1$ to T :
 - 1 Train a classifier h_j to the training data using weights w_i .
 - 2 Compute the error rate $\text{err}_j = \frac{\sum_{i=1}^N w_i I(y_i \neq h_j(x_i))}{\sum_{i=1}^N w_i}$
 - 3 Compute the weight $\alpha_j = \frac{1}{2} \cdot \ln\left(\frac{1 - \text{err}_j}{\text{err}_j}\right)$
 - 4 Compute fairness-related $\delta FNR^{1:j}$
 - 5 Compute fairness-related $\delta FPR^{1:j}$
 - 6 Compute fairness-related weights u_i
 - 7 Update the distribution as
$$w_i \leftarrow \frac{1}{Z_j} w_i \cdot e^{\alpha_j \cdot \hat{h}_j(x) \cdot \mathbb{I}(y_i \neq h_j(x_i))} \cdot (1 + u_i)$$
- 3 Output $H(x) = \sum_{j=1}^T \alpha_j h_j(x)$

Performance trade-off: error vs balanced error

- AdaFair optimizes for the balanced error rate.
- AdaFair selects the optimal number of weak learners $1 \dots \theta, \theta \leq T$ that minimizes BER.
- AdaFair considers both ER and BER in the objective function as follows:

$$\arg \min_{\theta} (c \cdot BER_{\theta} + (1 - c) \cdot ER_{\theta} + Eq.Odds_{\theta})$$

- Parameter c is user-defined and controls the impact of error and balanced error rate.

Baselines

- AdaBoost [Sch99]: vanilla AdaBoost.
- SMOTEBoost [CLHB03]: AdaBoost with SMOTE for imbalanced data.
- Krasanakis et al. [KXPK18]: Boosting method which minimizes Equalised Odds by approximating the underlying distribution of hidden correct labels.
- Zafar et al. [ZVGRG17]: Training logistic regression model with convex-concave constraints to minimize Equalised Odds.
- AdaFair NoCumul: Variation of AdaFair that computes the fairness weights based on individual weak learners.

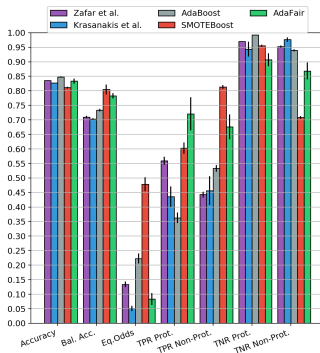
Datasets

	Adult Census	Bank	Compass	KDD Census
#Instances	45,175	40,004	5,278	299,285
#Attributes	14	16	9	41
Sen.Attr.	Gender	Marit. Status	Gender	Gender
Class ratio (+:−)	1:3.03	1:7.57	1:1.12	1:15.11
Positive class	>50K	<i>subscription</i>	<i>recidivism</i>	>50K

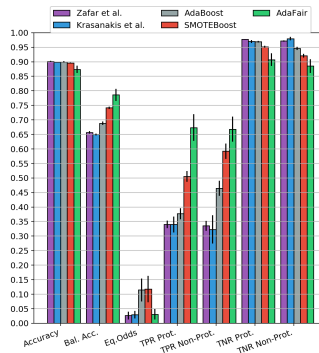
Employed datasets

We report on the average of 10 random splits [ZVG RG17], 50% training and 50% testing set.

AdaFair vs Baselines



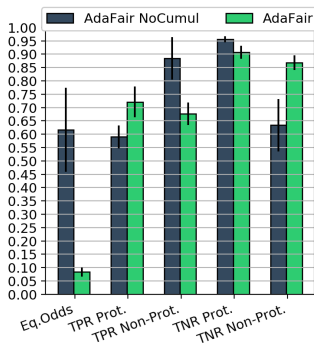
Adult Census



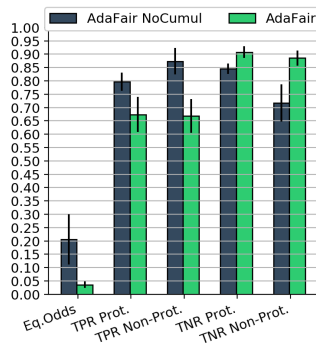
Bank

- AdaBoost and SMOTEBoost do not consider fairness (high Eq.Odds).
- Krasanakis et al. and Zafar et al. produce low TPRs and high TNRs.

Cumulative vs Non Cumulative Overall Performance



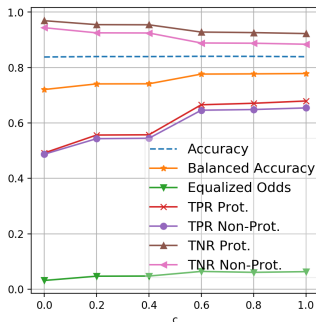
Adult Census



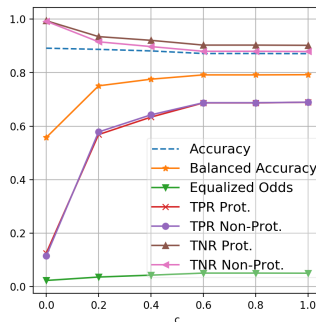
Bank

- AdaFair NoCumul has poor fairness performance.
- AdaFair NoCumul is very unstable.

Impact of parameter c



Adult Census



Bank

- For $c = 0$, the error rate is optimized and $c = 1$ the balanced error rate.

Conclusions

Conclusions

- AdaFair: fairness-aware boosting approach.
 - Data distributions alter based on *cumulative fairness*.
 - Deal with class-imbalance (indirectly).
- Substantial difference in performance compared to baselines.
- Cumulative fairness is superior to a non-cumulative approach.

Future Work

- Embed class-imbalance learning into training phase.
- Investigate theoretical properties e.g., convergence



Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer.

Smoteboost: Improving prediction of the minority class in boosting.
In *ECML PKDD*, 2003.



Moritz Hardt, Eric Price, Nati Srebro, et al.

Equality of opportunity in supervised learning.
In *NeurIPS*, 2016.



Emmanouil Kerasanakis, Eleftherios Spyromitros Xioufis, Symeon Papadopoulos, and Yiannis Kompatsiaris.

Adaptive sensitive reweighting to mitigate bias in fairness-aware classification.
In *WWW*, 2018.



Robert E Schapire.

A brief introduction to boosting.
In *IJCAI*, 1999.



Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi.

Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment.

Thanks.

Questions?

Contact: {iosifidis, ntoutsi}@L3S.de