

MACHINE LEARNING FAIRNESS WITH LIMITED PROTECTED ATTRIBUTES

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to US Provisional Application No. 63/507,381 filed on June 9, 2023, the contents of which are incorporated by reference in their entirety for all purposes.

BACKGROUND

[0002] Machine learning models rely on using a large amount of data to provide predictions. Generally, it is believed that the provided data is a complete or near-complete dataset with values for various datapoints within the set to be known. The availability of this information allows for machine learning models to make predictions. Further, a robust dataset allow for analysis of the machine learning models to understand if a trained model is biased against certain values within the datapoints. For instance, a choice of machine learning model or technique may lead to bias against a certain population within the dataset while a different choice may lead to bias against a different population. Once a model or technique is trained, it is possible to evaluate based on the dataset what biases may exist. However, this approach assumes that all information is available about the underlying dataset on which training occurred.

[0003] Further, the growing ubiquity of machine learning models in various aspects of decision making has brought fairness in machine learning in focus. For instance, various decisions, such as credit availability, loan approvals, hiring, and even criminal sentencing can rely on machine learning. Fairness in machine learning can refer to the process of eliminating or correcting bias which may exist in a machine learning model to prevent or correct outcomes which may exist from any characteristic related to the data. For example, there may be an emergent or inherent bias which can be related to race, ethnicity, gender, sexual orientation, disability, veteran status, or other characteristic which may be related to the data. Thus, ensuring fairness in machine learning models is essential for promoting the positive impacts of

machine learning while mitigating the negative impacts of machine learning on individuals, entities, and society.

[0004] Additionally, one of the primary challenges in achieving and determining whether a machine learning model is fair is the limited availability of certain values or protected attributes (PAs), such as for example gender, race, or age, due to privacy, consent, or legal barriers. As this information is scarce, it can hinder in the development of fair machine learning algorithms as it becomes difficult to control and mitigate bias without the necessary data on the protected attributes. Further, many countries and regulations prohibit the gathering of such data.

[0005] Despite obstacles regarding obtaining PAs, existing solutions to mitigate for bias and ensure fairness assume the availability of PAs and incorporate them into a machine learning model to avoid bias. Thus, a need for a new framework to achieve fairness in resource-constrained situations, such as when limited or unknown information about PAs is available, is needed.

BRIEF SUMMARY

[0006] Embodiments disclosed include an upstream model that can learn predictive models for protected attributes (PAs) using limited data and then synthesize new PAs for use in downstream target tasks. The estimated PAs can effectively replace original PAs, enabling the development of fair machine learning models that operate on limited access to PAs. Further, selecting high-quality examples and processing them with fairness in mind can help train competitive fair models from relatively small data sets.

[0007] Provided herein are systems, methods, and computer readable media that can achieve fairness in machine learning applications while using limited protected attributes. In some examples, a two-stage (a first stage (also referred to as an upstream stage, PA estimation stage, or bias estimation stage) and a second stage (also referred to as a downstream stage, fair model generation stage, or downstream fairness stage)) framework can leverage active learning techniques to predict protected attributes. The framework can utilize these predictions, which can be based on limited labeled data, into a machine learning training process to achieve fairness in a target task. In some examples, the first stage (upstream) can employ active learning to accurately predict protected attributes using limited labeled data. Further, in some examples, the second stage (downstream) can incrementally

sample low-biased training examples for fair learning based on estimates provided from first stage. The framework can increase effectiveness in both predictive performance on a target task and in achieving fairness. In some examples, sample selection strategies can be used to sample data from the available set of data. Sample selection strategies can further increase fairness while only accessing a limited number of PAs.

[0008] Embodiments disclosed include a method for correcting algorithmic bias in a machine learning framework. The method may comprise obtaining, a first dataset, the first dataset containing protected attributes from at least one category of protected attributes; obtaining, a second dataset, the second dataset not containing any protected attributes from the at least one category of protected attributes; training, using the first dataset, a protected attributes estimation model, wherein the protected attributes estimation model estimates information related to the least one protected attribute; and predicting, using protected attributes estimation model, at least one protected attribute for data within the second dataset. The method may further include any combination of the following aspects. The protected attributes estimation model is trained in iterations. Each iteration may involve an active learning process to improve the protected attributes estimation model at each iteration. Additional datapoints containing protected attributes may be added to the first dataset to update the training of the protected attributes. Additional datapoints may be obtained by annotating points from the second dataset with a protected attribute. Additional datapoints may be chosen from a ranked list of datapoints in the second dataset. Ranking may be based on empirically defined bias. The protected attributes estimation model may provides a confidence metric associated with the predicted protected attribute. An augmented dataset may be generated based on augmenting the second dataset with predicted protected attributes from the protected attributes estimation model.

[0009] Embodiments disclosed include a method comprising obtaining, an augmented dataset, the augmented dataset comprising datapoints with protected attributes predicted from an estimation model; training a plurality of bias mitigation models based on subsets of the augmented dataset; obtaining a fairness metric and an accuracy metric for each of the plurality of the bias mitigation models; and choosing one model from the plurality of bias mitigation models based on at least one of the fairness metric and the accuracy metric. The subsets of the augmented dataset may be generated based on statistical information of the protected attributes of the augmented dataset. The statistical information may be generated by

the estimation model. The statistical information may include a confidence metric associated with a predicted protected attributed. The subsets of the augmented dataset may be generated based on ranges of confidence metrics. The subsets of the augmented dataset may be generated based on statistical information of the protected attributes of the augmented dataset.

[0010] Embodiments disclosed include a method with a first stage and a second stage. The first stage may comprise obtaining, a protected attributes (PA) known dataset and a PA unknown dataset; training a bias estimation model based on datapoints from the PA known dataset; generating, estimates for protected attributes based on the trained bias estimation model; augmenting the PA unknown dataset to generate an augmented dataset. The second stage may comprise obtaining the augmented dataset; generating a plurality of subsets of the augmented dataset; training a plurality of bias mitigation models based on the plurality of subsets of the augmented dataset; and choosing one of the trained plurality of bias mitigation models as a final mitigation model. Additional datapoints may be added to the PA known dataset through an active learning process. The active learning process may be performed by annotating datapoints in the PA unknown dataset with protected attributes. A ranked entropy model may be used to select additional datapoints during training in the second stage.

[0011] Embodiments may include a system comprising one or more processor coupled to a computer readable medium, the computer readable medium comprising instructions, that when executed by the one or more processors perform any of the preceding methods.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1A illustrates an example dataset with protected attributes.

[0013] FIG. 1B illustrates a two stage framework to obtain fairness in machine learning applications.

[0014] FIG. 2 illustrates an example framework to generate a protected attributes estimation model.

[0015] FIG. 3 illustrates an example framework to generate and select one of a plurality of fair models.

[0016] FIG. 4 illustrates an example method related to a protected attributes estimation model.

[0017] FIG. 5 illustrates an example method related to generating fair models.

[0018] FIGS. 6, 7, and 8 illustrate performance and fairness aspects related to machine learning models.

[0019] FIG. 9 shows an example computing system according to an embodiment of the present disclosure.

TERMS

[0020] A “server computer” may include a powerful computer or cluster of computers. For example, the server computer can include a large mainframe, a minicomputer cluster, or a group of servers functioning as a unit. In one example, a server computer can include a database server coupled to a web server. The server computer may comprise one or more computational apparatuses and may use any of a variety of computing structures, arrangements, and compilations for servicing the requests for one or more “client computers.”

[0021] A “memory” may include any suitable device or devices that may store electronic data. A suitable memory may comprise a non-transitory computer readable medium that stores instructions that can be executed by a processor to implement a desired method. Examples of memories include one or more memory chips, disk drives, etc. Such memories may operate using any suitable electrical, optical, and/or magnetic mode of operation.

[0022] A “processor” may include any suitable data computation device or devices. A processor may comprise one or more microprocessors working together to accomplish a desired function. The processor may include a CPU that comprises at least one high-speed data processor adequate to execute program components for executing user and/or system generated requests. The CPU may be a microprocessor such as AMD’s Athlon, Duron and/or Opteron; IBM and/or Motorola’s PowerPC; IBM’s and Sony’s Cell processor; Intel’s Celeron, Itanium, Pentium, Xenon, and/or Xscale; and/or the like processor(s).

[0023] A “uniform distribution” may refer to a probability distribution (e.g., probability density function or probability mass function) where possible values associated with a random variable are equally possible. A fair dice is an example of a system corresponding to a uniform distribution (in that the probability of any two rolls are equal). The term “non-uniform distribution” may refer to a probability distribution where all possible values or

intervals are not equally possible. A Gaussian distribution is an example of a non-uniform distribution.

[0024] “Classification” may refer to a process by which something (such as a data value, feature vector, etc.) is associated with a particular class of things. For example, an image can be classified as being an image of a dog. “Anomaly detection” can refer to a classification process by which something is classified as being normal or an anomaly. An “anomaly” may refer to something that is unusual, infrequently observed, or undesirable. For example, in the context of email communications, a spam email may be considered an anomaly, while a non-spam email may be considered normal. Classification and anomaly detection can be carried out using a machine learning model.

[0025] The term “artificial intelligence model” or “machine learning model” can include a model that may be used to predict outcomes to achieve a pre-defined goal. A machine learning model may be developed using a learning process, in which training data is classified based on known or inferred patterns.

[0026] “Machine learning” can include an artificial intelligence process in which software applications may be trained to make accurate predictions through learning. The predictions can be generated by applying input data to a predictive model formed from performing statistical analyses on aggregated data. A model can be trained using training data, such that the model may be used to make accurate predictions. The prediction can be, for example, a classification of an image (e.g., identifying images of cats on the Internet) or as another example, a recommendation (e.g., a movie that a user may like or a restaurant that a consumer might enjoy).

[0027] A “machine learning model” may include an application of artificial intelligence that provides systems with the ability to automatically learn and improve from experience without explicitly being programmed. A machine learning model may include a set of software routines and parameters that can predict an output of a process (e.g., identification of an attacker of a computer network, authentication of a computer, a suitable recommendation based on a user search query, etc.) based on feature vectors or other input data. A structure of the software routines (e.g., number of subroutines and the relation between them) and/or the values of the parameters can be determined in a training process, which can use actual results of the process that is being modeled, e.g., the identification of

different classes of input data. Examples of machine learning models include support vector machines (SVM), models that classify data by establishing a gap or boundary between inputs of different classifications, as well as neural networks, collections of artificial “neurons” that perform functions by activating in response to inputs. The model may include linear regression, logistic regression, convolutional neural network (CNN), deep recurrent neural network (e.g., fully-connected recurrent neural network (RNN), Gated Recurrent Unit (GRU), long short-term memory, (LSTM)), transformed-based methods (e.g. XLNet, BERT, XLM, RoBERTa), hidden Markov model (HMM), linear discriminant analysis (LDA), k-means clustering, density-based spatial clustering of applications with noise (DBSCAN), random forest algorithm, adaptive boosting (AdaBoost), eXtreme Gradient Boosting (XGBoost), support vector machine (SVM), or any model described herein. A machine learning model can be trained using “training data” (e.g., to identify patterns in the training data) and then apply this training when it is used for its intended purpose.

[0028] A machine learning model can be trained using “training data” (e.g., to identify patterns in the training data) and then apply this training when it is used for its intended purpose.

[0029] A machine learning model may be defined by “model parameters,” which can comprise numerical values that define how the machine learning model performs its function. Training a machine learning model can comprise an iterative process used to determine a set of model parameters that achieve the best performance for the model. Supervised learning models can be trained in various ways using various cost/loss functions that define the error from the known label (e.g., least squares and absolute difference from known classification) and various optimization techniques, e.g., using backpropagation, steepest descent, conjugate gradient, and Newton and quasi-Newton techniques.

DETAILED DESCRIPTION

[0030] Embodiments include methods and systems for correcting algorithmic bias in machine learning or artificial intelligence frameworks.

[0031] In some examples, the method may comprise any combination of the following features. The method may comprise obtaining a first dataset, wherein the first dataset contains features or information related to at least one protected attribute. The first dataset may be a limited dataset for training purposes and smaller than a second dataset. A second

dataset may be obtained, wherein the second dataset does not contain classifications, information, or tags related to protected attributes. A protected attributes estimation model can be trained using first dataset. At least one protected attribute can be estimated for data within the second dataset.

[0032] This disclosed technology includes an upstream and downstream-based framework that leverages active learning to predict protected attributes from limited labeled data and utilizes these predictions into the learning process to achieve fairness in the target task. The upstream model employs active learning for accurate protected attribute prediction using limited labeled data, while the downstream model incrementally samples low-biased training examples for fair learning based on the upstream estimates.

[0033] The framework can comprise two stages. The first stage (also referred to as the upstream stage or PA estimation model generation stage) can comprise of training a protected attributes estimation model. The upstream stage can utilize active learning to generate a protected attributes estimating model. The first stage or upstream stage can comprise any of the steps described with respect to FIG. 2.

[0034] The second stage can comprise using a selected PA estimating model to generate a number of fair models. In some examples, the second stage can be considered to be a downstream stage. The second stage or downstream stage can comprise any of the steps described with respect to FIG. 3.

[0035] The selection of a final model can be used in a production environment to analyze data while minimizing bias. Thus, the final selected model is generated while using a limited or restricted amount of data containing protected attributes while generating a low-biased model.

[0036] Experiments on benchmark datasets illustrate the two-stage framework's effectiveness in both predictive performance and fairness.

I. FAIRNESS IN MACHINE LEARNING

[0037] It may be difficult to train a machine learning model while ensuring it is unbiased when certain features related to the dataset are unavailable. This problem is exacerbated when the size of the data with certain features (e.g., protected attributes) is small (e.g., not large

enough, small compared to the size of the entire dataset, not large enough to produce statistically meaningful training, etc.).

[0038] A description of fairness in the context of machine learning is useful. Fairness generally refers to correcting bias in decision processes (e.g., outputs) implemented using machine learning models. Decisions made by computers after a machine-learning process may be considered unfair if they were based on variables (also referred to as features or attributes) considered sensitive, such as for example, race, gender, sexuality, religion. Fairness may be viewed from a group perspective (e.g., a specific group is not discriminated against) and from an individual perspective (e.g., whether similar individuals are treated similarly when excluding a sensitive variable). At times, ML algorithms may affect people in unfair ways when used to automate decision making, such as in legal, academic, or other context.

[0039] A machine learning process may become unfair (also referred to as being biased) due to various reasons. These reasons may include issues with the underlying training data, such as skewed samples (e.g., bias compounded in the dataset due to an initial bias in the dataset), limited features (e.g., features which may be less informative for certain groups), sample size disparity (e.g., the sample size for certain groups may be smaller than those for the remaining groups or the largest group within the data), and the presence of proxies (e.g., other features correlated to sensitive features which may lead to unfairness).

[0040] It is to be understood that various quantitative formulations of “fairness” may be possible. As one example, fairness may be defined in terms of a proportion of a class that receives a particular outcome. For example, for a binary category (e.g., a color of a toy such as green or blue), if one color receives a positive outcome less than a certain percentage of the other color, it can be deemed that the fairness criteria is violated. The threshold may be a defined percentage of the other group. For example, the threshold may be a fixed percentage (e.g., the proportion of green toy should be at least 80% of the proportion of blue toys for a particular classification). The particular portion may be based on legislation or other rules. For example, Title VII of the Civil Rights Act of 1964 provides that there is disparate impact when the ratio of probabilities of a particular outcome between two groups is less than 80%. The definition of fairness can be generalized for multiple features within a dataset (e.g., blue, green, purple, red, pink, etc.).

II. EXAMPLE DATASETS WITH PROTECTED ATTRIBUTES

[0041] FIG. 1 illustrates an example dataset 100. Dataset 100 is a fabricated dataset for illustrative purposes and describes features related to various toys. It is to be understood that dataset 100 may contain any number of rows (e.g., datapoints) and columns (e.g., each column related to a feature). Dataset 100 describes various features related of toys, identified with IDs 1 to 10 in column 110. For each toy, the toy may have a name (Toy_Name in column 111), color (column 112), shape (column 113), manufacturer (column 114), material type (Material_Type in column 115), age group (Age_Group in column 116), price (column 117), country of origin (column 118), supply chain method (column 119), and demand (column 120). Demand, which is either low, medium, or high, may be a target task or output on which the dataset can be trained.

[0042] Once trained, a machine learning model may classify new toys into one of the three demand categories. Additionally, the machine learning model may provide statistical information related to an output, such as a confidence metric related to the output. For instance, the classification may be the demand for a toy as low, medium, or high. In other examples, the classification may be binary, such as a yes or a no output from the model.

[0043] Specific features within dataset 100 may be considered as protected features. Protected features may also be referred to herein as protected attributes (PAs). For instance, the country of origin for the toy may be deemed a protected attribute. Ideally, in a “fair” model, this protective attribute should have a minimal or no predictive effect on the trained machine learning model. As another example, the supply chain method chosen for the toy may be a protected attribute. Similarly, for example, the supply chain method may be thought to be unrelated to how strong the demand for a particular toy would be. Thus, the supply chain method may also be deemed to be a protective attribute. Whether or not a particular feature has predictive value with respect to an outcome may be considered in its selection as a protected attribute.

[0044] FIG. 1 also illustrates dataset 199. In this example, dataset 199 is derived from dataset 100 by removing elements from the dataset. Thus, in dataset 199, a protected attribute is accessible for only three points of data, corresponding to IDs 1, 5, and 6. As the training data containing protected attributes is limited, it would be difficult to robustly train a machine learning model based on dataset 199 while also ensuring that the predictive effect of the

protected attributes is minimized. For example, the limited number of points with protected attributes would lead to a model which is inaccurate or not robustly trained.

[0045] The fairness of models trained on the example datasets 100 and 199 may be examined. Various approaches may be taken with respect to ensuring that a trained model is fair. For example, a fairness metric may be measured for an algorithm during the training of the algorithm. If the fairness metric is violated, an objective function may be penalized to better reflect the fairness criteria described. Stated alternatively, a constraint may be placed on the training of the object function that it can respect the fairness criteria. This may also be referred to as constraint-based optimization approach. However, the constraint-based optimization approach relies on the ability to distinguish between group membership and requires group membership information for all the data instances in the dataset. Thus, while this approach may work for dataset 100, it will not be useful to dataset 199 as the dataset is incomplete. Thus, such solutions rooted in computing a fairness metric are inapplicable to the problem of fairness when considering restraints on access to protected attributes. These solutions are provided using the two-stage framework discussed herein.

[0046] Thus, as further explained below, a two-stage framework using active learning can be used to ensure fairness where only a small subset of the data contains protected attributes. As further explained below, at the first stage of the framework, to overcome the problem of classification with limited data, an estimation model to estimate the protected attributes can be trained. This model may also generate a probability for each estimated protected attributes (e.g., a probability metric related to the estimation). At the second stage, data points with probability close to 50% where it is difficult to determine which class the data point belongs in can be used to train a model. The range of probability can be expanded to train additional models from which one model can be chosen to be a final model (e.g., a model used in a production environment to make classifications in the future).

III. OVERVIEW OF TWO STAGE MODEL

[0047] An overview of the two stage model is provided prior to additional discussion of each stage of the two stage model. FIG. 1B illustrates framework 150 with an upstream stage 151 (also referred to as a first stage or a protected attributes (PA) estimation stage) and a downstream stage 152 (also referred to as a second stage or a fair model generation stage). Variables discussed with reference to FIG. 1B are described in additional detail below.

[0048] Referring first to the upstream stage, 151, a dataset D may be used to train a bias estimation model (assigned symbol M_{be} and also referred to as a PA estimation model). Dataset D may comprise of two subsets – a first subset of data containing protected attributes information (assigned symbol D_{pal} and also referred to as PA known dataset) and a second subset which does not contain information related to protected attributes (assigned symbol D_{pau} and also referred to as PA unknown dataset). The training of the bias estimation model is further discussed herein.

[0049] Once trained, the bias estimation model may predict protected attributes for datapoints within the PA unknown dataset (D_{pau}) and also provide quantitative information about those predictions (e.g., the confidence in the prediction expressed as a probability). Information outputted by the bias estimation model may be appended to the dataset without protected attributes and used in the downstream stage 152. Once appended, this may be referred to as the augmented dataset (assigned symbol D_{pauPL}). The augmented dataset may also optionally contain statistic information related to the predictions obtained from the bias estimation model.

[0050] At the downstream stage 152, multiple models may be trained using the augmented dataset (D_{pauPL}). Subsets of the augmented dataset may be created through a selection process. Each such subset of the augmented dataset may be used to train a different bias mitigation model. The selection process may be based on, for example, a confidence range of each prediction of the bias estimation model. For example, “i” subsets may be generated and “i” bias mitigation models may be generated. These “i” bias mitigation models may be referred to as M_{bmi} . One model from the generated “i” bias mitigation models may be chosen based on performance criteria (e.g., for use in a production environment. This model may be assigned symbol M_{bm} .

[0051] The two stage model may be advantageous as it allows for bias correction (e.g., through training a bias mitigation model) even when the underlying protected attributes are not known. The two stage model thus allows for training of fair models (e.g., bias mitigation models) on a small dataset PA known while retaining robust performance. Further, the two stage model may not need information from any the PA known dataset (D_{pal}) to train the bias estimation model.

IV. FIRST STAGE: PROTECTED ATTRIBUTES (PA) ESTIMATION MODEL

[0052] As further explained below with respect to FIG. 2, a protected attributes (PAs) estimation model may be trained on a subset of data which has protected attributes. The protected attributes estimation model may also be referred to as an upstream model or a bias estimation model. After training, the protected attributes estimation model may take as an input data which does not contain the protected attributes (e.g., the PA unknown dataset D_{pau}) and output the protected attributes for that data.

[0053] The training of the PA estimation model may be done through active learning. Active learning is semi-supervised learning where the performance of the model may be gradually improved. A small subset of the training data originally be used for training with additional and more valuable data from the remaining of the dataset being used to supplement the training of the model. During the training of the PA estimation model, an attempt will be made to minimize the number of queries to the protected attributes. If an iteration of the trained protected attributes model is not robust or does not meet certain performance criteria, additional datasets may be requested. In some examples, human annotator may be used to label or provide additional data instances to provide additional data.

[0054] FIG. 2 illustrates the framework 200 which may be used to generate a PA estimation model. Figure 2 illustrates a datasets 210A, 210B, and 210C, and PA estimation models 220A, 220B, and 220C corresponding to and trained on each dataset respectively. An active learning step between the data sets is also illustrated. For example, the data in dataset 210A may be supplemented to create dataset 210B through an active learning process. Similarly, dataset 210A or 210B may be supplemented to create dataset 210C. Although only 3 datasets and PA estimation models are illustrated in FIG. 2, it should be understood that the process of generating datasets is iterative and any number of datasets may be generated through active learning and a corresponding number of PA estimation models may be trained.

[0055] Active learning allows an algorithm to determine whether to obtain additional information regarding a dataset based on a cost-benefit analysis. In active learning, an algorithm may proactively select a subset of examples to be labeled next from the pool of unlabeled data. Active learning may include stream-based selective sampling techniques, pool-based sampling techniques, and membership query synthesis techniques. In stream-based selective sampling, a decision is made for data instances and whether it would be

worthwhile to query that data instance for a label. In pool-based sampling, a determination is made on which subset of data would be most beneficial to insert into the next training set of data for an active learning loop. In membership query synthesis, synthetic data may be generated for inclusion in datasets. In some instances of active learning, a human operator may label the data with desired outputs. With respect to the PA estimation model described in FIG. 2, the desired outputs may be one or more of the protected attributes for which the PA estimation model is being trained.

[0056] Datasets 210A, 210B, and 210C may be subsets of a larger data set which include information about protected attributes. As one example, with reference to FIG. 1, rows 1, 5, and 6 of dataset 199 may form dataset 210A. As may be seen in FIG. 1, certain information regarding one or more protected attributes (e.g., country of origin and supply chain method in columns 118 and 119 of dataset 199) may be missing. As previously explained, this information may be supplemented through an active learning technique. For instance, a human operator may provide information regarding the supply chain method in row 1 and the country of origin in row 5. This information may be included in dataset 210B, on which training may be performed to generate PA estimation model 220B, which may be more accurate than PA estimation model 220A as more data is available. Additional information may transform dataset 210A or dataset 210B into dataset 230C by including additional selected examples for which protected attributes are added. For instance, row 3 in table 199 which has no information related to protected attributes may be selected and protected attribute data may be added to this row to transform this datapoint into one that may be used for training a PA estimation model. Thus, in this manner, the dataset may be recursively updated through active learning and through providing additional datapoints to further train PA estimation models.

[0057] PA estimation models 220A, 220B, and 220C may be machine learning models. For example, PA estimation model may be a machine learning model that is trained to generate predicted classifications for a value of a protected attribute. These output classifications may be one or more protected attributes. Additionally, the PA estimation models 220A, 220B, and 220C may generate additional statistical information along with the generated classifications. For example, metrics related to the confidence of the classification may be provided by PA estimation models. In some examples, additional columns may be appended to the table illustrated with respect to FIG. 1 to include statistical information.

[0058] Statistical performance for a PA estimation model may be generated upon training of the model. This statistical information may include a probability that a particular data point contains a protected attribute. For example, the protected attribute estimation model may be tested on the validation dataset and rank the confidence of the model on these data instances and find out at which datapoint(s) the model is not confident. Among the least confident data instances, a subset of the data (e.g., the last ten data instances) may be selected to be annotated with through an active learning framework. The training process may be iterated until a threshold for the training is reached (e.g., a resources budget or model performance metric).

[0059] Upon training, the PA estimation model may be used as further described below in a second stage.

[0060] Additional embodiments of training of a PA estimation model are provided below. The following symbols may be further discussed prior to discussing additional embodiments.

- M_{be} : M_{be} is a machine learning model for bias estimation. M_{be} may be trained on the PA known dataset (D_{pal}). After training, M_{be} may generate a protected attribute for each data instance in the PA unknown dataset. M_{be} may be trained in upstream stage. M_{be} may also be referred to as a PA estimation model. M_{be} may be trained in stages until a final model is obtained through an active learning process. D_{pal} : A subset of dataset D in which protected attributes are known (e.g., labeled). This is also referred to as the PAs known dataset. This dataset may be updated with additional datapoints for which PAs are known through active learning.
- P_{pau} : An estimated probability outputted for one or more points from a PA estimation model.
- D_{pau} : A subset of dataset D in which protected attributes are unknown (e.g., unlabeled). This is also referred to as the PAs unknown dataset.
- D: A dataset which contains data with both known protected attributes and unknown protected attributes. It may also be referred to as a complete dataset. For example, table 199 may be an example of such a dataset as it contains both datapoints for which protected attributes are known and datapoints for which

protected attributes are unknown. D is a dataset formed from two subsets D_{pal} and D_{pau} . In some examples, the size of $D_{pau} \gg D_{pal}$.

- D_{paupl} : Dataset generated from D_{pau} using M_{be} , and includes estimate protected attributes. This is also referred to as the augmented dataset.

[0061] The PA estimation model (M_{be}) may be trained with the following method. M_{be} is an ML model which takes data D and produces outcome PA for each individual with a particular protected attribute PA .

Input Data to Method: Dataset PA known (D_{pal}), and dataset PA unknown (D_{pau})

Output of Method: Trained PA estimation model M_{be}

Steps of Method :

```

1  $M_{be} \leftarrow \text{train}(D_{pal})$ 
2  $P_{pau} \leftarrow M_{be}.\text{predict}(D_{pau})$ 
3 Compute  $Q_u$  using the Eqn. 1.
4 repeat steps 5 to 8 until  $M_{be}$  sufficiently trained
   /* Select k number of examples from unlabeled pool */
5    $D_q \leftarrow M_{be}.Q(D_{pau}, k)$ ; where  $k > 0$ 
6   Annotate selected examples  $D_q$  via human experts
7    $D_{pal} \leftarrow D_{pal} \cup D_q$ 
8    $M_{be} \leftarrow M_{be}.\text{update}(D_{pal})$ 
9 until
10 return  $M_{be}$ 

```

[0062] The selection of samples from the set of D_{pau} may include the use of a ranked query approach. Examples of such an approach includes the batch-mode active learning and ranked batch-mode active learning techniques described in Cardoso, T. N. C., Silva, R. M., Canuto, S., Moro, M. M., & Gonçalves, M. A. "Ranked batch-mode active learning." Information Sciences, vol. 379, no. 10, I. 313-337, 2017. Q_u may be calculated for each datapoints, which may then be ranked based on the calculated Q_u values. The approach may include both uncertainty and similarity (e.g., as measured by a Euclidean distance metric) between data points used for training by using the following equation:

$$Q_u = \alpha * (1 - \sigma(x, D_{pau})) + (1 - \alpha) * U(x), \text{ where:}$$

x represents a particular example or datapoint within a data subset (e.g., such as within the group D_{pal});

σ measures the similarity between a given example and the pool of examples in D_{pau} ; and

U indicates the uncertainty of that example, and α is a weight used to combine uncertainty and similarity.

[0063] After training and choosing the final PA estimation model (M_{be}), protected attributes for unlabeled data may be generated using the PA estimation model. Statistical information may also be output by M_{be} . For example, the data instances in D_{pau} may be analyzed to generate the dataset D_{paupl} . Each data instance in D_{paupl} may contain statistical information (e.g., uncertainty information, probability information, confidence values). As further explained below, the set D_{paupl} may be sampled to produce a number of models in the second stage.

V. STAGE TWO: TRAINING FAIR MODELS

[0064] During the stage two (also referred to as a downstream stage, second stage, or a fair model training stage), multiple machine learning models may be trained. The training of each of the multiple machine learning models may be based on a different subset of the augmented dataset (D_{paupl}). The augmented dataset includes predicted PAs, outputted from the bias estimation model discussed above, which are augmented to the PA unknown dataset (D_{pau}). Each of the machine learning models (which may be referred to as fair models and assigned symbol M_{bmi} , where i is an index) may be measured for accuracy and fairness. One of the multiple fair models may be selected for use in predicting on a target task. The final selected fair model may be represented as M_{bm} . The selection of the final model may be based on requirements from the final model.

[0065] The training of a fair model may be performed on the augmented dataset (D_{paupl}), rather than the PA dataset (D_{pal}) or complete dataset (D), as training on either of those datasets would propagate any bias that is present in the data into the fair model. The propagation of bias may be mitigated by using the augmented dataset where statistical information about each predicted PA is known.

A. Overview

[0066] FIG. 3 illustrates a framework 300 to generate a bias mitigation model from dataset D. More specifically, the bias mitigation model may be generated from the augmented dataset (D_{paupl}) 320. Illustrated in FIG. 3 is PA Estimation Model 310, PA unknown dataset (D_{pau}) 311, augmented dataset 320, datasets generated from uncertainty sampling 330A, 330B, and 330C, a plurality of bias mitigation models 340A, 340B, and 340C, and a final bias mitigation model 350. PA estimation model 310 may be the bias estimation model (M_{be}) described in FIG. 2. Dataset 320 may be the augmented dataset (D_{paupL}), which may include information outputted from the PA estimation model 310 and the PA unknown dataset 311.

[0067] Datasets 330A, 330B, and 330C may be chosen based on statistical information (e.g., uncertainty of predictions datapoints within the datasets) from dataset 320. Datasets 330A, 330B, and 330C may be subsets of dataset 320. Bias mitigation models 340A, 340B, and 340C may be models which are trained on datasets 340A, 340B, and 340C respectively. Although only 3 datasets and 3 corresponding models are illustrated, any “i” number of datasets may be selected to train “i” potential models. Final bias mitigation model 350 may be chosen from one or more models based on criteria, such as for example performance, accuracy and bias in the model. Examples of criteria which may be considered are further described with respect to FIGs. 6–8, such as precision (such as measured by the number of true positives divided by the total number of positive predictions) and F-scores.

[0068] The generation and training of models in framework 300 relies on output from the PA estimation model. Thus, at this stage, protected attributes that were originally known from the dataset D (e.g., protected attributes known dataset D_{pal}) need not be used. Attributes, which were estimated from the PA estimation model alone, may be used. Thus, the dataset D_{paupL} along with other algorithms, such as a disparate impact remover (DIR) algorithm, may be used to training. The training of each bias mitigation model (340A, 340B, 340C, etc.) may be based on a modification to the training data rather than a modification to the underlying machine learning model.

B. Creating Subsets from Augmented Dataset 320

[0069] Each datapoint in the augmented dataset 320 (D_{paupl}) contains both (i) features unrelated to the (e.g., features which are not protected attributes) and (ii) protected attributes

that are estimated from the bias estimation model (M_{be}). Each estimated protected attribute outputted from the bias estimation model may be associated with statistical information. For example, with respect to a protected attribute which is binary (e.g., round or square), the statistical information may be the confidence of the value estimated for the protected attribute by the bias estimation model. Datasets 330A, 330B, and 330C may be generated from the augmented dataset 320 (D_{paupL}). Each dataset may be based on statistical attributes of the protected attributes, such as how confident the bias estimation model may be regarding a prediction. Dataset 330A may contain protected attributes for which it is harder to distinguish between group membership as the confidence about a prediction is low. Dataset 330B may include additional datapoints for which there is a higher confidence regarding their prediction. Dataset 330C may include even additional datapoints for which there is an even higher confidence regarding their prediction. In this manner, each dataset may be used to train a different bias mitigation model. As additional datapoints are included, the accuracy of the trained bias mitigation model may increase. However fairness may decrease as points which are more biased are included.

[0070] The confidence may be represented by a value from a range, such as between 0 and 1. For binary values, may be between a range (e.g., 0 to 1) such that a value of 0 implies a 100% confidence that the estimated value is the first of the binary values (e.g., round) while a value of 1 may imply a 100% confidence that the estimate value is the second of the binary values (e.g., square). Such extreme values would imply a perfect prediction. A confidence metric of .5 may imply that the bias estimation model is 0% confident about the value it has predicted (e.g., the bias estimation model is entirely unsure whether the protected attribute should be round or square). Confidence metrics between 0 and .5, and .5 and 1 may similarly denote confidences which are not extremes. For example, a confidence metric of .25 may imply that there is a 75% confidence that predicted value is the first of the binary values (e.g., round).

[0071] Subsets of the augmented dataset 320 (D_{paupL}) may be determined based on values which lay within a range of confidence metrics. As discussed above, each datapoint in the augmented dataset 320 will have a confidence metric associated for the protected attribute predicted for that datapoint. Based on the confidence metrics, the datapoints within the augmented dataset may be divided into subsets. For instance, datasets that are iteratively larger from a central value (e.g., .5) may be used for training bias mitigation models. For

instance, a k^{th} subset may include all points which have confidence metrics between $[\cdot 5 - kq, \cdot 5 + kq]$, where q is an arbitrary real number and k is an index. As k increases, each subset may be nested within a next subset. Each subset may be used to train a bias mitigation model.

[0072] Above, for index $n < m$, the m^{th} subset will contain all datapoints from the n^{th} subset and additional datapoints. However, the m^{th} subset will contain a broader range of confidence metrics which are further from $\cdot 5$ (corresponding to a confidence of 0%). As a 0% confidence implies a completely “neutral” value for a protected attribute, values farther from 0% may reflect bias for that predicted protected attributes. Thus, including such values will reduce fairness in a bias mitigation model generated. However, for robust training, additional datapoints are required to improve precision. Thus, each bias mitigation model trained on a different dataset may contain different characteristics in terms of tradeoffs between fairness and precision.

[0073] In some examples, the subsets may be chosen by increasing the number of datapoints within a confidence range but not increasing the confidence range. Additionally, pre-processing steps may be used to ensure that a robust sample of datapoints is included in the subsets.

C. Training Bias Mitigation Model from Datasets

[0074] Datasets 330A, 330B, and 330C may be created from the augmented dataset 320 (D_{pauPL}). For example, each dataset may be based on intervals related to statistical information of the generated protected attributes. For instance, dataset 330A may include all datapoints in D_{pauPL} , which have a confidence metric of $\cdot 45 - \cdot 55$ associated with an estimated protected attribute. Dataset 330B may include a larger range. For example, dataset 330B may include all datapoints in D_{pauPL} which have a confidence metric of $\cdot 4 - \cdot 6$ associated with them. Dataset 330C may contain an ever larger range. For example, dataset 330C may have contain all data points between $\cdot 3$ and $\cdot 7$ in their confidence metric. In some examples, a pre-processing algorithm may be used to process the data instances being selected for datasets 330A, 330B, and 330C. For example, fairness preprocessing algorithm, such as the disparate impact remover may be used. A disparate impact remover (DIR) is a preprocessing technique that edits feature values to increase group fairness while preserving rank-ordering within groups.

[0075] DIR techniques can reduce or remove inherent biases from a set of data to minimize the effect of such biases on an algorithm trained with that data. Certain DIR algorithms employ an optimization approach to transform a dataset with the aim of eliminating bias while preserving data utility to the extent possible. Certain DIR methodologies are rooted in the legal concept of disparate impact, making them relevant for real-world applications when legal compliance is critical. In some examples, the same DIR technique applied to training data may also be applied to production data (e.g., data which is to be classified by a machine learning model).

[0076] Datasets may be chosen in this manner to obtain robust models. By choosing low-biased training examples (e.g., those close to a probability of .5) and increasing the number of datapoints incrementally with additional datapoints that are more biased (e.g., those in a larger range around .5), training can be achieved with a sufficient number of points to obtain good performance while still limiting the amount of datapoints which may cause unneeded bias into the model. It may be expected that as the number of datapoints increases (e.g., as the datasets used for training are larger), precision of the trained model will increase but bias may decrease.

[0077] Points where the protected attribute estimation model is less confident may imply that the information or data contents have not contributed to bias as the protected attribute estimation model is not able to accurately predict the protected attribute for that particular data instance. As one example, if a confidence metric (e.g., probability) that a particular toy is one of two values is $1/2$ (e.g., low or high), it implies that the PA estimation model is not sure which group the toy belongs to. A value of $1/2$ in this example may imply perfectly unbiased datapoint as the model is unable to determine which value to assign for the protected attribute for this point.

[0078] A final model 350 may be selected from the multiple trained models based on accuracy and fairness requirements. Any other metric may be chosen as a basis for selecting a final model 350. Final model 350 may be selected for use in a production environment to determine outputs for data which requires classification.

[0079] An additional description of training a fair model is described herein. The following symbols and variables may be defined prior to describing the mathematical problem.

- L , r , and τ are the hyper-parameters for sample selection. l and r are used to gradually increase the sample sizes in each iteration in terms of the confidence score of each iteration of the PA estimation model. τ or τ_e is a regularizer. To refine the quality of the samples with increasing iterations, we regularize by reducing the number of samples with the regularization factor τ .
- D_{mb} : This is a dataset which trains in the second stage (downstream stage). It may also be referred to as a bias mitigation dataset as it is intended to mitigate bias. This may also be referred to as D_{pauPL} .
- τ_e is a regularizer.
- $|l - r|$ represents the distance from the current lower biased region to the current high biased region. Stated alternatively, this represents a range of values.
- $|D_{mb}|$ is the number of examples in the new examples pool

Method: Training Fair Models

Input : Datasets D_{pauPL} , Thresholds l, r, τ, m_b

Output: Target model M_{bm}

```

1 Procedure trainDTM( $D_{mb}, G$ )
2    $D_{mb} \leftarrow DIR(D_{mb}, G)$ 
3    $M_{bm} \leftarrow \text{train}(D_{mb})$  // train ERM
4    $P_{valid} \leftarrow M_{bm}.\text{predict}(D_{valid})$ 
5    $E_m \leftarrow$  model performance.
6   return  $M_{bm}, E_m$ 
7  $i = 0$ 
8 repeat (9 to 28)
9   if  $i < 1$  then
10     $D_{mb} \leftarrow D_{pauPL} [l : r]$ 
11     $M_{bm}, E_m \leftarrow \text{trainDTM}(D_{mb}, G);$ 
12  end
13  else
14    /* Sample additional data instances by moving further in confidence
15       range */
16     $l_{i-1} \leftarrow l; r_{i-1} \leftarrow r$ 
17     $l \leftarrow l - \tau; r \leftarrow r + \tau$ 
18    /*  $D_{mb}$  from previous iteration */
19     $D_{mb_{i-1}} \leftarrow D_{pauPL} [l_{i-1} : r_{i-1}]$ 
20     $D_{mb_i} \leftarrow D_{pauPL} [l_{i-1} : l] \cup D_{pauPL} [r_{i-1} : r]$ 
21     $D_{mb_i} \leftarrow \text{rank}(M_{bm}.Q_u(D_{mb_i}))$ 
22     $\Delta = 1 / (1 - |l - r|)$ 
23    if  $i \% m = 0$  (modulo operator) then
24       $\lambda = 1/\tau_e$ 
25    end

```

```

23   Get  $k_{ed}$ 
24    $D_{mbi} \leftarrow D_{mbi} [: k_{ed}]$ 
25    $D_{mb} \leftarrow D_{mbi-1} \cup D_{mbi}$ 
26    $M_{bm}, E_m \leftarrow \text{trainDTM}(D_{mb}, G)$ 
27    $i = i + 1$ 
28   end
29 until
30 return  $M_{bm}$ 

```

[0080] By categorizing training data based on the estimated bias from the bias estimation model, low-biased and high-biased estimated can be empirically obtained based on confidence values. The threshold for what is included in a training data set can be increased based on, for example, the parameters above. For example, in binary cases, threshold parameters may be entered around .5. The training dataset may be constructed by including additional instances of data with each iteration. As discussed above, the fair model (a chosen final M_{bm}) may be chosen and put into a production environment.

VI. EXAMPLE METHODS

[0081] Example methods are illustrated below related data selection, active learning, and training of machine learning models (e.g., protected attributes estimation models (in a first stage), classification models (in the second stage)) according to embodiments of the disclosed technology.

A. Generating Protected Attribute Estimation Model

[0082] With respect to FIG. 4, an example method 400 is provided to generate an upstream estimation model. The upstream estimation model may also be referred to as a protected attributes estimation model. Example method 400 may utilize or combine any of the techniques described with respect to the disclosure herein.

[0083] At step 402, a first dataset can be obtained. The first dataset can contain features related to at least one protected attribute. In some examples, the first dataset can be selected or generated from a larger dataset which contains protected attributes. For example, first dataset may be PA known dataset (D_{pai}) which may be obtained from dataset D , which are described above.

[0084] At step 404, a second dataset can be obtained. The second dataset may not contain features, information, tags, or classifications related to at least one protected attribute. In some examples, the second dataset can be larger than the first dataset and representative of data which would be classified as part of a target task. For example, second dataset may be the PA unknown dataset (D_{pau}) which may be obtained from dataset D , which are described above.

[0085] At step 406, a first dataset can be used to train a protected attributes estimation model. The protected attributes estimation model can be a machine learning model which can estimate information or classifications related to at least one protected attribute. In some examples, the protected attributes estimation model may be trained for a single attribute or multiple attributes.

[0086] At step 408, the protected attributes estimation model can be updated. In some examples, a human expert can annotate or provide information on a selected number of examples of the second dataset to allow for a next iteration or version of the protected attributes estimation model. Step 408 may be iterated or repeated as required. In some examples, the annotation can be simulated. For example, to replicate the role of a human expert or human operator in annotating additional data, additional labels can be revealed to simulate interactive annotation. For example, a subset of the dataset which is labeled may be used to train a PA estimation model (e.g., 30% of the labeled data). The remainder of the dataset may be treated as “unlabeled” despite the PA for that data being known. After training on the subset, the PA estimation model may be used to make predictions regarding the protected attributes for the other data (e.g., the other 70% of the labeled data). For those predictions, the datapoints where uncertainty is the highest may be chosen. Rather than having a human expert label or annotate those points (which are known, but hidden to the PA estimation model), those datapoints (along with the known protected attributes) may be provided to further train the PA estimation model by adding those specific instances to the training dataset.

[0087] In some examples, datapoints from the second dataset (e.g., PA unknown dataset) may be chosen and annotated with a human expert to update the first dataset (e.g., PA known dataset). The updated first dataset may be used in the training of iteration. In some examples, the datapoints chosen from the second dataset may be used to be included in the first dataset.

[0088] At step 410, at least one protected attribute for data within the second dataset can be estimated. The estimation can be based on the trained protected attributes estimation model. The estimation (also referred to as a prediction) by the protected attributes estimation model may also provide statistical information related to the prediction.

[0089] In some examples, method 400 can be considered an upstream estimation model and be used in conjunction with a downstream model. In some examples, updating of the protected attributes estimation model can occur iteratively, based on some criteria, such as the resource, budget, number of examples, or availability of human experts. In some examples, the method can further comprise ranking the selected number of examples from the second dataset by empirically defined bias.

[0090] One output of method 400 may be a protected attributes model. This model may be used on a dataset, such as dataset D_{pau} , which does not contain any information related to protected attributes.

B. Generating a Bias Mitigation Model

[0091] FIG. 5 illustrates method 500. Example method 500 can be used to generate a number of fair models and select one of the fair models as a final model based on fairness, bias, accuracy, or other metrics. Example method 500 may utilize or combine any of the techniques described with respect to the disclosure herein inclusive of the appendices attached. The generated fair models may be considered downstream as they are based on sampling predicted protected attributes from an unlabeled dataset and generated “downstream” from such unlabeled data. In some examples, the fairness models may be referred to as bias mitigation models (M_{bm} or M_{bmi}).

[0092] At block 502, a dataset containing protected attributes predicted using a PA estimation model may be obtained (e.g., an augmented dataset). In some examples a dataset containing predicted protected attributes (e.g., an augmented dataset (D_{paupl})) can be generated by using a PA estimation model on the PA unknown dataset (D_{pau}). The dataset D_{paupl}

may also contain statistical information about each predicted PA for each data instance. This information may be used as described below to create training sets. The statistical information may be obtained from the PA estimation model.

[0093] At block 504, statistical data from the dataset may be sampled to obtain a set of training data. The sampling may be performed based on a range associated with a statistical metric of the predicted protected attribute (e.g., confidence of a prediction expressed as a probability). For example, a small range may be chosen at this step (such as 48-52% confidence). In some examples, a preprocessing algorithm may be applied on the set of training data. These may include fairness preprocessing algorithm such as a disparate impact remover.

[0094] At block 506, a bias mitigation model may be trained using the set of training data, as modified by the preprocessing algorithm. Statistical information and performance information of the model may also be obtained at this step.

[0095] At block 508, a fairness metric and an accuracy metric can be obtained for one or more of the trained models. An accuracy metric may be one such as precision of the predictions made for a target task (e.g., a target classification). A fairness metric may include one or more measures of fairness, including measures based on group fairness, group unawareness, demographic parity, disparate impact, equal opportunity, equal odds, positive predicted value parity, false positive rate parity, and negative predicted value parity.

[0096] In some examples, a confusion matrix may be used to study one or more fairness metrics. The confusion matrix may also be used to determine an accuracy metric. An accuracy metric may correspond to the number of correct predictions/ total number of predictions. A confusion matrix may include a matrix representing, for a dataset, the number of true positives (the total number of outcomes where the model correctly predicts the positive class), true negatives (the total number of outcomes where the model correctly predicts the negative class), false positives (the total number of outcomes where the model incorrectly predicts the positive class), and false negatives (the total number of outcomes where the model incorrectly predicts the negative class.) The various numbers in the matrix may be compared or evaluated to determine a fairness metric and an accuracy metric.

[0097] Demographic parity may refer to the extent a protected attribute affects an outcome. For example, for a binary group, a prediction should be statistically independent of the protected attribute. Various metrics may be calculated based on demographic parity. For example, if there are two values for a binary group (A and B), we can define that for any outcome “O”, the probability for that outcome should be the same for both groups. A fairness

metric may be calculated as the difference between such probabilities or ratio between such probabilities. For a perfectly fair model, the accuracy ratio would be expected to be “1.”

[0098] Disparate impact may refer to when one group has a higher probability of getting a certain outcome than another outcome. In some examples, a fairness metric may be a binary function which returns a value of 0 or 1 when the ratio between the outcome for the two groups is higher than a certain ratio. For example, if the probability of an outcome for a first group divided by the probability of the same outcome for a second group is above a threshold (e.g., 4/5), the fairness metric function may return a 1 to indicate that there is a disparate impact. If not, the fairness metric function may return a 0. The threshold may be adjusted according to a use case.

[0099] Equal opportunity may refer to the true positive rate between two groups. A fairness metric may look at the difference between the true positive rate between two (or more) groups. For example, the true positive may be the payback of a loan. Other variations based on an equal opportunity may also form the basis for a fairness metric. For example, a false positive rate may be measured between two group and metrics may be based on the same.

[0100] Positive predicted value parity and negative predicted value parity can refer to when two groups have the same positive predicted value or the same negative predicted value. The positive predicted value may be defined in a binary case (e.g., the likelihood that an event occurs). A fairness metric may be based on variations from the parity values expected in both cases.

[0101] The described example fairness metrics and methods of calculations may further be modified based on requirements.

[0102] At block 509, additional models can be generated by iterating steps 504, 506, and 508. The iteration may occur by expanding the statistical range to obtain additional points of data. Thus, additional sets of training data may be obtained to train additional models. Step 509 may be undertaken as a sufficient number of examples may not be available based on the selected number of examples or datapoints, additional examples or datapoints can be obtained.

[0103] In some examples, additional datapoints can be selected based on criteria described herein, such as using a ranked entropy model. In the ranked entropy model, entropy Q_u can be

ranked for the i^{th} iteration of the set of training data D_{mbi} . D_{mb} may be a dataset with both non-estimated data attributes and protected attributes estimated from a PA estimation model. Examples within D_{mbi} may be ranked from high entropy examples to low entropy examples in order to prune that dataset to select the most informative and difficult-to-predict examples from the target task's perspective. Further, to account for the quality of the examples from the PA estimation models example, we may select a number, k_{ed} from D_{mbi} . In some examples, the following equation may be used to determine the number: $k_{ed} \leftarrow \min(\lambda \exp^{-\Delta |D_{mbi}|}, \epsilon)$. λ and Δ have been previously defined above. ϵ may be chosen to ensure a minimum number of samples are selected.

[0104] In some examples, the output of a protected attributes estimation model, such as a predicted feature or group (e.g., race or gender) can be provided to a fairness pre-processing algorithm, such as a disparate impact remover algorithm, to modify feature values to improve group or feature values.

[0105] At block 509, the training process may be iterated to generate additional fair models. In some examples, the statistical range may be expanded to include additional data points for training. For example, as explained above with respect to FIG. 3, the confidence metric associated with protected attributes may be expanded. The iterative process of training fair models may continue until a sufficient number of models are generated. In some examples, the iterative process may continue until a model with sufficient performance is determined. In other examples, the iterative process may be continued until a threshold for the statistical range is reached (e.g., a certain percentile of the confidence metric, a number of data points, or a percentage of the data being analyzed).

[0106] At block 510, one of the selected final models can be utilized in a production or other environment. The selected model can reduce algorithmic bias or be selected based on threshold criteria for bias or fairness. The selection of a final model can be used in a production environment to analyze data while minimizing bias. The selection of models can be determined based on, for example, a tradeoff between accuracy and expected fairness of the model. In some examples, the model can be based on the expected target database or use case.

[0107] As can be appreciated from the figures above, the final selected model is generated while using a limited or restricted amount of data containing protected attributes while generating a low-biased model.

VII. EXAMPLE EXPERIMENTAL SETUP AND RESULTS

[0108] Benefits of the disclosed technology may be better appreciated with reference to example results. The advantages disclosed below are for illustrative purposes and are not to be construed to be limited to the datasets discussed. The benefits are further below with respect to FIGS. 6 and 7.

[0109] The example respect to experimental results were derived from data of two publicly available datasets: (1) the Adult Income dataset and (2) the Default of Credit Card Clients (DCCC) dataset. These datasets and related tasks are further described below.

[0110] The first dataset is the Adult Income dataset. The dataset is derived from US census data. For this dataset, a target task for training a machine learning model is a binary classification problem to predict if an individual's income exceeds \$50,000 a year. The dataset contains features such as gender, race, marital status, capital gains, etc. For this particular example dataset, gender was chosen as a protected attribute while predicting the income level.

[0111] For the DCCC dataset, the task a binary classification problem, that is whether a user will default on their credit card payment or not. The dataset includes 23 attributes such as age, payment history, bill amount, and previous payment amount. In this dataset, age was considered to be a protected attribute. Users aged under 25 were one group of the protected attribute and those aged over 25 were the other group.

A. Summary of Results

[0112] FIG. 6 illustrates example results on both the adult income dataset and the DCC dataset. Illustrated in FIG. 6 is table 610. Table 610 illustrates the results obtained from the two datasets (the Adult Income Dataset and the DCCC dataset) with and without proxies included for a baseline model, a disparate income removal model, a reweighting model, and models generated using the methods described herein (labeled disclosed with k_{ed} and disclosed without k_{ed}).

[0113] As can be seen from Table 610, for both datasets, the inclusion of proxies within the features studied led to higher precision but poorer group fairness as compared to removing features considered proxies, which are further discussed below. Table 610 thus illustrates a correlation between proxy features and target tasks classifications. Thus, removing proxy features may be beneficial in enhancing model fairness. A trend in Table 610 is the improvement in terms of disparate impact (DI) fairness metric from the baseline to (i) a disparate impact remover (DIR) and reweighting technique and (ii) to the disclosed method. Additionally, the complex models such as RF and XGB yield higher precision but poorer fairness compared to LR models.

[0114] Further, the disclosed technology may lead to improvement in fairness for both the “w. proxy” and “w/o” proxy settings, which are further explained below. These improvements may be obtained due to the estimation of protected attributes from the first stage (upstream stage). Additionally, the DI metric obtained using DIR with the disclosed technology is better than the results obtained using DIR with the original PAs. This improvement can be attributed to the disclosed technology’s ability to select a smaller number of high-quality examples for training as compared to using the entire original training dataset, indicating the potential of the disclosed technology to enhance the effectiveness of DIR.

B. Details of Experimental Setup

[0115] Additional detail regarding the experimental setup related to table 610 is further provided herein. In more detail, the datasets referenced in FIG. 6 were divided into a training, validation, and test dataset. The results illustrated in FIG. 6 are the results for a test dataset. Following the tuning of hyperparameters for both first stage and second stage models on a validation set of data, the tuned models were used and the results reported were on the results from the test set.

[0116] As referred to in FIG. 6, in terms of feature selection, two different feature sets may be considered. The first set, referred to as “with proxies (w. proxies),” included attributes such as relationship status and marital status that could serve as proxies for the protected attributes (PAs). These proxy features were combined with regular features, which are dependent variables relevant for predicting the target downstream task. The second set, referred to as “without proxies (w/o proxies),” solely consisted of regular features. For the

Adult Income dataset, the feature list included regular features such as age, education, capital gain, capital loss, hours-per-week, workclass, and occupation, along with proxy features such as marital status, relationship, and native country. For the DCCC dataset, the regular features included limit on balance, payment, bill amount, and payment amount, while the proxy feature was education.

[0117] As further illustrated in FIG. 6, a “baseline” may be created against which to compare the results from the disclosed technique models. A baseline may be developed without incorporating fairness considerations in the model development. Two existing fairness preprocessing techniques: DIR (disparate impact remover) and reweighting may also be considered for metrics against which to compare the results. For both DIR and reweighting, the entire training data, including the original PAs, may be used as input. The disclosed technique, in contrast, may leverage estimated PAs and may only require only a smaller subset of the training data that includes the original PAs.

[0118] FIG. 6 also illustrates logistic regression (LR), Random Forest (RF), and XGBoost (XGB) techniques which may be used as part of the classification model used for a first stage within an active learning framework. For instance, a model may be initiated with a seed dataset (labeled D_l), which may include 0.1% of the training examples. Through active learning, new examples may be sampled from the remaining pool of instances D_u . For instance, the sampling may continue until the , which constituted 10% of the original training set. Additionally, various criteria and hyperparameters may be selected. In some examples, such as those illustrated with respect to FIG. 6, hyperparameters may be selected as follows selected as follows. For LR “balanced” class weight, $L1$ & $L2$ penalty, specified a maximum 1,000 number of iterations for the solvers to converge and set the inverse of regularization strength to 0.1 may be used. For RF, the number of trees may be set to 30, a maximum depth to 20, and used the Gini criterion. For XGB logistic regression as the criterion and set the maximum depth to 6 may be used. These hyperparameters are exemplary.

[0119] In some examples, due to the issue of class imbalances in the datasets, precision and F1 scores may be used as evaluation metrics to report classification performance. To evaluate fairness, a disparate impact ratio (DI) may be used. For instance, DI values within

the range of [0.8, 1.2] may be considered fair, with 1 being the best possible outcome (e.g., being entirely fair).

[0120] In some examples, to emphasize on fairness evaluation, “relaxed” versions of group fairness metrics may be considered, such as equalized odds difference (EOD) and equal of opportunity (EOP). A lower number may be better for EOD and EOP, with 0 being the perfect fairness. To select the best upstream model, performances may be monitored on the validation set throughout the iterations of the upstream model. In some examples, and as previously explained above, a “final” model may be selected based on precision and DI from iterations of training models. To achieve a final model (e.g., in the downstream stage), candidate models may be ranked first by precision in ascending order, followed by $\Delta_1 = |1 - DI|$ in descending order, indicating the extent to which DI deviated from the best possible value.

C. Additional Results from Ablation Study

[0121] FIG. 7 illustrates an ablation study. An ablation study investigates the performance of a machine learning system by removing certain components (e.g., features) to understand the contribution of the component (e.g., feature) to the overall system. FIG. 7 illustrates the number of data instances (e.g., data points) required to train the second stage (downstream model) both with (solid line) and without (dotted line) using the exponential decay equation for K_d discussed above. Graph 710 illustrates the study with the use of proxies while graph 720 illustrates the study without the use of proxies. In both graph 710 and 720, the x-axis (horizontal axis) represents the number of iterations while the y-axis (vertical axis) represents the number of selected instances. Overall, FIG. 7 illustrates that employing the exponential decay equation for K_d results in the selection of a smaller number of examples while attaining similar, or better results, compared to the baseline, DIR, and reweighting approaches, particularly for models such as RF and XGB.

D. Additional Results

[0122] FIG. 8 illustrates additional charts 810, 820, and 830 which respectively illustrate improvements in three different machine learning models, namely, logistic regression, random forest, and XGBoost, as compared to a baseline measurement. In FIG. 8, for all charts, both the precision change and the disparate impact ratio improvement, a higher

number reflects an improvement. As can be seen from FIG. 8, an improvement in the disparate impact ratio is observed across all machine learning models for both datasets. Further, the precision change is a small decrease in most cases. Thus, the charts illustrate that the disclosed technology allows for an improvement in disparate impact for only a small change.

E. Non-Limiting Summary of Advantages

[0123] A summary of the advantages of the disclosed technology are provided below.

[0124] The inclusion of proxy features may improve fairness. However, fairness (e.g., group fairness) was decreased during such inclusion. Thus, a correlation between proxy features and the target task is suggested. Model fairness may be enhanced by removing proxy features. The disclosed technology consistently improves fairness in both settings (with proxies and without proxies). As one specific example, with the XGB model on the adult income dataset, there were substantial improvements in fairness compared to other methods (baseline, DIR, reweighting), while precision remained competitive.

[0125] Another advantage was a trend demonstrating that as the model addresses fairness, from Baseline to DIR and reweighting and finally to the disclosed technology method, there is a general improvement in fairness as measured by disparate impact (DI).

[0126] Improvements were achieved using estimated PAs from an upstream model, proving the effectiveness of the PA estimation model. When the estimated PAs were used instead of original PAs for further data preprocessing in the downstream task model, better results were achieved. This suggests that the new framework is efficient as it selects fewer but higher quality examples.

VIII. PRIVACY OF INFORMATION

[0127] As described above, one aspect of the present technology is the gathering and use of publicly available information to preserve privacy and remove disparate impact with respect to protected attributes. Thus, the spirit of the invention is to promote fairness in models by removing disparate impact to individuals and groups of individuals by not relying on protected attributes, and proxies thereof, to make predictions. It is the spirit of the invention

to remove any information which causes disparate impact due to proxies (e.g., zip code) or protected attributes.

[0128] The present disclosure recognizes that the use of such personal information data, in the present technology, can be used beneficially to users. As one example, the personal information data can be used to predict financial instruments or products which may be of greater interest or utility to a user in accordance with their personal information data. As another example, particular offers which may aligned with a user's interest based on personal information data may be provided to the user.

[0129] The present disclosure contemplates that in some instances, this gathered data may include personal information data that uniquely identifies or can be used to identify a specific person. Such personal information data can include demographic data, location-based data, online identifiers, telephone numbers, email addresses, home addresses, date of birth, or any other personal information. However, such information is not intended to identify a specific individual but be generalized and anonymized for use.

[0130] Additionally, the present disclosure also contemplates embodiments in which users selectively block the use of, or access to, personal information data. For example, users may "opt out" of the collection of one or more pieces of personal information during the registration of services. In other examples, users may be informed that the provided information is anonymized and not specifically tied to the specific user for identification of that user. In yet other examples, information may be deleted after a specified period of time.

[0131] Moreover, it is the intent of the present disclosure that personal information data should be managed and handled in a way to minimize risks of unintentional or unauthorized access or use. Risk can be minimized by limiting the collection of data and deleting data once it is no longer needed. In addition, and when applicable, including in certain health related applications, data de-identification can be used to protect a user's privacy. De-identification may be facilitated, when appropriate, by removing identifiers, controlling the amount or specificity of data stored (e.g., collecting location data at city level rather than at an address level), controlling how data is stored (e.g., aggregating data across users), and/or other methods such as differential privacy.

[0132] The present disclosure contemplates that those entities responsible for the collection, analysis, disclosure, transfer, storage, or other use of such personal information

data will comply with well-established privacy policies and/or privacy practices, including all applicable laws and regulations. Non-limiting examples of policies and practices which may be used include federal and state laws, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States, the California Consumer Privacy Act (CCPA) in the State of California, or the General Data Protection Regulation (GDPR) applicable in Europe.

[0133] Additionally, it is the intent of the disclosure to limit the use of personal information. Thus, the privacy preserving and enhancing techniques, such as through not using proxies for certain protected attributes, and the non-use of protected attributes in making decisions is described.

IX. EXAMPLE COMPUTING SYSTEM AND TRAINING OF ML MODELS

A. Training of Machine Learning Models

[0134] The process of training a machine learning model can involve determining the set of parameters that achieve the “best” performance, often based on the output of a loss or error function. A loss function typically relates the expected or ideal performance of a machine learning model to its actual performance. For example, if there is a training data set comprising input data paired with corresponding labels (indicating, e.g., whether that input data corresponds to normal data or anomalous data), then a loss function or error function can relate or compare the classifications or labels produced by the machine learning model to the known labels corresponding to the training data set.

[0135] If the machine learning model produces labels that are identical to the labels associated with the training data, then the loss function may take on a small value (e.g., zero), while if the machine learning model produces labels that are totally inconsistent with the labels corresponding to the training data set, then the loss function may take on a larger value (e.g., one, a value greater than one, infinity, etc.). In some cases, the training process can involve iteratively updating the model parameters during training rounds, batches, epochs, or other suitable divisions of the training process.

[0136] In each round, a computer system can evaluate the current performance of the model for a particular set of parameters using the loss or error function. Then, the computer system can use metrics such as the gradient and techniques such as stochastic gradient

descent to update the model parameters based on the loss or error function. In summary terms, the computer system can predict which change to the model parameters result in the fastest decrease in the loss function, then change the model parameters based on that prediction. This process can be repeated in subsequent training rounds, epochs, etc.

[0137] The computer system can perform this iterative process until training is complete. In some cases, the training process involve a set number of training rounds or epochs. In other cases, training can proceed until the model “converges”, i.e., the model shows little to no further improvement in successive training rounds, or that the difference in the output of the error or loss function between successive training rounds approaches zero.

B. Example Computer System

[0138] Any of the computer systems mentioned herein may utilize any suitable number of subsystems. Examples of such subsystems are shown in FIG. 9 in computer system 900. In some embodiments, a computer system includes a single computer apparatus, where the subsystems can be the components of the computer apparatus. In other embodiments, a computer system can include multiple computer apparatuses, each being a subsystem, with internal components. A computer system can include desktop and laptop computers, tablets, mobile phones and other mobile devices.

[0139] The subsystems shown in FIG. 9 are interconnected via a system bus 75. Additional subsystems such as a printer 74, keyboard 78, storage device(s) 79, monitor 76 (e.g., a display screen, such as an LED), which is coupled to display adapter 82, and others are shown. Peripherals and input/output (I/O) devices, which couple to I/O controller 71, can be connected to the computer system by any number of means known in the art such as input/output (I/O) port 77 (e.g., USB, FireWire®). For example, I/O port 77 or external interface 81 (e.g., Ethernet, Wi-Fi, etc.) can be used to connect computer system 700 to a wide area network such as the Internet, a mouse input device, or a scanner. The interconnection via system bus 75 allows the central processor 406 to communicate with each subsystem and to control the execution of a plurality of instructions from system memory 72 or the storage device(s) 79 (e.g., a fixed disk, such as a hard drive, or optical disk), as well as the exchange of information between subsystems. The system memory 72 and/or the storage device(s) 79 may embody a computer readable medium. Another subsystem is a data collection device 85, such as a camera, microphone, accelerometer, and the like. Any of the

data mentioned herein can be output from one component to another component and/or can be output to a user.

[0140] A computer system can include a plurality of the same components or subsystems, e.g., connected together by external interface 81, by an internal interface, or via removable storage devices that can be connected and removed from one component to another component. In some embodiments, computer systems, subsystem, or apparatuses can communicate over a network. In such instances, one computer can be considered a client and another computer a server, where each can be part of a same computer system. A client and a server can each include multiple systems, subsystems, or components. In various embodiments, methods may involve various numbers of clients and/or servers, including at least 4, 20, 50, 40, 200, 500, 1,000, or 4,000 devices. Methods can include various numbers of communication messages between devices, including at least 40, 200, 500, 1,000, 4,000, 50,000, 40,000, 500,00, or one million communication messages. Such communications can involve at least 1 MB, 10 MB, 100 MB, 1 GB, 10 GB, or 100 GB of data.

[0141] Any of the computer systems mentioned herein may utilize any suitable number of subsystems. In some embodiments, a computer system includes a single computer apparatus, where the subsystems can be components of the computer apparatus. In other embodiments, a computer system can include multiple computer apparatuses, each being a subsystem, with internal components.

[0142] A computer system can include a plurality of the components or subsystems, e.g., connected together by external interface or by an internal interface. In some embodiments, computer systems, subsystems, or apparatuses can communicate over a network. In such instances, one computer can be considered a client and another computer a server, where each can be part of a same computer system. A client and a server can each include multiple systems, subsystems, or components.

[0143] It should be understood that any of the embodiments of the present disclosure can be implemented in the form of control logic using hardware (e.g., an application specific integrated circuit or field programmable gate array) and/or using computer software with a generally programmable processor in a modular or integrated manner. As used herein a processor includes a single-core processor, multi-core processor on a same integrated chip, or multiple processing units on a single circuit board or networked. Based on the disclosure and

teachings provided herein, a person of ordinary skill in the art will know and appreciate other ways and/or methods to implement embodiments of the present disclosure using hardware and a combination of hardware and software.

[0144] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C, C++, C#, Objective-C, Swift, or scripting language such as Perl or Python using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer readable medium for storage and/or transmission, suitable media include random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a compact disk (CD) or DVD (digital versatile disk), flash memory, and the like. The computer readable medium may be any combination of such storage or transmission devices.

[0145] Such programs may also be encoded and transmitted using carrier signals adapted for transmission via wired, optical, and/or wireless networks conforming to a variety of protocols, including the Internet. As such, a computer readable medium according to an embodiment of the present disclosure may be created using a data signal encoded with such programs. Computer readable media encoded with the program code may be packaged with a compatible device or provided separately from other devices (e.g., via Internet download). Any such computer readable medium may reside on or within a single computer product (e.g., a hard drive, a CD, or an entire computer system), and may be present on or within different computer products within a system or network. A computer system may include a monitor, printer, or other suitable display for providing any of the results mentioned herein to a user.

[0146] Any of the methods described herein may be totally or partially performed with a computer system including one or more processors, which can be configured to perform the steps. Thus, embodiments involve computer systems configured to perform the steps of any of the methods described herein, potentially with different components performing a respective steps or a respective group of steps. Although presented as numbered steps, steps of methods herein can be performed at a same time or in a different order. Additionally, portions of these steps may be used with portions of other steps from other methods. Also,

all or portions of a step may be optional. Additionally, and of the steps of any of the methods can be performed with modules, circuits, or other means for performing these steps.

WHAT IS CLAIMED IS:

1. A method of correcting algorithmic bias in a machine learning framework, the method comprising:
 - obtaining, a first dataset, the first dataset containing protected attributes from at least one category of protected attributes;
 - obtaining, a second dataset, the second dataset not containing any protected attributes from the at least one category of protected attributes;
 - training, using the first dataset, a protected attributes estimation model, wherein the protected attributes estimation model estimates information related to the least one protected attribute; and
 - predicting, using protected attributes estimation model, at least one protected attribute for data within the second dataset.
2. The method of claim 1 wherein the protected attributes estimation model is trained in iterations.
3. The method of claim 2 wherein each iteration involves an active learning process to improve the protected attributes estimation model at each iteration.
4. The method of claim 2 wherein additional datapoints containing protected attributes are added to the first dataset to update the training of the protected attributes.
5. The method of claim 4 wherein the additional datapoints are obtained by annotating points from the second dataset with a protected attribute.
6. The method of claim 4 wherein the additional datapoints are chosen from a ranked list of datapoints in the second dataset.
7. The method of claim 6 wherein the ranking is based on empirically defined bias.

8. The method of claim 1 wherein the protected attributes estimation model provides a confidence metric associated with the predicted protected attribute.
9. The method of claim 1 wherein an augmented dataset is generated based on augmenting the second dataset with predicted protected attributes from the protected attributes estimation model.
10. A method comprising:
 - obtaining, an augmented dataset, the augmented dataset comprising datapoints with protected attributes predicted from an estimation model;
 - training a plurality of bias mitigation models based on subsets of the augmented dataset;
 - obtaining a fairness metric and an accuracy metric for each of the plurality of the bias mitigation models; and
 - choosing one model from the plurality of bias mitigation models based on at least one of the fairness metric and the accuracy metric.
11. The method of claim 10 wherein the subsets of the augmented dataset are generated based on statistical information of the protected attributes of the augmented dataset.
12. The method of claim 11 wherein the statistical information is generated by the estimation model.
13. The method of claim 12 wherein the statistical information is a confidence metric associated with a predicted protected attributed.
14. The method of claim 13 wherein the subsets of the augmented dataset are generated based on ranges of confidence metrics.
15. The method of claim 10 wherein the subsets of the augmented dataset are generated based on statistical information of the protected attributes of the augmented dataset.

16. A method comprising:
 - (A) performing a first stage, the first stage comprising:
 - obtaining, a protected attributes (PA) known dataset and a PA unknown dataset;
 - training a bias estimation model based on datapoints from the PA known dataset;
 - generating, estimates for protected attributes based on the trained bias estimation model;
 - augmenting the PA unknown dataset to generate an augmented dataset;
 - (B) performing a second stage, the second stage comprising:
 - obtaining the augmented dataset;
 - generating a plurality of subsets of the augmented dataset;
 - training a plurality of bias mitigation models based on the plurality of subsets of the augmented dataset; and
 - choosing one of the trained plurality of bias mitigation models as a final mitigation model.
17. The method of claim 16 wherein additional datapoints are added to the PA known dataset through an active learning process.
18. The method of claim 17 wherein the active learning process is performed by annotating datapoints in the PA unknown dataset with protected attributes.
19. The method of claim 17 further comprising using a ranked entropy model to select additional datapoints during training in the second stage.
20. A system comprising one or more processor coupled to a computer readable medium, the computer readable medium comprising instructions, that when executed by the one or more processors perform a method of any of claims 1–19.