Three Proposals for Fairness in Speech Recognition

Mark Hasegawa-Johnson and Heting Gao

August 6, 2020

Outline

Speech Recognition

Speech recognition is the task of converting $\boldsymbol{X}$ to $\boldsymbol{Y}$, where

- $\boldsymbol{X} = [X_1, \ldots, X_T]$ is a sequence of random spectra whose instance value is $\mathbf{x} = [x_1, \ldots, x_T]$, $x_t \in \mathcal{X}$.
- $\boldsymbol{Y} = [Y_1, \ldots, Y_S]$ is a sequence of phones, words, or characters whose instance value is $\mathbf{y} = [y_1, \ldots, y_S]$, $y_s \in \mathcal{Y}$, $S \leq T$.

The problem we have: speech recognition is less accurate for people with high-pitched voices (women), less accurate for people with dialects that it has not seen during training (African American English, Hispanic English, Indian English, Bronx, Cockney, ...), and less accurate for people with some types of disabilities (e.g., Cerebral Palsy, Parkinson's Disease, ALS).

Data Sources

- AA (CORAAL): `https://oraal.uoregon.edu/coraal`
- AF (AST Afrikaans English) `https://vlo.clarin.eu/record/https_58__47__47_hdl.handle.net_47_20.500.12185_47_411_64_format_61_cmdi?2`
- AM (Librispeech): `http://www.openslr.org/12`
- BR (Cambridge Read News): LDC95S24
- IN (maheshchandra-20160719-cgc, maheshchandra-20160719-com, maheshchandra-20160719-e01) `http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/Audio/Main/8kHz_16bit`
- SP (Hisp-eng): LDC2014S05
- XH (AST Black English): `https://repo.sadilar.org/handle/20.500.12185/433`
- UA: `http://ifp-08.ifp.uiuc.edu/protected/UASPEECH` (Password: please ask Mark. Participants gave permission for research use, but not commercial use or redistribution.)

Experiments

Gao, 2020

- AA for African American, AF for Afrikaans, AM for American, BR for British, IN for Indian, SP for Spanish in central-south America and XH for Xhosa-Zulu-Sotho.

	Base20000	**Base3000**		**BaseIndv3000**		
Dialect	CTCLoss	CTCLoss	CER	Train Size	CTCLoss	CER
AA	2.43	2.47	67.71%	3000	2.45	68.91%
AF	1.69	1.70	49.66%	3000	1.14	26.87%
AM	1.61	1.62	49.34%	3000	1.39	41.43%
BR	1.79	1.83	54.07%	3000	1.62	47.96%
IN	2.28	2.40	61.16%	300	4.05	93.11%
SP	2.08	2.07	56.86%	200	2.86	79.99%
XH	1.94	2.00	55.78%	2619	1.75	46.83%

- Difficulty varies across different dialect (AA)
- Training on Individual Language can reduce the error rate (AF, AM, BR, XH).
- Sample size affect accuracy (IN)

Fairness in AI: General Definitions

Our goal is to estimate Y given X, where X is stuff we know (e.g., employment history) and Y is something we'd like to know (e.g., probability you'll default on your mortgage). We want to find $\hat{Y}(X)$, a neural net, that is "fair" to people with different values of some protected attribute A (e.g., gender, race, dialect, disability). What does "fair" mean?

1. **Demographic Parity**

$$P(\hat{Y} = y|A = a) = P(\hat{Y} = y|A = a') \quad \forall y, a, a'$$

2. **Equal Opportunity**

$$P(\hat{Y} = y|A = a, Y = y) = P(\hat{Y} = y|A = a', Y = y) \quad \forall y, a, a'$$

3. **Counterfactual Fairness**

$$P(\hat{Y}_{A\leftarrow a}(U) = y|Y = y, A = a) = P(\hat{Y}_{A\leftarrow a'}(U) = y|Y = y, A = a)$$

Demographic Parity

$$P(\hat{Y} = y|A = a) = P(\hat{Y} = y|A = a') \quad \forall y, a, a'$$

- Pro:
  - Easy to compute.
  - Useful if the ground truth labels, $Y = y$, depend on A in a way that is historically unfair, and you don't want to reproduce that unfairness.
- Con:
  - Useless if demographic parity is a socially undesirable outcome.
  - Population measure, not an individual measure: Doesn't measure the degree to which any particular individual is treated fairly.

Demographic Parity Fails for Speech

$$P(\hat{Y} = y|A = a) \neq P(\hat{Y} = y|A = a') \quad \forall y, a, a'$$

- DP is inappropriate for speech. Blacks and whites/men and women/Irish and English people say different things. We don't want an ASR that forces them to have the same posterior probabilities.
- This fails even if you apply it to individual phonemes. Different dialects use different words (and therefore different characters), and use different phonemes even to say the same words.

Equal Opportunity

$$P(\hat{Y} = y|A = a, Y = y) = P(\hat{Y} = y|A = a', Y = y) \quad \forall y, a, a'$$

- Pro:
  - Easy to compute.
  - Useful if we trust the ground truth labels, $Y = y$, and we want our algorithm to reproduce them with equal accuracy for all demographic groups.
- Con:
  - Useless if the dependence of Y on A is already unfair.
  - Population measure, not an individual measure: Doesn't measure the degree to which any particular individual is treated fairly.

Equal Opportunity Works for Speech

$$P(\hat{Y} = y|A = a, Y = y) \neq P(\hat{Y} = y|A = a', Y = y) \quad \forall y, a, a'$$

- This is exactly what we want: equal accuracy for all values of A.
- It's still a population measure: it doesn't measure the accuracy for any given individual in a population.

The "Death in Childbirth" Clause

- In 1900, almost 1% of all women died in childbirth. 0% of men died in childbirth. The solution was not to increase the number of men who died during childbirth; the solution was to decrease the number of women who died that way.
- Similarly, in the pursuit of fairness, if we ever develop a loss function that says "increase fairness by decreasing the error rate for group A, and increasing the error rate for group B," then we should just delete the second term in the loss function.
- Treat it as a minimax problem (minimize the maximum of two error rates).

Outline

Connectionist Temporal Classification

Graves, Fernández, Gomez & Schmidhuber, ICML 2016

- $\theta =$ model parameters.
- **Spectrogram:** $\mathbf{X} = [X_1, \ldots, X_T] =$ sequence of random spectra whose instance value is $\mathbf{x} = [x_1, \ldots, x_T]$, $x_t \in \mathcal{X}$.
- **Transcription:** $\mathbf{Y} = [Y_1, \ldots, Y_S] =$ sequence of characters whose instance value is $\mathbf{y} = [y_1, \ldots, y_S]$, $y_s \in \mathcal{Y}$, $S \leq T$.
- **Time-Aligned Transcription:** $\mathbf{\Pi} = [\Pi_1, \ldots, \Pi_T] =$sequence of time-aligned characters whose instance value is $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_T]$, where each time-aligned character is either one of the transcription characters ($\pi_t = y_s$ for some s), or else $\pi_t = \varnothing$. $\varnothing$ is a special "blank" character. For example, suppose an $S = 5$-character text ("hello") is encoded in a $T = 14$-frame speech waveform, it might be:

$$\boldsymbol{y} = [h, e, l, l, o], \quad \boldsymbol{\pi} = [h, h, e, e, e, \varnothing, \varnothing, l, l, l, \varnothing, l, \varnothing, o]$$

The time-compression function

In order to compress $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_T]$ down to the output length $\boldsymbol{y} = [y_1, \ldots, y_S]$, for $S \leq T$, we need a time-compression function:

- $\mathcal{B} : (\mathcal{Y} \cup \{\varnothing\})^+ \rightarrow \mathcal{Y}^+$ is a surjective function:
  - For any valid $\boldsymbol{\pi}$, $\mathcal{B}(\boldsymbol{\pi})$ is a unique $\boldsymbol{y}$.
  - For any valid $\boldsymbol{y}$, $\mathcal{B}^{-1}(\boldsymbol{y})$ is a set of possible values of $\boldsymbol{\pi}$.

The most common $\mathcal{B}$-function (the one invented by Graves et al.) does this:

Start with a length-T sequence:	$\boldsymbol{\pi}$ =hheee$\varnothing\varnothing$lll$\varnothing$l$\varnothing$o
(1) eliminate duplicates:	he$\varnothing$l$\varnothing$l$\varnothing$o
(2) eliminate blanks:	$\boldsymbol{y}$ =hello

The CTC loss function

A CTC-based speech recognizer is trained to minimize the CTC loss. The CTC loss is the cross-entropy (negative log probability) of the correct label sequence:

$$\mathcal{L}_{CTC}(\theta) = -\ln P(\hat{\boldsymbol{Y}} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})$$

$$= -\ln \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y})} \prod_{t=1}^{T} q_t(\pi_t)$$

$$= -\text{logsumexp}_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y})} \sum_{t=1}^{T} \ln q_t(\pi_t)$$

where $q_t(\pi_t)$ is the output of a softmax layer at time t. The input of this softmax layer is a bLSTM, Transformer, or some other neural net parameterized by θ and having access to the whole sequence $\boldsymbol{x}$, so

$$q_t(\pi) = P(\hat{\Pi}_t = \pi | \boldsymbol{x}, \theta)$$

How to compute CTC loss: turn it into EM

Taking advantage of $\frac{d}{d\theta} \ln f(\theta) = \frac{1}{f(\theta)} \frac{df}{d\theta}$, you can prove that

$$\nabla_\theta \mathcal{L}_{CTC}(\theta) = \nabla_\theta \mathcal{L}_{EM}(\theta, \bar{\theta})$$

where $\mathcal{L}_{EM}$ is the weighted average, over all frames t and over all characters y, of $-\ln q_t(y)$:

$$\mathcal{L}_{EM}(\theta, \bar{\theta}) = -\sum_y \sum_t \gamma_t(y) \ln q_t(y)$$

The weighting function is the alignment posterior, defined as

$$\gamma_t(y) = P(\Pi_t = y | \boldsymbol{x}, \boldsymbol{y}, \theta) = \frac{\sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y}), \pi_t = y} \exp\left(\sum_{t'=1}^{T} \ln q_{t'}(\pi_{t'})\right)}{\sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\boldsymbol{y})} \exp\left(\sum_{t'=1}^{T} \ln q_{t'}(\pi_{t'})\right)}$$

(Remember that $q_t(y)$ is the recognition posterior:)

$$q_t(y) = P(\Pi_t = y | \boldsymbol{x}, \theta)$$

Outline

Deep F-measure Maximization for End-to-End Speech Understanding

Sarı& Hasegawa-Johnson, Interspeech 2020

Maximum-accuracy training biases a DNN toward the majority class. Reduced bias is achieved by trying to optimize the harmonic average of precision and recall, averaged across all classes.

- N_k=true number of tokens in class k
- $TP(k)$ = true positive detections of class k
- $FP(k)$ = false positive detections of class k
- $\text{Rec}(k) = TP(k)/N_k$ = recall of class k
- $\text{Prec}(k) = TP(k)/(TP(k) + FP(k))$ = precision of class k
- $F_\beta(k) = \frac{(1+\beta^2)\text{Rec}(k)\text{Prec}(k)}{\beta^2\text{Prec(k)}+\text{Rec}(k)}$ = β-weighted harmonic average of precision and recall for class k.
- Macro-averaged F_β measure:

$$F_\beta = \frac{1}{K}\sum_{k=1}^{K}\frac{(1+\beta^2)TP(k)}{\beta^2 N_k + (TP(k)+FP(k))}$$

Deep F-measure Maximization for End-to-End Speech Understanding

Sarı& Hasegawa-Johnson, Interspeech 2020

F_β requires counting the number of true positives and false positives; this is not a differentiable operation. A differentiable approximation can be achieved as follows:

$$TP(k) \approx \sum_{n \in S_k} q'_n(k), \quad FP(k) \approx \sum_{n \notin S_k} q'_n(k)$$

where S_k is the set of training tokens from class k, S is the total training dataset, and $q'_n(k)$ is the renormalized softmax output of the classifier,

$$q'_n(k) = \frac{q_n(k)}{\max_l q_n(l)}$$

Therefore

$$\mathcal{L} = -\frac{1}{K} \sum_{k=1}^{K} \frac{(1+\beta^2) \sum_{n \in S_k} q'_n(k)}{\beta^2 N_k + \sum_{n \in S} q'_n(k)}$$

Soft Counts for Equal Opportunity Training

Mina, 2020

An Equal-Opportunity AI is one that satisfies

$$\left| P(\hat{Y} = y | A = a, Y = y) - P(\hat{Y} = y | A = a', Y = y) \right| = 0 \ \forall y, a, a'$$

Rusty Mina proposed estimating the probabilities using Deep-F-style soft counts:

$$P(\hat{Y} = y | A = a, Y = y) = \frac{\sum_{n \in S_{y,a}} q'_n(y)}{|S_{y,a}|}$$

where $S_{y,a}$ is the data subset with labels $Y = y, A = a$ and $|S_{y,a}|$ is its cardinality. Equal Opportunity is achieved by setting the following loss function to zero:

$$\mathcal{L}_{EO} = \sum_{y} \sum_{a,a'} \left| \frac{\sum_{n \in S(y,a)} q'_n(y)}{|S_{y,a}|} - \frac{\sum_{n \in S(y,a')} q'_n(y)}{|S_{y,a'}|} \right|$$

Soft-Count Equal Opportunity for Speech

$$\mathcal{L}_{EO} = \sum_{y} \sum_{a,a'} \left| \frac{\sum_{n \in S(y,a)} q'_n(y)}{|S_{y,a}|} - \frac{\sum_{n \in S(y,a')} q'_n(y)}{|S_{y,a'}|} \right|$$

In speech, $\boldsymbol{y} = [y_1, \ldots, y_S]$. Here are three main possibilities:

1. **Matched Transcriptions:** $S(\boldsymbol{y}, a)$, could be defined as "the set of waveforms that have exactly the same transcription." Only works for UASPEECH and TIMIT.
2. **Matched Frames**: $S(y, a)$ could be defined as "the set of frames for which the recognizer should output character y." The problem is, we have no time alignment, so we don't know which frames those are.
3. **Equal Opportunity Accuracy:** In speech, the metric $P(\hat{Y}|Y)$ is just sentence accuracy. We could demand that accuracy is the same for different demographic groups, even if they never say exactly the same things.

Equal Opportunity Accuracy

Sentence accuracy is normally defined, in ASR, as the probability that the recognizer output transcription $\hat{\boldsymbol{Y}}$ exactly matches the correct transcription $\boldsymbol{y}$, averaged over all transcribed utterances $(\boldsymbol{x}, \boldsymbol{y})$ in the training dataset. Sentence accuracy is equal, for every pair of demographic attributes (a, a'), if the following loss function is zero:

$$\mathcal{L}_{EOA} = \sum_{a,a'} \left| \frac{1}{|S_a|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in S_a} P(\hat{\boldsymbol{Y}} = \boldsymbol{y}|\boldsymbol{x}) - \frac{1}{|S_{a'}|} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in S_{a'}} P(\hat{\boldsymbol{Y}} = \boldsymbol{y}|\boldsymbol{x}) \right|$$

where $S_a = \{(\boldsymbol{x}, \boldsymbol{y})|A = a\}$.

The "Death in Childbirth" Clause

As defined on the previous slide, $\mathcal{L}_{EOA}$ has a "death in childbirth" problem: it specifically encourages the recognizer to **increase the error rate** of the more accurately-recognized demographic. Instead, let's modify it to focus exclusively on **decreasing the error rate** of the less-accurately recognized demographic:

$$\mathcal{L}_{EOA} = \sum_{a_1, a_2 \sim A} \max_{a \in \{a_1, a_2\}} \left(-\frac{1}{|S_a|} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in S_a} P(\hat{\boldsymbol{Y}} = \boldsymbol{y} | \boldsymbol{x}) \right)$$

Integration with CTC

$\mathcal{L}_{EOA}$ is now just a weighted form of $\mathcal{L}_{CTC}$: it gives more weight to $(\boldsymbol{x}, \boldsymbol{y})$ pairs that come from a demographic group $A = a$ that currently has worse error rates. Just like CTC, $\nabla_\theta \mathcal{L}_{EOA}(\theta) = \nabla_\theta \mathcal{L}_{WEM}(\theta, \bar{\theta})$, where $\mathcal{L}_{WEM}$ is a demographically-weighted EM measure defined as:

$$\mathcal{L}_{WEM}(\theta, \bar{\theta}) = \sum_{a_1, a_2 \sim A} \max_{a \in \{a_1, a_2\}} \left(-\frac{1}{|S_a|} \sum_y \sum_{t \in S_a} \gamma_t(y) \ln q_t(y) \right)$$

where γ and q should (I think) have the demographic features as inputs:

$$q_t(y) = P(\Pi_i = y | \mathbf{x}, a, \theta)$$

$$\gamma_t(y) = P(\Pi_i = y | \mathbf{x}, \mathbf{y}, a, \bar{\theta})$$

Training Procedure

1. Maybe start with a few epochs of standard CTC:

$$\mathcal{L}_{EM}(\theta, \bar{\theta}) = -\sum_{y}\sum_{t} \gamma_t(y) \ln q_t(y)$$

2. Then assigning weights:

$$\mathcal{L}_{WEM}(\theta, \bar{\theta}) = \sum_{a_1, a_2 \sim A} \max_{a \in \{a_1, a_2\}} \left(-\frac{1}{|S_a|} \sum_{y}\sum_{t \in S_a} \gamma_t(y) \ln q_t(y) \right)$$

$$= \sum_{a \sim A} \left(-\frac{N_{\leq a}}{|S_a|} \sum_{y}\sum_{t \in S_a} \gamma_t(y) \ln q_t(y) \right)$$

where $N_{\leq a} = \#$ other groups that have lower loss than group a. I think that loss measure should actually be pretty robust, and can probably run until convergence.

Outline

Counterfactual Fairness

(Kusner, Loftus, Russell & Silva, NIPS 2017)

$$P(\hat{Y}_{A\leftarrow a}(U) = y | X = x, A = a) = P(\hat{Y}_{A\leftarrow a'}(U) = y | X = x, A = a)$$

- Pro:
  - Useful regardless of whether or not the dependence of Y on A is historically biased.
  - Characterizes the individual by protected attribute $A = a$, other observable attributes $X = x$, and underlying latent attributes $U = u$. Demands fairness for every individual: demands that $\hat{Y}$ be unchanged, even if A were changed to a', keeping all other attributes (even the unknown latent attributes $U = u$) the same.
- Con:
  - In speech recognition (at least), it seems to have many different possible interpretations.

How to compute a "counterfactual" posterior:

$$P(\hat{Y}_{A\leftarrow a}(U) = y|Y = y, A = a) = P(\hat{Y}_{A\leftarrow a'}(U) = y|Y = y, A = a)$$

1. **Abduction:** Assume latent variables U with some prior $P(U)$. Given each training datum $(X = x, Y = y, A = x)$, compute the posterior $P(U|X = x, Y = y, A = a)$.
2. **Action:** Set A to a' (don't recompute any probabilities yet).
3. **Prediction:** Compute

$$P(\hat{Y}_{A\leftarrow a'}(U) = y|Y = y, A = a) =$$

$$\sum_u P(U = u|X = x, Y = y, A = a)P(\hat{Y} = y|X = x, U = u, A = a')$$

Levels of increasing modeling power

The following types of models are counterfactually fair. (Kusner et al.) claim that models at higher levels of the following ontology can predict Y with better precision than lower-level models:

1. Build $\hat{Y}$ using only observable non-descendants of A.
   Problem: in real data, most observables are influenced by A.
2. Postulate latent variables with some prior $P(U)$, estimate $P(X|A, U)$ from the data, then abduct $P(U|X = x, Y = y, A = a)$ for each training example.
3. Postulate a fully deterministic model with latent "residual" variables $e \sim E$ such that
   1. $x = f(a, e)$ is a deterministic function, e.g., linear regression $x = wa + b + e$ with constant weight w and bias b, and
   2. E is independent of A.

   Then train $\hat{Y} = g(e)$, independent of A by design.

Counterfactual Fairness and Speech

- Level-3 counterfactual fairness is inappropriate for speech, for the same reason that DP is inappropriate: We don't want $\hat{Y} = g(e)$ for E independent of A, because we don't want $\hat{Y}$ to be independent of A (blacks and whites say different things; we want to preserve that difference, not ignore it).
- Level-2 can model this. Assume $P(U)$, then
  1. learn $P(X|A, Y, U)$,
  2. abduct $P(U|X = x, Y = y, A = a)$ for each datum,
  3. predict $P(\hat{Y}|U = u, X = x, A = a')$.

  The problem is how to design U so that it contains all the information that X carries about Y, including any information about A that is embedded in the relationship between X and Y, but none of the extra information about A that might be in X but absent in Y.

Counterfactual fairness and speech

The problem is how to design U so that:

- it contains all the information that X carries about Y, including any information about A that is embedded in the relationship between X and Y. In terms of mutual information, we want $I(X, Y) = I(U, Y)$. If U is a deterministic function of X, then there is a theorem that says $I(X, Y) \geq I(U, Y)$ always, so our goal is to make $I(U, Y)$ as large as possible.
- it contains none of the extra information about A that might be in X but absent in Y. In terms of mutual information, we want $I(Y, (U, A)) = I(Y, A)$. There is a theorem that says $I(Y, (U, A)) \geq I(Y, A)$ always, so our goal is to make $I(Y, (U, A))$ as small as possible.

Counterfactual Fairness based on Mutual Information

Suppose we constrain $U = g(X, A)$ to be a deterministic function of X and A (so that it can be computed during test time). Then

- We want $I(U, Y)$ as large as possible:

$$U = \arg\max I(U, Y) = \arg\max H(Y) - H(Y|U)$$

- We want $I((U, A), Y)$ as small as possible:

$$U = \arg\min I((U, A), Y) = \arg\min H(Y) - H(Y|U, A)$$

Thus $U = g(X, A)$ is trained to minimize

$$\mathcal{L}_{CF} = H(Y|U) - H(Y|U, A)$$

Counterfactual Fairness based on Mutual Information

$\boldsymbol{U} = g(\boldsymbol{X}, A)$ is trained to minimize

$$\mathcal{L}_{CF} = H(\boldsymbol{Y}|\boldsymbol{U}) - H(\boldsymbol{Y}|\boldsymbol{U}, A)$$

Putting that into the CTC framework, we want $U = g(X, A)$ trained to minimize

$$\mathcal{L}_{CFMI} = -\sum_{y}\sum_{t} \gamma_t(y) \left(\ln q_t(y|\mathbf{u}) - \ln q_t(y|\mathbf{u}, a)\right)$$

where

- $\boldsymbol{u}(\boldsymbol{x}, a)$ is a bLSTM layer.
- $q_t(y|\boldsymbol{u})$ and $q_t(y|\boldsymbol{u}, a)$ are two separate branches after the layer $\boldsymbol{u}$, each separately pre-trained to optimize $\mathcal{L}_{CTC}$.
- After $q_t(y|\boldsymbol{u})$ and $q_t(y|\boldsymbol{u}, a)$ have been pre-trained, then we fix them, and re-train $\boldsymbol{u}(\boldsymbol{x}, a)$ in order to minimize

$$(1-\lambda)\mathcal{L}_{EM}+\lambda\mathcal{L}_{CFMI} = -\sum_{y}\sum_{t} \gamma_t(y) \left(\ln q_t(y|\mathbf{u}) - \lambda \ln q_t(y|\mathbf{u}, a)\right)$$

Outline

Synthetic minority oversampling technique

Chawla, Boyle, Hall & Kegelmeyer, JAIR 2002

SMOTE is a data augmentation trick. I'm using it here to exemplify all possible data augmentation and active learning methods. It usually outperforms most other methods.
SMOTE generates artificial data for the minority class by iterating the following process, until all classes have equal numbers of samples:

1. Choose, at random, two samples from the minority class, x_1 and x_2.
2. Choose a number λ uniformly at random between 0 and 1.
3. Generate a new training sample as $\lambda x_1 + (1 - \lambda)x_2$.

SMOTE for Time-Series Data?

- SMOTE was proposed for vector data. It has rarely been used for time-series data, and never (as far as I know) to generate synthetic data for ASR. The "curse of dimensionality" makes the process $\lambda x_1 + (1 - \lambda)x_2$ unreliable for generating realistic training examples.
- Zhu, Lin & Liu (`https://arxiv.org/pdf/2004.06373.pdf`, 2020) propose OHIT (Oversampling High-dimensional Imbalanced Time-series), which uses robust covariance estimators to cluster the minority class samples prior to SMOTE.
- Speech would need one more step: use dynamic time warping (DTW) to time-align the two examples, prior to SMOTE. As far as I know, it has never been tried, and should be tried.

SMOTE for ASR?

For example, SMOTE for ASR could do this:

1. Choose, at random, two samples from the minority class, x_1 and x_2, with the same text $Y = y$. Use dynamic time warping to align them.
2. Choose a number λ uniformly at random between 0 and 1.
3. Generate a new training sample as $\lambda x_1 + (1 - \lambda)x_2$ (interpolate the MFCCs, not the spectra).

SMOTE and EO

Equal opportunity says that we want

$$P(\hat{Y} = y|A = a, Y = y) = P(\hat{Y} = y|A = a', Y = y) \quad \forall y, a, a'$$

So instead of choosing utterances x_1 and x_2 uniformly at random, we could choose them with a probability proportional to

$$P(\text{choose } y|a) \propto \max\left(0, \max_{a' \neq a} P(y|a') - P(y|a)\right)$$

Outline

Conclusions

I've sketched four possible methods to make ASR more fair:

1. Equal Opportunity Accuracy: results in a one-part loss function, a weighted average of CTC losses in different demographics.
2. Counterfactual Fairness based on Mutual Information: results in a two-part loss function, very similar to adversarial training, but with a different definition of the adversary.
3. Data Augmentation for Equal Opportunity: create augmented data based on the samples that are treated least fairly by the ASR.