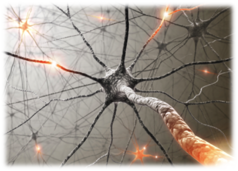
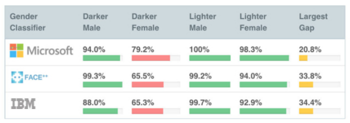
Machine learning: group DRO



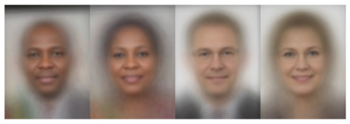
- Thus far, we have focused on finding predictors that minimize the training loss, which is an average (of the loss) over the training examples.
- While averaging seems reasonable, in this module, I'll show that averaging can be problematic and lead to inequalities in accuracy across groups.
- Then I'll briefly present an approach called **group distributional robust optimization (group DRO)**, which can mitigate some of these inequalities.

[Buolamwini & Gebru 2018]

Gender Shades



Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

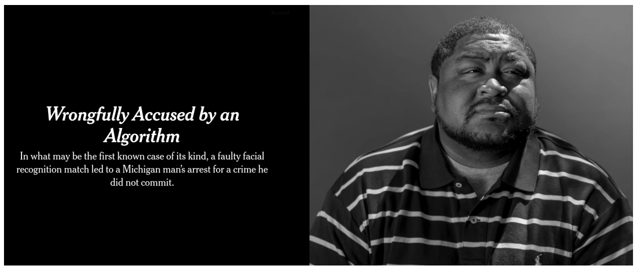


Inequalities arise in machine learning

- is the Gender Shades project by Joy Buolamwini and Timnit Gebru (paper).
- In this project, they collected an evaluation dataset of face images, which aimed to have balanced representation of both faces with lighter and darker skin tones, and across men and women. To do this they curated the public face images from members of parliament across several African and European countries.
- Then, they evaluated three commercial systems, from Microsoft, Face++, and IBM, on the task of gender classification (which in itself maybe a dubious task, but was one of the services offered by these companies).
- The results were striking: all three systems got nearly 100% accuracy for lighter-skinned males, but they all got much worse accuracy for darker-skinned females.
- Gender Shades is just one of many examples pointing at a general and systemic problem: machine learning models, which are typically optimized to maximize average accuracy, can yield poor accuracies for certain **groups** (subpopulations).
- These groups can be defined by protected classes as given by discrimination law such as race and gender, or perhaps defined by the user identity or location.

[from a New York Times article]

False arrest due to facial recognition



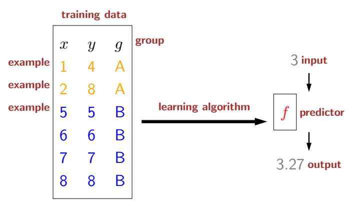
Wrongfully Accused by an Algorithm

In what may be the first known case of its kind, a faulty facial recognition match led to a Michigan man's arrest for a crime he did not commit.

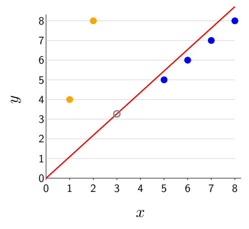
Real-life consequences

- Poor accuracy of machine learning systems can have serious real-life consequences. In one vivid case, a Black man by the name of Robert Julian-Borchak Williams was wrongly arrested due to a incorrect match with another Black man captured from a surveillance video, and this mistake was made by a facial recognition system.
- Given the Gender Shades project, we can see that lower accuracies for some groups might even lead to more arrests, which adds to the already problematic inequalities that exist in our society today.
- In this module, we'll focus only on performance disparities in machine learning and how we can mitigate them.
- But even if facial recognition worked equally well for all groups of people, should it even be used for law enforcement? Should it be used at all? These are bigger ethical questions, and it's always important to remember that sometimes, the issue is not with the solution, but the framing of the problem itself.

Linear regression with groups



x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



$$f_{\mathbf{w}}(x) = \mathbf{w} \cdot \phi(x) \qquad \mathbf{w} = [w] \qquad \phi(x) = [x]$$

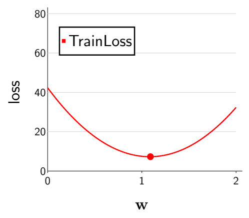
Note: predictor $f_{\mathbf{w}}$ does not use group information g

- Gender Shades was an example of classification, but to make things simpler, let us consider linear regression.
- We start with training data, but this time, each example consists of not only the input x and the output y, but also a **group** g (e.g., gender).
- In this example, we will assume we have two groups, A and B. As we see on the plot, the A points rise quickly, and the B points lie on the identity line, so the points behave differently.
- Recall the goal of regression is to produce a predictor that can take in a new input and predict an output.
- In linear regression, each predictor $f_{\mathbf{w}}$ computes the dot product of the weight vector $\mathbf{w}$ and a feature vector $\phi(x)$, and for this example, we define the feature map $\phi(x)$ to be the identity map, so that our hypothesis class consists of lines that pass through the origin.
- Looking ahead a bit, we see that there is a bit of a tension, where what slope w is best for group A is not the same as what is best for group B. The question is how we can compromise.
- Note that in this setting, the predictor $f_{\mathbf{w}}$ does not use the group information g, so it cannot explicitly specialize to the different groups. The group information is only used by the learning algorithm as well as to evaluate the performance across different groups.

Average loss

$$\text{Loss}(x, y, \mathbf{w}) = (f_{\mathbf{w}}(x) - y)^2$$

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



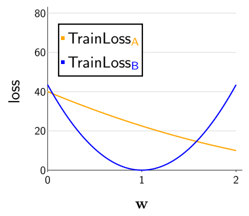
$$\text{TrainLoss}(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}} \text{Loss}(x, y, \mathbf{w})$$

$$\text{TrainLoss}(1) = \tfrac{1}{6}((1-4)^2 + (2-8)^2 + (5-5)^2 + (6-6)^2 + (7-7)^2 + (8-8)^2) = 7.5$$

- Recall that in regression, we typically use the squared loss, which measures how far away the prediction $f_{\mathbf{w}}(x)$ is away from the target y.
- Also recall that we defined the training loss to be an average of the per-example losses. This gives us a loss value for each value of $\mathbf{w}$ (see plot).
- So if we evaluate the training loss at $\mathbf{w} = 1$, we're averaging over the 6 examples. For each example, the prediction $f_{\mathbf{w}}(x)$ is just x (since $\mathbf{w} \cdot \phi(x) = [1] \cdot [x] = x$, so we can average the squared differences between x and y, producing a final answer of 7.5.
- If you evaluate $\mathbf{w}$ at a different value, you get a different training loss.
- If we minimize the training loss, then we can find this point $\mathbf{w} = 1.09$.

Per-group loss

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



$$\text{TrainLoss}_g(\mathbf{w}) = \frac{1}{|\mathcal{D}_{\text{train}}(g)|} \sum_{(x,y) \in \mathcal{D}_{\text{train}}(g)} \text{Loss}(x, y, \mathbf{w})$$

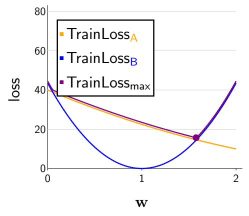
$$\text{TrainLoss}_{\text{A}}(1) = \tfrac{1}{2}((1-4)^2 + (2-8)^2) = 22.5$$

$$\text{TrainLoss}_{\text{B}}(1) = \tfrac{1}{4}((5-5)^2 + (6-6)^2 + (7-7)^2 + (8-8)^2) = 0$$

- Let us now take a careful look at the loss of each group.
- First, define $\mathcal{D}_{\text{train}}(g)$ to be the set of examples in group g. For example, $\mathcal{D}_{\text{train}}(\text{A}) = \{(1,4),(2,8)\}$.
- Then define the **per-group loss** TrainLoss_g of a weight vector $\mathbf{w}$ to be the average loss over the points in group g.
- If we choose $\mathbf{w} = [1]$ as our running example (because it's easy to compute), we see that the per-group loss on group A is 22.5, while the per-group loss on group B is 0.
- So note that even though the average loss was only 7.5, there is a huge performance disparity, with group A suffering a much larger loss.

Maximum group loss

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



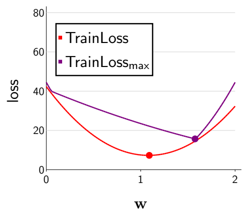
$$\text{TrainLoss}_{\text{max}}(\mathbf{w}) = \max_g \text{TrainLoss}_g(\mathbf{w})$$

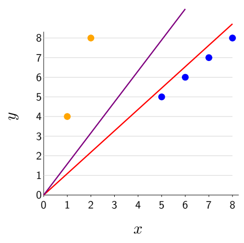
$\text{TrainLoss}_{\text{A}}(1) = 22.5$
$\text{TrainLoss}_{\text{B}}(1) = 0$
$\text{TrainLoss}_{\text{max}}(1) = \max(22.5, 0) = 22.5$

- We now want to somehow capture the per-group losses by one number. To do this, we look at the worst-case over groups.
- We now define the **maximum group loss** to be the largest over all groups. This function is the pointwise maximum of the per-group losses, which you can see on the plot as taking the upper envelope of the two per-group losses.
- We call this method group distributionally robust optimization (group DRO), because it is a special case of a broader framework Distributionally robust optimization (DRO).
- Going back to our running example with $\mathbf{w} = [1]$, we take the maximum of 22.5 and 0, which is 22.5.
- Note that this is much higher than the average loss (7.5), signaling that some group(s) are far less well off than the average.

Average loss versus maximum group loss

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B





Standard learning:

minimizer of average loss: $\mathbf{w} = 1.09$

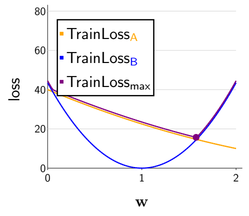
Group distributionally robust optimization (group DRO):

minimizer of maximum group loss: $\mathbf{w} = 1.58$

- Let us now compare the old average loss (the standard training loss) TrainLoss and the new maximum group loss $\text{TrainLoss}_{\text{max}}$.
- We minimize the average loss by setting the slope $\mathbf{w} = [1.09]$, yielding an average loss of 7.29. However, this setting gets much higher maximum group loss (over 20). Pictorially, we see that this $\mathbf{w}$ heavily favors the majority group (B).
- If instead we minimize the maximum group loss directly, we would choose $\mathbf{w} = 1.58$, yielding a maximum group loss of 15.59. Pictorially, we see that this solution pays more attention to (is closer to) the A points.
- Intuitively, the average loss favors majority groups over minority groups, but the maximum group loss gives a stronger voice to the minority groups, and as we see here, their influence is felt to a greater extent.

Training via gradient descent

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B



$$\text{TrainLoss}_{\text{max}}(\mathbf{w}) = \max_g \text{TrainLoss}_g(\mathbf{w})$$

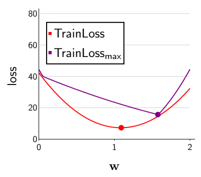
$$\nabla\text{TrainLoss}_{\text{max}}(\mathbf{w}) = \nabla\text{TrainLoss}_{g^*}(\mathbf{w})$$

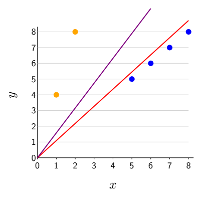
$$\text{where } g^* = \arg\max_g \text{TrainLoss}_g(\mathbf{w})$$

- In general, we can find minimize the maximum group loss by gradient descent.
- We just have to be able to take the gradient of $\text{TrainLoss}_{\text{max}}$, which is a maximum over the per-group losses.
- The gradient of a max is simply the gradient of the term that achieves the max.
- So algorithmically, it's very intuitive: you first compute whichever group (g^*) has the highest loss, and then you just evaluate the gradient only on per-group loss of that group (g^*).
- but you can see this paper for more details.

Summary

x	y	g
1	4	A
2	8	A
5	5	B
6	6	B
7	7	B
8	8	B





- Maximum group loss $\neq$ average loss
- Group DRO: minimize the maximum group loss
- Many more nuances: intersectionality? don't know groups? overfitting?

- To summarize, we've introduced the setting where examples are associated with groups. We see that by default, doing well on average is not the same as doing well on all groups (in other words, average loss is not the same as the maximum group loss), and the optimal solution for one objective is not optimal for the other objective.
- We presented an approach, group DRO that can ensure that the worst-off group is doing well.
- One parting remark is that while group DRO offers a mathematically clean solution, there are many subtleties when applying this to the real world. Intersectionality refers to the fact that groups which are defined by the conjunction of multiple attributes (e.g., White woman) may behave differently than the groups defined by individual attributes. What if you don't even have group information? How do you deal with overfitting (we've only looked at the training loss)?
- These are questions that are beyond the scope of this module, but I hope this short module has raised the need to think about about inequality as a first-class citizen, and piqued your interest to learn more.
- For further reading, consider checking out the book Fairness and machine learning: Limitations and Opportunities,