2021 IEEE East Asian School of Information Theory

Fair machine learning

Lecture 3

Changho Suh
EE, KAIST

Aug. 3, 2021

A fair classifier using kernel density estimation

Reading: TN3

Outline of Lecture 3: Fair classifier #2

1. Revisit the MI-based optimization.
2. Explore a way to directly compute the fairness measure DDP.
3. Introduce a trick that allows us to well approximate DDP:

   Kernel Density Estimation (KDE)

4. Develop a KDE-based optimization for a fair classifier.
5. Study how to solve the optimization.

Revisit: MI-based optimization

$$\min_{w} \frac{1-\lambda}{m} \sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \lambda \cdot I(Z; \hat{Y})$$

Rationale behind the training instability:

$I(Z; \hat{Y})$ = "max" optimization

⇨ "min-max" optimization often suffers from training instability.

How to address this?

Go back to the fairness measure DDP

$$\text{DDP} := \sum_{z \in \mathcal{Z}} |\mathbb{P}(\tilde{Y} = 1 | Z = z) - \mathbb{P}(\tilde{Y} = 1)|$$

Let's try to compute this directly.

First focus on:

$$\mathbb{P}(\tilde{Y} = 1) = \mathbb{P}(\hat{Y} \geq \tau) \qquad \tilde{Y} := \mathbf{1}\{\hat{Y} \geq \tau\}$$

$$= \int_{\tau}^{\infty} \underbrace{f_{\hat{Y}}(t)}_{\text{pdf}} dt$$

pdf uknown!

Instead: We are given samples $\{\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}\}$

Question: A way to infer the pdf from samples?

Kernel density estimation (KDE)

$$\mathbb{P}(\tilde{Y} = 1) = \int_{\tau}^{\infty} f_{\hat{Y}}(t)dt$$

Given samples $\{\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}\}$, KDE is defined as:

$$\widehat{f}_{\hat{Y}}(t) := \frac{1}{mh} \sum_{i=1}^{m} f_{\text{ker}}\left(\frac{t - \hat{y}^{(i)}}{h}\right)$$

a smoothing parameter (bandwidth) → h

a kernel function (e.g., Gaussian kernel) → f_{ker}

Accuracy of KDE

$$\mathbb{P}(\tilde{Y} = 1) = \int_{\tau}^{\infty} f_{\hat{Y}}(t)dt$$

Given samples $\{\hat{y}^{(1)}, \ldots, \hat{y}^{(m)}\}$, KDE is defined as:

$$\widehat{f}_{\hat{Y}}(t) := \frac{1}{mh} \sum_{i=1}^{m} f_{\text{ker}} \left(\frac{t - \hat{y}^{(i)}}{h} \right)$$

Jiang ICML17: $|\widehat{f}(t) - f(t)|_{\infty} \lesssim \frac{1}{m^{\frac{1}{d}}}$ ← dim. of an interested r.v.

→ Yields an inaccurate estimate under **high-dim.** cases

Good news: In our setting, $d = 1$

Approximation via KDE

$$\mathbb{P}(\tilde{Y} = 1) = \int_{\tau}^{\infty} f_{\hat{Y}}(t) dt$$

$$\widehat{\mathbb{P}}(\tilde{Y} = 1) = \int_{\tau}^{\infty} \widehat{f}_{\hat{Y}}(t) dt$$

$$= \int_{\tau}^{\infty} \frac{1}{mh} \sum_{i=1}^{m} f_{\mathsf{ker}} \left(\boxed{\frac{t - \hat{y}^{(i)}}{h}} \right) dt \qquad \boxed{\cdots} =: y$$

$$= \frac{1}{m} \sum_{i=1}^{m} \int_{\frac{\tau - \hat{y}^{(i)}}{h}}^{\infty} f_{\mathsf{ker}}(y) dy$$

$$= \frac{1}{m} \sum_{i=1}^{m} Q \left(\frac{\tau - \hat{y}^{(i)}}{h} \right) \quad \text{(Gaussian kernel)}$$

Approximation via KDE

$$\widehat{\mathbb{P}}(\tilde{Y} = 1) = \frac{1}{m} \sum_{i=1}^{m} Q\left(\frac{\tau - \hat{y}^{(i)}}{h}\right)$$

Remember: $\text{DDP} := \sum_{z \in \mathcal{Z}} |\mathbb{P}(\tilde{Y} = 1 | Z = z) - \mathbb{P}(\tilde{Y} = 1)|$

Similarly, one can obtain:

$$\widehat{\mathbb{P}}(\tilde{Y} = 1 | Z = z) = \frac{1}{m_z} \sum_{i \in I_z} Q\left(\frac{\tau - \hat{y}^{(i)}}{h}\right)$$

m_z: $|I_z|$

I_z: $\{i : z^{(i)} = z\}$

Approximated DDP

$$\text{DDP} := \sum_{z \in \mathcal{Z}} |\mathbb{P}(\tilde{Y} = 1 | Z = z) - \mathbb{P}(\tilde{Y} = 1)|$$

$$\approx \sum_{z \in \mathcal{Z}} |\widehat{\mathbb{P}}(\tilde{Y} = 1 | Z = z) - \widehat{\mathbb{P}}(\tilde{Y} = 1)|$$

$$= \sum_{z \in \mathcal{Z}} \left| \frac{1}{m_z} \sum_{i \in I_z} Q\left(\frac{\tau - \hat{y}^{(i)}}{h}\right) - \frac{1}{m} \sum_{i=1}^{m} Q\left(\frac{\tau - \hat{y}^{(i)}}{h}\right) \right|$$

$$\approx \sum_{z \in \mathcal{Z}} \left| \frac{1}{m_z} \sum_{i \in I_z} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} - \frac{1}{m} \sum_{i=1}^{m} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} \right|$$

Can express DDP in terms of samples (thus *w*)

KDE-based optimization

Cho-Hwang-Suh NeurIPS20

$$\min_{w} \frac{1-\lambda}{m} \sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{m} \cdot \sum_{z \in \mathcal{Z}} \left| \frac{m}{m_z} \sum_{i \in I_z} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} - \sum_{i=1}^{m} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} \right|$$

Algorithm: Gradient descent

Issues: How to deal with the absolute function?

How to choose bandwidth h?

How to deal with the absolution func?

$$\min_{w} \frac{1-\lambda}{m} \sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{m} \cdot \sum_{z \in \mathcal{Z}} \left| \frac{m}{m_z} \sum_{i \in I_z} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} - \sum_{i=1}^{m} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} \right|$$

Instead, one can employ Huber loss:

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \le \delta \\ \delta\left(|x| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases}$$

This enables us to readily obtain gradient.

How to choose bandwidth *h*?

$$\min_{w} \frac{1-\lambda}{m} \sum_{i=1}^{m} \ell_{\mathsf{CE}}(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{m} \cdot \sum_{z \in \mathcal{Z}} H_\delta \left(\frac{m}{m_z} \sum_{i \in I_z} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} - \sum_{i=1}^{m} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} \right)$$

Turns out:

There is a sweet spot for h that miminizes the mean square error of KDE estimate.

Advise us to find h^* that minimizes the MSE.

See [Cho-Hwang-Suh NeurIPS20] for details.

Extension to another fairness measure DEO

$$\text{DEO} := \sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} \left| \mathbb{P}(\tilde{Y} = 1 | Y = y, Z = z) - \mathbb{P}(\tilde{Y} = 1 | Y = y) \right|$$

$$\approx \sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} \left| \widehat{\mathbb{P}}(\tilde{Y} = 1 | Y = y, Z = z) - \widehat{\mathbb{P}}(\tilde{Y} = 1 | Y = y) \right|$$

$$\approx \sum_{y \in \mathcal{Y}} \sum_{z \in \mathcal{Z}} \left| \frac{1}{m_{yz}} \sum_{i \in I_{yz}} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} - \frac{1}{m_y} \sum_{i \in I_y} e^{-\frac{(\tau - \hat{y}^{(i)})^2}{2h^2}} \right|$$

m_{yz}: $|I_{yz}|$

I_{yz}: $\{i : y^{(i)} = y, z^{(i)} = z\}$

Experiments

A benmark real dataset: **COMPAS**

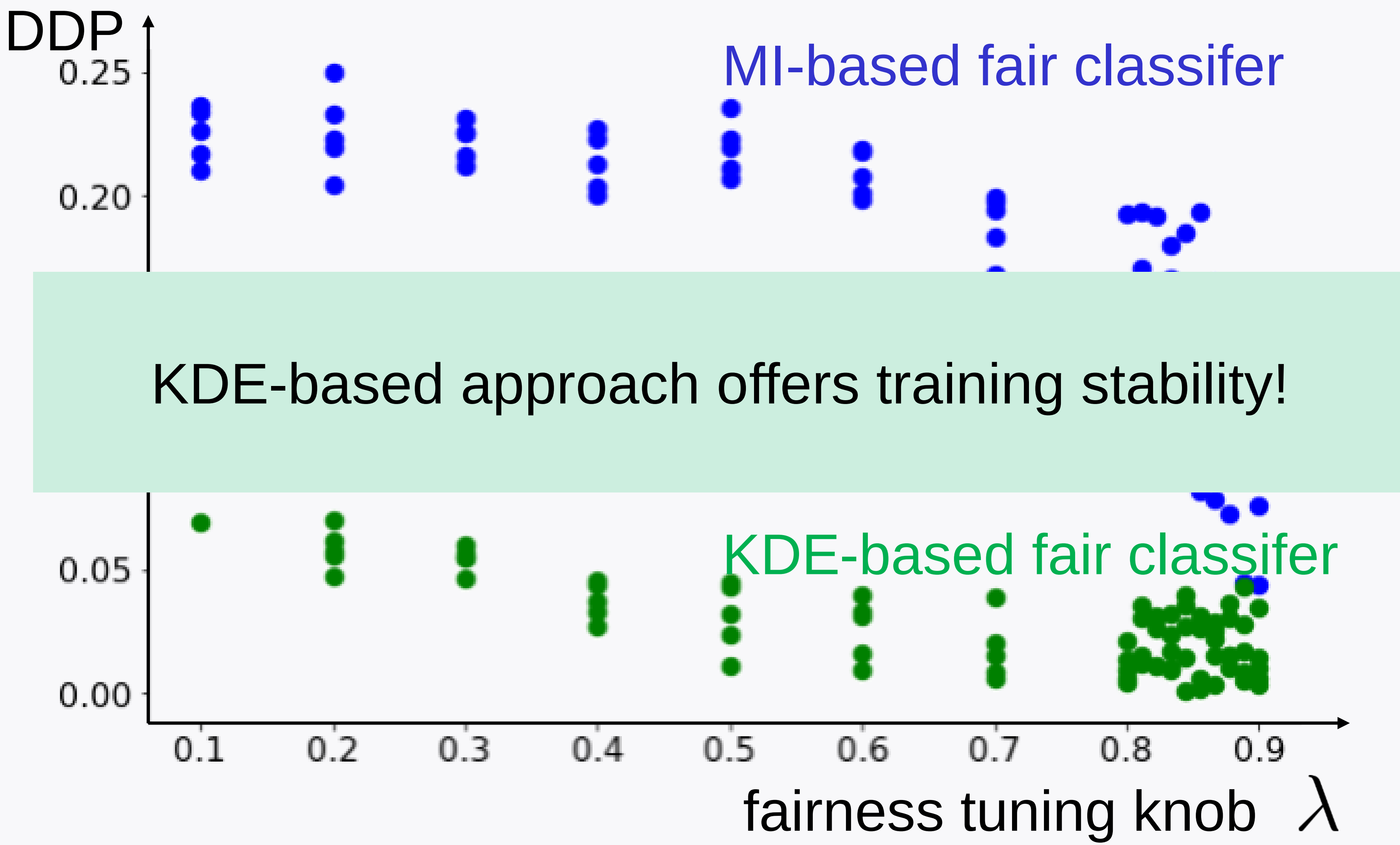
$$(x, z, y)$$

criminal records → x

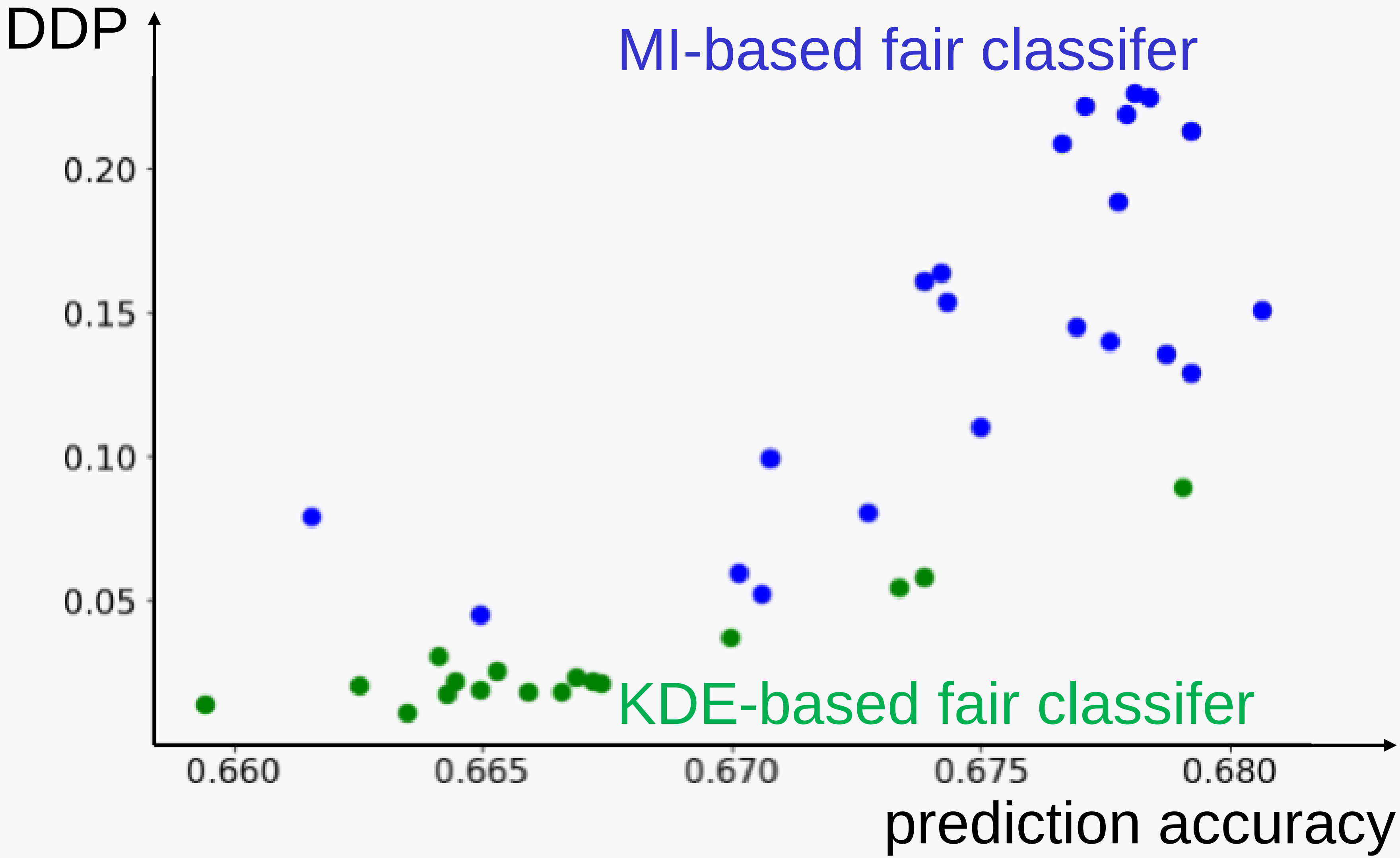
black or white → z

reoffend or not in near future → y

A challenge


DDP
0.25
0.20
0.05
0.00
MI-based fair classifer
KDE-based approach offers training stability!
KDE-based fair classifer
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
fairness tuning knob λ

Accuracy vs DDP tradeoff



Summary of Lectures 1/2/3

1. Explore fairness measures in fair classifiers.
2. Study an MI-based fair classifier which yields a good tradeoff while suffering from training instability.
3. Investigate another fair classifer based on KDE, which addresses the training instability issue.

Look ahead

Will move onto the 2nd context:

Fair generative models

Reference

[1] J. Cho, G. Hwang and C. Suh. A fair classifier using mutual information. *IEEE International Syposium on Inofrmation Theory* (*ISIT),* 2020.

[2] J. Cho, G. Hwang and C. Suh. A fair classifier using kernel density estimation. *In Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[3] H. Jiang. Uniform convergence rates for kernel density estimation. *International Conference on Machine Learning (ICML)*, 2017.

[4] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: There's software used across the country to 272 predict future criminals. And it's biased against blacks. *https://www.propublica.org/article/machine-bias-risk-assessments-incriminal-sentencing*, 2015.