Influence Based Approaches to Algorithmic Fairness: A Closer Look

Soumya Ghosh
MIT-IBM Watson AI Lab, IBM Research
Cambridge, MA
ghoshso@us.ibm.com

Prasanna Sattigeri
MIT-IBM Watson AI Lab, IBM Research
Cambridge, MA
psattig@us.ibm.com

Manish Nagireddy
MIT-IBM Watson AI Lab, IBM Research
Cambridge, MA
manish.nagireddy@ibm.com

Inkit Padhi
MIT-IBM Watson AI Lab, IBM Research
Yorktown Heights, NY
inkit.padhi@ibm.com

Jie Chen
MIT-IBM Watson AI Lab, IBM Research
Cambridge, MA
chenjie@us.ibm.com

Abstract

Off-the-shelf pre-trained models are increasingly common in machine learning. When deployed in the real world, it is essential that such models are not just accurate but also demonstrate qualities like fairness. This paper takes a closer look at recently proposed approaches that edit a pre-trained model for group fairness by re-weighting the training data. We offer perspectives that unify disparate weighting schemes from past studies and pave the way for new weighting strategies to address group fairness concerns.

1 Introduction

Pre-trained models, either as is or after minor adaptation, are increasingly used in machine learning practice. These models are typically trained via empirical risk minimization to maximize some notion of predictive accuracy, often producing highly accurate predictions on in-distribution test data. However, accuracy is usually only one of many attributes of interest in real-world applications. For example, ensuring the model does not produce systematically biased predictions for any data sub-group is crucial in high-stakes applications such as healthcare.

Motivated by these observations, several *post-hoc* approaches [9, 7, 14, 12, 18, 19] have been proposed that either edit the pre-trained model or transform its predictions to satisfy a desired measure of fairness, such as demographic parity and equality of opportunity. We focus on a set of recently proposed data-centric, post-hoc techniques [15, 16, 17] that use influence functions [6, 10] to gauge the impact of training data on fairness measures. Not only do they offer interpretability by highlighting influential training instances, but they also re-weight these instances — by omitting, doubling, or otherwise re-weighting them to mitigate unfairness. Here, we show that these different re-weighting approaches can be viewed as specific instantiations of the same optimization problem. Moreover, the optimization view immediately suggests new re-weighting schemes that empirically improve on existing ones.

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

Finally, we provide preliminary insights into the source of the favorable accuracy-fairness tradeoff demonstrated by the re-weighted model of [16].

2 Background

Consider a standard supervised learning setup where we have access to a dataset $\mathcal{D} = \{\mathbf{z}_n = (\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ of N feature ($\mathbf{x}_n \in \mathbb{R}^p$) and response ($\mathbf{y}_n \in \mathcal{Y}$) pairs, a model $h_{\boldsymbol{\theta}}(\mathbf{x})$ parameterized by $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^D$, and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$. We fit the model to $\mathcal{D}$ by minimizing the empirical risk, $\mathcal{L}(\boldsymbol{\theta}) \overset{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N \ell(\mathbf{y}_n, h_{\boldsymbol{\theta}}(\mathbf{x}_n))$, with respect to $\boldsymbol{\theta}$ and denote the empirical risk minimizer, $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\Theta}{\text{argmin}}\ \mathcal{L}(\boldsymbol{\theta})$. We note that for modern large language models (LLMs), $\boldsymbol{\theta}$ may be initialized by minimizing an alternate loss ℓ' on a potentially much larger pre-training dataset $\mathcal{D}_{pre}$. In this scenario, the empirical risk minimization setup described here corresponds to the supervised fine-tuning (SFT) step of training LLMs. Our experiments with LLMs in this paper use this SFT setup.

2.1 Influence functions and the infinitesimal jackknife approximation

Next, consider the weighted empirical risk, $\mathcal{L}(\boldsymbol{\theta}, \mathbf{w}) \overset{\text{def}}{=} \frac{1}{N}\sum_{n=1}^N w_n \ell(\mathbf{y}_n, h_{\boldsymbol{\theta}}(\mathbf{x}_n))$, that weights each training instance's loss by a scalar weight $w_n \in \mathcal{W} \subseteq \mathbb{R}$ and $\mathbf{w}$ is a column vector $[w_1, w_2, \ldots, w_N]^T \in \mathcal{W}^N \subseteq \mathbb{R}^N$. Observe that the optimal solution of the weighted problem,

$$\boldsymbol{\theta}^*(\mathbf{w}) = \underset{\boldsymbol{\theta}\in\Theta}{\text{argmin}}\ \mathcal{L}(\boldsymbol{\theta}, \mathbf{w}) = \underset{\boldsymbol{\theta}\in\Theta}{\text{argmin}} \frac{1}{N}\sum_{n=1}^N w_n \ell(\mathbf{y}_n, h_{\boldsymbol{\theta}}(\mathbf{x}_n)), \tag{1}$$

is a function of $\mathbf{w}$, $\boldsymbol{\theta}^*(\mathbf{w}) : \mathbb{R}^N \rightarrow \mathbb{R}^D$. While, in general, the functional form of $\boldsymbol{\theta}^*(\mathbf{w})$ is unknown, a common strategy is to form a first-order Taylor approximation to it about $\mathbf{1} \overset{\text{def}}{=} [w_1 = 1, w_2 = 1, \ldots, w_N = 1]^T$,

$$\boldsymbol{\theta}^*(\mathbf{w}) \approx \hat{\boldsymbol{\theta}} + \mathcal{I}(\mathbf{w} - \mathbf{1}), \tag{2}$$

where $\mathcal{I} \in \mathbb{R}^{D\times N}$ is shorthand for the Jacobian matrix $\nabla_{\mathbf{w}}\boldsymbol{\theta}^*(\mathbf{w})\Big|_{\mathbf{w}=\mathbf{1}}$ and we have used the fact that $\boldsymbol{\theta}^*(\mathbf{1}) = \hat{\boldsymbol{\theta}}$. When $\mathcal{L}(\boldsymbol{\theta})$ is twice differentiable in $\boldsymbol{\theta}$, and at a stationary point of $\mathcal{L}(\boldsymbol{\theta})$, i.e., when $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = 0$, an analytical expression for each column of the Jacobian matrix becomes available from the application of the implicit function theorem [11][1],

$$\mathcal{I}_n = \frac{\partial \boldsymbol{\theta}^*(\mathbf{w})}{\partial w_n}\bigg|_{\mathbf{w}=\mathbf{1}} = -\mathbf{H}^{-1}\mathbf{g}_n, \tag{3}$$

where $\mathcal{I}_n$ is the n^{th} column of $\mathcal{I}$, $\mathbf{H} \overset{\text{def}}{=} \nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$, and $\mathbf{g}_n \overset{\text{def}}{=} \nabla_{\boldsymbol{\theta}}\ell(y_n, h_{\boldsymbol{\theta}}(\mathbf{x}_n))|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$. $\mathcal{I}_n$ measures how $\boldsymbol{\theta}^*(\mathbf{w})$ varies as we re-weight the n^{th} training instance and is popularly called the influence function [6, 10]. This expression allow us to form a linear approximation, the infinitesimal jackknife (IJ) [8] approximation, to the solution of Equation 1 given a pre-trained model ($\hat{\boldsymbol{\theta}}$), a weight vector $\mathbf{w}$, and $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_N] \in \mathbb{R}^{D\times N}$,

$$\boldsymbol{\theta}^*(\mathbf{w}) \approx \hat{\boldsymbol{\theta}} - \mathbf{H}^{-1}\mathbf{G}(\mathbf{w} - \mathbf{1}). \tag{4}$$

2.2 Group Fairness

We assume that each data instance has an additional sensitive attribute $s_n \in [k]$, i.e., $\mathcal{D} = \{\mathbf{z}_n = (\mathbf{x}_n, s_n, y_n)\}_{n=1}^N$, that encodes the group membership of the n^{th} data instance. Here, the responses $y_n \in \{0, 1\}$ are binary labels associated with each data instance. In this setup, the goal is to learn accurate classifiers that minimize disparities in predictions across groups.

[1] For a detailed derivation see Appendix A of [10], or Appendix J of [5]

To quantify disparities, we rely on common fairness metrics — demographic (or statistical) parity (DP) [2] and equality of odds (EO) [7]. Let X and S denote random variables representing the features and the sensitive attribute. DP requires the classifier's predictions to be statistically independent of the sensitive attribute, $h_{\boldsymbol{\theta}}(X) \perp S$, and EO requires the classifier's predictions to be statistically independent of the sensitive attribute *conditioned* on the true outcome, $h_{\boldsymbol{\theta}}(X) \perp S \mid Y$ For a binary sensitive attribute, DP implies, $P(h_{\boldsymbol{\theta}}(X) = 1 \mid S = 1) = P(h_{\boldsymbol{\theta}}(X) = 1 \mid S = 0)$ while EO implies $P(h_{\boldsymbol{\theta}}(X) = 1 \mid S = 1, Y = y) = P(h_{\boldsymbol{\theta}}(X) = 1 \mid S = 0, Y = y)$ for both $y = 0$ and $y = 1$. Difference in Equality of Opportunity (DEO) [7] is a special case of equality of odds that requires the predictions to be conditionally independent of the sensitive attribute given positive true outcome. We use smooth surrogates to the absolute difference in demographic parity and equality of odds as our model editing loss functions,

$$\begin{aligned}
\mathcal{F}^{\Delta\mathrm{DP}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}) &= |\mathbb{E}_{p_{\mathcal{D}_{\mathrm{val}}}(X=\mathbf{x}|S=1)}[h_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{p_{\mathcal{D}_{\mathrm{val}}}(X=\mathbf{x}|S=0)}[h_{\boldsymbol{\theta}}(\mathbf{x})]| \\
\mathcal{F}^{\Delta\mathrm{EO}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}) &= \sum_{y=0}^{1} |\mathbb{E}_{p_{\mathcal{D}_{\mathrm{val}}}(X=\mathbf{x}|S=0,Y=y)}[h_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{p_{\mathcal{D}_{\mathrm{val}}}(X=\mathbf{x}|S=1,Y=y)}[h_{\boldsymbol{\theta}}(\mathbf{x})]|.
\end{aligned} \tag{5}$$

We use these particular surrogates because they are widely used in the literature to regularize for fairness in the literature [20]. However, the approaches we study here are surrogate agnostic and others could [3] be used just as easily.

2.3 Fairness Influence

We can use the influence function machinery to understand how re-weighting a training instance affects the fairness disparity. For illustration, consider the case of demographic parity, other cases follow analogously. Equation 3 and the chain-rule allows us to quantify how demographic parity changes with re-weighting of the n^{th} training instance,

$$\begin{aligned}
\mathcal{I}_{\mathcal{F}^{\mathrm{DP}},n} \overset{\mathrm{def}}{=} \frac{\partial \mathcal{F}^{\Delta\mathrm{DP}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})}{\partial w_n}\Bigg|_{\mathbf{w}=\mathbf{1}, \boldsymbol{\theta}^*(\mathbf{w})=\boldsymbol{\theta}^*(\mathbf{1})} &= \nabla_{\boldsymbol{\theta}} \mathcal{F}^{\Delta\mathrm{DP}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})\Bigg|^{T}_{\mathbf{w}=\mathbf{1}, \boldsymbol{\theta}^*(\mathbf{w})=\boldsymbol{\theta}^*(\mathbf{1})} \frac{\partial \boldsymbol{\theta}^*(\mathbf{w})}{\partial w_n}\Bigg|_{\mathbf{w}=\mathbf{1}}, \\
&= -\nabla_{\boldsymbol{\theta}} \mathcal{F}^{\Delta\mathrm{DP}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})\Bigg|^{T}_{\mathbf{w}=\mathbf{1}, \boldsymbol{\theta}^*(\mathbf{w})=\boldsymbol{\theta}^*(\mathbf{1})} \mathbf{H}^{-1}\mathbf{g}_n, \\
&= -\mathbf{g}_{\mathrm{DP}}^{T}\mathbf{H}^{-1}\mathbf{g}_n,
\end{aligned} \tag{6}$$

which we refer to as the fairness influence.

3 Editing pre-trained models for fairness

Several recent works [15, 16, 17] have used this notion of fairness influence to edit pre-trained models such that they exhibit smaller fairness disparity. The key idea is to estimate a training instance's influence on a desired fairness disparity and then to drop, by setting the corresponding weight (w_n) to zero (or up-weight by setting w_n to two), instances that increase (or decrease) the fairness disparity. Finally, they recover the fair model either by explicitly minimizing Equation 1 [17, 15] or by leveraging the IJ approximation [16].

Moving beyond these intuitive yet heuristic data reweighting schemes, here we seek to find a re-weighting of the data, $\mathbf{w}^*$, such that,

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{W}^N}{\mathrm{argmin}}\ \mathcal{F}^{\Delta\mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}), \tag{7}$$

where a $\in$ [DP, EO, DEO] is a desired fairness discrepancy. Unfortunately, each evaluation of the objective in Equation 7 requires solving the weighted risk minimization problem of Equation 1 thus rendering a direct optimization of the objective in Equation 7 computationally prohibitive for most problems of interest encountered in practice.

To cope, we consider a linearization $\mathcal{F}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})$ about $\mathbf{w} = \mathbf{1}$,

$$\begin{aligned}\bar{\mathcal{F}}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) &\approx \mathcal{F}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{1}), \mathbf{1}) + \nabla_{\mathbf{w}} \mathcal{F}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) \Big|^{T}_{\mathbf{w}=\mathbf{1}, \boldsymbol{\theta}^*(\mathbf{w})=\boldsymbol{\theta}^*(\mathbf{1})} (\mathbf{w} - \mathbf{1}), \\ &= \mathcal{F}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{1}), \mathbf{1}) + \mathcal{I}^{T}_{\mathcal{F}^{\mathrm{a}}}(\mathbf{w} - \mathbf{1}),\end{aligned} \tag{8}$$

where $\mathcal{I}_{\mathcal{F}^{\mathrm{a}}} \overset{\mathrm{def}}{=} [\mathcal{I}_{\mathcal{F}^{\mathrm{a}},1}, \ldots, \mathcal{I}_{\mathcal{F}^{\mathrm{a}},N}]^T \in \mathbb{R}^N$, and $\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n} = -\mathbf{g}_{\mathrm{a}}^T \mathbf{H}^{-1} \mathbf{g}_n$ from Equation 6. Without any constraints on $\mathbf{w}$, the linearized surrogate $\bar{\mathcal{F}}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})$ can be trivially minimized by setting the components of $\mathbf{w}$ to $-\infty$. For non-trivial solutions, we need to constrain $\mathbf{w}$. We thus consider the following problem instead,

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{W}^N}{\operatorname{argmin}} \; \bar{\mathcal{F}}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) + \frac{1}{\lambda} \mathbb{D}(\mathbf{w}, \mathbf{1}), \tag{9}$$

where, $\mathbb{D} : \mathcal{W}^N \times \mathcal{W}^N \rightarrow \mathbb{R}_+$, is a proximal regularizer that encourages $\mathbf{w}$ to stay close to $\mathbf{1}$, and $\lambda \in \mathbb{R}_+$ controls the regularization strength. Beyond avoiding trivial solutions, the proximal regularization restricts $\mathbf{w}$ to regions where both the error in Equation 2, and the error stemming from the linearlization of $\mathcal{F}$ are expected to be small and can be driven to zero by setting $\lambda = 0$. Moreover, for many reasonable choices of $\mathcal{W}$ and $\mathbb{D}$ Equation 9 can be minimized *analytically* without requiring iterative gradient based minimization.

3.1 Existing weighting rules use Hamming distance as the proximal regularizer

Discrete $\mathcal{W}$ We begin by showing that when $\mathcal{W} = \{0, 1\}$ and $\mathbb{D}$ is the Hamming distance, i.e., $\mathbb{D}(\mathbf{w}, \mathbf{1}) = \sum_{n=1}^{N} \mathbb{1}[w_n \neq 1]$, where $\mathbb{1}[a \neq b]$ is an indicator function that takes a value of one when $a \neq b$, and zero otherwise, we recover the *drop-K* re-weighting rule proposed in [16] and subsequently used in [15]. Following [16] we alternatively refer to this as the `Fair-IJ` weighting rule.

Proposition 3.1 (***drop-K*** **/** **`Fair-IJ`** **weighting rule).** *Let $\mathcal{W}^N = \{0, 1\}^N$ be the set of all N dimensional binary vectors, and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the Hamming distance between $\mathbf{w}$ and $\mathbf{1}$, then Equation 9 is minimized by setting the weights of the K largest positive influence training instances to zero.*

Proof. Plugging in the constraint and the proximal regularizer in Equation 9, we have,

$$\mathbf{w}^* = \underset{\mathbf{w} \in \{0,1\}^N}{\operatorname{argmin}} \; \bar{\mathcal{F}}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) + \frac{1}{\lambda} \mathcal{H}(\mathbf{w}, \mathbf{1}) \; = \underset{\mathbf{w} \in \{0,1\}^N}{\operatorname{argmin}} \; \bar{\mathcal{F}}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) + \frac{1}{\lambda} \sum_{n=1}^{N} |w_n - 1|,$$

where in the second equality we have plugged in the definition of Hamming distance for binary vectors. Expanding the linearization term, gives us the following minimization problem,

$$\min_{\mathbf{w} \in \{0,1\}^N} \mathcal{F}^{\Delta \mathrm{a}}_{\mathcal{D}_{\mathrm{val}}}(\boldsymbol{\theta}^*(\mathbf{1}), \mathbf{1}) + \mathcal{I}^{T}_{\mathcal{F}^{\mathrm{a}}}(\mathbf{w} - \mathbf{1}) + \frac{1}{\lambda} \sum_{n=1}^{N} |w_n - 1| = \min_{\mathbf{w} \in \{0,1\}^N} \sum_{n=1}^{N} \big(\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n}(w_n - 1) + \frac{1}{\lambda}|w_n - 1|\big) + \text{constant}.$$

Ignoring the constant independent of $\mathbf{w}$, we have the equivalent problem,

$$\min_{\mathbf{w} \in \{0,1\}^N} \sum_{n=1}^{N} \mathbb{1}[\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n} > 0] \big(\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n}(w_n - 1) + \frac{1}{\lambda}|w_n - 1|\big) + \sum_{n=1}^{N} \mathbb{1}[\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n} < 0] \big(\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n}(w_n - 1) + \frac{1}{\lambda}|w_n - 1|\big),$$

where $\mathbb{1}[a > b]$ is an indicator function that takes a value of one when $a > b$ and zero otherwise. We arrive at the result by observing that this objective is minimized when the second summation is zero, i.e., by setting $w_n = 1$, when $\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n} < 0$. Furthermore, from the first summation we see that setting $w_n = 0$ when $\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n} > 0$ reduces the objective by $-\mathcal{I}_{\mathcal{F}^{\mathrm{a}},n}$ at the cost of $\frac{1}{\lambda}$. The first term is thus minimized by sorting the positive influences and proceeding in descending order, setting the corresponding $w_n = 0$ until the K^{th} instance such that $\frac{K}{\lambda} > \sum_{n=1}^{K} \mathcal{I}_{\mathcal{F}^{\mathrm{a}},n}$. □

Thus `Fair-IJ` is a special case within our framework. Moreover, when $\mathcal{W}^N$ is the space of ternary vectors, minimizing Equation 9 recovers the *drop-K, upweight-M* re-weighting rule employed by [17]. See Appendix C for details.

3.2 Mahalanobis distance based regularizers yield new weighting schemes

Continuous $\mathcal{W}$ Next, we consider weights vectors $\mathbf{w}$ that are not constrained to be binary or ternary but instead live in $\mathbb{R}^N$.

Proposition 3.2 (I-JACK **weighting rule).** *Let $\mathcal{W}^N = \mathbb{R}^N$ and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the squared ℓ_2 norm, $\frac{1}{2}||\mathbf{w} - \mathbf{1}||_2^2$, then Equation 9 is minimized by setting, $\mathbf{w}^* = -\lambda \mathcal{I}_{\mathcal{F}^a} + \mathbf{1}$.*

Proposition 3.3 (SIM-JACK **weighting rule).** *Let $\mathcal{W}^N = \mathbb{R}^N$ and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the squared Mahalanobis distance $\frac{1}{2}(\mathbf{w} - \mathbf{1})^T \mathbf{C}^{-1}(\mathbf{w} - \mathbf{1})$, where $\mathbf{C} \in \mathcal{S}_+^{N \times N}$ is a symmetric positive definite matrix, then Equation 9 is minimized by setting, $\mathbf{w}^* = -\lambda \mathbf{C} \mathcal{I}_{\mathcal{F}^a} + \mathbf{1}$.*

Both these results follows from setting the gradient of Equation 9 to zero and rearranging terms. See for a simple proof Appendix C of these results. Differently from the other rules, SIM-JACK through the matrix $\mathbf{C}$, allows us to model correlated weights. Such correlated weights can be particularly useful if the influence estimation itself is noisy, due to numerical imprecision or approximations stemming from computational considerations. By encouraging similar data instances to have similar influences SIM-JACK can provide a degree of robustness to noisy influence estimation.

While various parameterizations of $\mathbf{C}$ are possible, working with a full rank $N \times N$ matrix is prohibitively expensive for large N. Instead, we use linear kernels of the form $\mathbf{C} = \Phi^T \Phi + \varepsilon \mathbf{I}$, where $\Phi = [\phi(\mathbf{z}_1), \ldots, \phi(\mathbf{z}_N)] \in \mathbb{R}^{P \times N}$, and ε is a small positive number to ensure invertibility of $\mathbf{C}$ and $P << N$. While various feature transformations $\phi(\mathbf{z}_n))$ are possible, we find that while simply using an intermediate layer of the pre-trained model, i.e., $\phi(\mathbf{z}_n) = l_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_n)$, where l is an intermediate layer of the network works well for tabular data, for natural language it is often beneficial to use an auxiliary *encoder-only* transformer (for example, the RoBerta family of models were used in our experiments).

3.3 Updating for fairness without sacrificing empirical risk

Plugging in the different weighting rules into Equation 4 provides us with different rules for updating the model parameters.

Fair-IJ update rule

$$\boldsymbol{\theta}_{\text{fair}} = \hat{\boldsymbol{\theta}} + \sum_m \mathbf{H}^{-1} \mathbf{g}_m, \tag{10}$$

where m ranges over training instance indices corresponding to the K largest positive influence $(-\mathbf{g}_m^T \mathbf{H}^{-1} \mathbf{g}_{\text{a}} > 0)$ instances. Note that since $\mathbf{g}_{\text{a}}^T(\mathbf{H}^{-1}\mathbf{g}_m) < 0$, the Newton *ascent* defined in Equation 10 reduces the fairness discrepancy of interest denoted by a.

I-JACK update rule

$$\boldsymbol{\theta}_{\text{fair}} = \hat{\boldsymbol{\theta}} - \mathbf{H}^{-1} \left(\lambda \sum_n \mathbf{g}_n \mathbf{g}_n^T \right) \mathbf{H}^{-1} \mathbf{g}_{\text{a}}. \tag{11}$$

SIM-JACK update rule

$$\boldsymbol{\theta}_{\text{fair}} = \hat{\boldsymbol{\theta}} - \mathbf{H}^{-1} \left(\lambda \sum_n \mathbf{g}_n \big(\sum_m \mathbf{C}_{nm} \mathbf{g}_m^T \big) \right) \mathbf{H}^{-1} \mathbf{g}_{\text{a}}. \tag{12}$$

In all the above a ∈ [DP, EO, DEO]. Contrasting with the standard gradient update, $\hat{\boldsymbol{\theta}} - \lambda \mathbf{g}_{\text{a}}$, we immediately see that all three IJ based updates account for the curvature of $\mathcal{L}(\boldsymbol{\theta})$ via $\mathbf{H}$ while the gradient update does not. In particular, the spectral decomposition of $\mathbf{H}^{-1}$ is $\sum_{p=1}^D \frac{1}{\nu_p} \mathbf{u}_p \mathbf{u}_p^{T2}$, where $\nu_p, \mathbf{u}_p$ are the p^{th} eigenvalue, eigenvector pair of $\mathbf{H}$. For any vector $\mathbf{v} \in \mathbb{R}^D$, the update,

$$\hat{\boldsymbol{\theta}} - \mathbf{H}^{-1}\mathbf{v} = \hat{\boldsymbol{\theta}} - \sum_p \frac{\mathbf{u}_p^T \mathbf{v}}{\nu_p} \mathbf{u}_p, \tag{13}$$

[2] assuming $\mathbf{H}$ is positive definite. In practice, we will add a small diagonal for positive definiteness

deemphasizes update directions corresponding to large ν_p. Since eigenvectors associated with the larger eigenvalues point in directions where $\mathcal{L}(\boldsymbol{\theta})$ increases faster, updates of the form $\mathbf{H}^{-1}\mathbf{v}$, encourage $\boldsymbol{\theta}_{\text{fair}}$ to lie in flatter regions of $\mathcal{L}(\boldsymbol{\theta})$. This property allows the IJ based updates to find $\boldsymbol{\theta}_{\text{fair}}$ that do not substantially increase the original training loss. Moreover, `I-JACK` and `SIM-JACK` adaptively control the degree to which larger eigenvalues are penalized, and smaller emphasized. Moreover, in the `I-JACK` update rule the scaled empirical covariance of the per-sample gradients $\lambda \sum_n \mathbf{g}_n \mathbf{g}_n^T$[3] adaptively control the degree to which larger eigenvalues are penalized, and smaller emphasized. When $\lambda \sum_n \mathbf{g}_n \mathbf{g}_n^T$ is well approximated by the identity matrix, the `I-JACK` update is $\hat{\boldsymbol{\theta}} - \sum_p \frac{\mathbf{u}_p^T \mathbf{g}_{\text{a}}}{\nu_p^2} \mathbf{u}_p$ an even stronger, quadratic regularization towards flat regions of $\mathcal{L}(\boldsymbol{\theta})$, while for a well-specified generalized linear model, and ℓ is the negative log-likelihood, as $N \to \infty$, both $\frac{1}{N} \sum_n \mathbf{g}_n \mathbf{g}_n^T$ and $\frac{1}{N}\mathbf{H}$ tend to the Fisher information matrix, we recover Equation 13. The `SIM-JACK` update is similar, but replaces the outer product $\mathbf{g}_n \mathbf{g}_n^T$ with the outer-product between $\mathbf{g}_n$ and the linear combination $\sum_m \mathbf{C_{nm}} \mathbf{g_m^T}$, where $\mathbf{C}_{nm}$ capture a user defined notion of similarity between training instances n and m. The update collapses to the `I-JACK` update when $\mathbf{C}$ is a $N \times N$ identity matrix.

4 Experiments

Experimental Setup: We empirically validate the proposed methods on two datasets: the tabular *Adult* dataset and the natural language *CivilComments* dataset. Both have been used in prior work to benchmark post-hoc fairness algorithms [16]. We assess fairness using the EO, DEO, and DP metrics. For the *Adult* dataset, our model is a 1-hidden layer MLP with 100 units and SeLU non-linearities. For the *CivilComments* dataset, we employ a T5-base model adapted via soft prompt-tuning. We experiment with different number of soft-prompt or *virtual* tokens. In the `SIM-JACK` setup, a pre-trained RoBERTa model is used for feature transformation. Further details are available in the appendix.

Datasets: The *Adult* dataset contains demographic attributes such as age, race, and gender, with the goal of predicting if an individual's income surpasses a specific threshold. We use gender as the sensitive attribute and standardize all features. The *CivilComments* dataset's task is to determine if a comment is toxic. We've noted, as have others, that model performance can vary across demographics. In our experiments, we identify `Muslim` as the sensitive group and apply our approach to ensure fairness according to the EO and DEO metrics.

Results: The *Adult* dataset results are in Figure 1. `SIM-JACK` consistently out-performs `FairIJ` [16] (the *Drop-K* variant of our approach) in error rate for comparable fairness metrics. We also compare against strong bias mitigation baselines from the literature. For post-processing methods, we include FST [18] and HPS [7] which operate on the logits or probability of the predictions and transform them to achieve better fairness. These often are restricted in the amount of change that that they can impose on pre-trained models, as also seen in Figure 1. For in-processing methods, we compare against a HGR [13] based regularizer and a mixup based approach, *FairMixup* [4]. We also compare to fine-tuning to directly minimize the fairness measure in Table 2 (appendix). Both in-processing and fine-tuning approaches are able to improve fairness but substantially increased error rates. This highlights the benefit of the re-wieghting based approaches of first learning an ERM model and carefully editing it to achieve fairness. In the case of *CivilComments*, we compare the three rewieghting approaches — `SIM-JACK`, `I-JACK`, and `Fair-IJ`. We find that `SIM-JACK` typically provides a better fairness-accuracy tradeoff compared to `Fair-IJ` and `I-JACK` by either producing more accurate predictions at the same level of fairness or producing more fair predictions at the same accuracy depending on the number of virtual tokens used. The results are available in the appendix (Table 3).

[3] recall that we are in the vicinity of a stationary point, $\sum_n \mathbf{g}_n \approx 0$, hence the covaraiance is $\approx E[\mathbf{g}_n \mathbf{g}_n^T]$

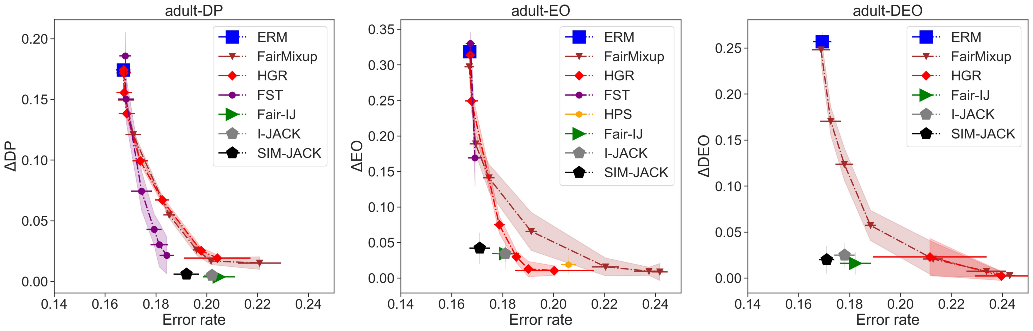

Figure 1: Performance of various bias mitigation algorithms on the *Adult* dataset averaged over 10 runs. The error bars represent two standard errors.

5 Discussion

In this paper, we took a closer look at influence function based approaches to fairness and showed that various existing approaches can be unified under a single optimization framework. We additionally provided insights into the favorable fairness-accuracy tradeoffs of these approaches.

While effective in many cases, these approaches to fairness editing are local in nature, wherein they search for edited models within the vicinity of the empirical risk minimized model. Their utility might be limited when finding an useful edited model requires hopping between different minima of the loss. Additionally, their reliance on the loss landscape's curvature can pose computational challenges. Yet, even crude (*diagonal*) curvature approximations can result in valuable model edits.

References

[1] Juhan Bae, Nathan Hoyen Ng, Alston Lo, Marzyeh Ghassemi, and Roger Baker Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, 2022.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[3] Henry C Bendekgey and Erik Sudderth. Scalable and stable surrogates for flexible classifiers with fairness constraints. In *Advances in Neural Information Processing Systems*, volume 34, pages 30023–30036, 2021.

[4] Ching-Yao Chuang and Youssef Mroueh. Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*, 2021.

[5] Soumya Ghosh, Will Stephenson, Tin D. Nguyen, Sameer Deshpande, and Tamara Broderick. Approximate cross-validation for structured models. *Advances in Neural Information Processing Systems*, 33:8741–8752, 2020.

[6] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[7] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016.

[8] L. Jaeckel. The infinitesimal jackknife, memorandum. Technical report, MM 72-1215-11, Bell Lab. Murray Hill, NJ, 1972.

[9] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *IEEE International Conference on Data Mining*, pages 924–929, 2012.

[10] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

[11] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications.* Springer Science & Business Media, 2002.

[12] Pranay K. Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R. Varshney, and Ruchir Puri. Bias mitigation post-processing for individual and group fairness. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2847–2851, 2019.

[13] Jérémie Mary, Clément Calauzenes, and Noureddine El Karoui. Fairness-aware learning for continuous attributes and treatments. In *International Conference on Machine Learning*, pages 4382–4391, 2019.

[14] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. *Advances in Neural Information Processing Systems*, 30, 2017.

[15] Brianna Richardson, Prasanna Sattigeri, Dennis Wei, Karthikeyan Natesan Ramamurthy, Kush Varshney, Amit Dhurandhar, and Juan E Gilbert. Add-remove-or-relabel: Practitioner-friendly bias mitigation via influential fairness. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 736–752, 2023.

[16] Prasanna Sattigeri, Soumya Ghosh, Inkit Padhi, Pierre Dognin, and Kush R. Varshney. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. In *Advances in Neural Information Processing Systems*, 2022.

[17] Andrew Silva, Rohit Chopra, and Matthew Gombolay. Cross-loss influence functions to explain deep network representations. In *International Conference on Artificial Intelligence and Statistics*, pages 1–17. PMLR, 2022.

[18] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio Calmon. Optimized score transformation for fair classification. In *International Conference on Machine Learning*, pages 1673–1683, 2020.

[19] Yilun Xu, Hao He, Tianxiao Shen, and Tommi S. Jaakkola. Controlling directions orthogonal to a classifier. In *International Conference on Learning Representations*, 2021.

[20] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*, pages 1171–1180, 2017.

A Practical Considerations

We note that the machinery presented in the main paper assumes that we are at a stationary point of the $\mathcal{L}(\boldsymbol{\theta})$, i.e., $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = 0$. However, modern machine learning models, trained with SGD, and potentially employing early stopping rarely satisfy this criteria. However, recent work [5] establish that the error in Equation 3 grows smoothly with distance from the stationary point, allowing the analytical influence expression to still be useful in the vicinity of a stationary point. Moreover, [1] show that influence functions computed at non-converged solutions, when employing a Fisher information matrix with a dampner as an approximation to the Hessian, can be viewed as linear approximations to a particular proximal Bregman response function. This is the view we adapt in this paper and approximate the Hessian with the Fisher (and add a diagonal dampner). In preliminary experiments with the 100 unit MLP on *Adult* data we further found a conjugate gradient based approach for computing the inverse Fisher vector products needed by our influence based re-estimation schemes did not consistently out-perform a diagonal *empirical* Fisher matrix approximation to the Hessian. Since, the diagonal approximation is efficient to compute, store, and invert, our experiments use it to explicitly perform the required inverse matrix vector products, doing away with the expensive conjugate gradient method.

B Experimental Details

Dataset statistics Table 1 summarizes the different datasets used in this paper.

Table 1: A summary of different datasets used in Section 4.

Dataset	$\mathcal{Y}$	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
Adult	$0, 1$	21815	10746	12661
Civil Comments	$0, 1$	269038	45180	133782

Fairness experiments For our experiments on the *Adult* dataset we use 21815, 10746 and 12661 as the training, validation and test instances, respectively. Each training run uses a random split of the datasets and we average the results over 10 runs. We search for λ among 50 uniformly chosen values in the range $0.01 - 10$ and pick the one with the lowest fairness discrepancy as measured on the validation set. Each training run, including searching for the optimal λ took less than 2 hours on a single NVIDIA A100 Tensor Core GPU. Table 2 additionally compares fine-tuning for fairness against `SIM-JACK`. Observe that while fine-tuning can lower the fairness disparity this comes at a substantial cost — much lower accuracy compared to ERM and `SIM-JACK`.

Method	Test Error	Test DP	Test EO	Test DEO
`SIM-JACK`	0.190±0.020	0.005 ± 0.002	0.040 ± 0.022	0.020 ± 0.015
Fine-tuning-DP	0.240±0.010	0.003 ± 0.001		
Fine-tuning-EO	0.240±0.002		0.003 ± 0.007	
Fine-tuning-DEO	0.350±0.210			0.020 ± 0.023

Table 2: Results of fine-tuning with respect to different fairness objectives on the ADULT test split. Fine-tuning-DP, Fine-tuning-EO, Fine-tuning-DEO are used to indicate pre-trained models fine-tuned to minimize DP, EO, and DEO on the ADULT validation split. We compare against `SIM-JACK` trained on the same validation split. We note that while fine-tuning does reduce the fairness discrepancy it is trained to minimize it also produces substantially less accurate models, resulting in noticeably worse fairness-accuracy trade-offs.

For the CivilComments experiments, we used soft prompt-tuning to adapt T5-base models. We experimented with different number of virtual tokens — 20, 40, and 60. We used an auxiliary model, RoBERTa-base, for computing the similarity matrix $\mathbf{C}$ needed by `SIM-JACK`. In particular, we use the RoBERTa-base model's hidden-state of the first token of the last layer to encode each training instance. In order to find the optimal λ, we do a logarithmic search between the range of 0.01 and 50. The numbers in Table 3 reports the performance of `SIM-JACK`, `I-JACK`, `Fair-IJ` for each choice of the number of virtual tokens averaged over five random initializations.

C Proofs

Proposition C.1 (**Drop-K Upweight-M weighting scheme**). *Let $\mathcal{W} = \{0, 1, 2\}^N$ be the set of all N dimensional ternary vectors, and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the Hamming distance between $\mathbf{w}$ and $\mathbf{1}$, then Equation 9 is minimized by setting the weights of the K largest positive influence training instances to zero and the M smallest negative influence training instances to two.*

Proof. Plugging in the constraint and the proximal regularizer in Equation 9, we have,

$$\mathbf{w}^* = \underset{\mathbf{w}\in\{0,1,2\}^N}{\operatorname{argmin}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})\frac{1}{\lambda}\mathcal{H}(\mathbf{w}, \mathbf{1}) = \underset{\mathbf{w}\in\{0,1,2\}^N}{\operatorname{argmin}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})\frac{1}{\lambda}\sum_{n=1}^{N} |w_n - 1|,$$

Metric / Method	ERM	`Fair-IJ`	`I-JACK`	`SIM-JACK`
		$\lvert V_t\rvert = 20$		
BA	0.61±0.01	0.57±0.00	0.57±0.00	0.58±0.02
ΔEO	0.14±0.01	0.10±0.00	0.10±0.00	0.10±0.02
BA	0.61±0.01	0.57±0.01	0.57±0.00	0.59±0.02
ΔDEO	0.14±0.01	0.09±0.01	0.09±0.00	0.09±0.01
		$\lvert V_t\rvert = 40$		
BA	0.62±0.01	0.57±0.00	0.57±0.00	0.62±0.03
ΔEO	0.16±0.02	0.11±0.00	0.11±0.01	0.11±0.02
BA	0.62±0.01	0.57±0.00	0.57±0.00	0.62±0.02
ΔDEO	0.16±0.02	0.11±0.00	0.11±0.00	0.08±0.03
		$\lvert V_t\rvert = 60$		
BA	0.61±0.02	0.57±0.00	0.55±0.03	0.56±0.02
ΔEO	0.12±0.01	0.09±0.01	0.07±0.03	0.06±0.02
BA	0.61±0.02	0.57±0.00	0.57±0.00	0.57±0.00
ΔDEO	0.12±0.01	0.09±0.01	0.09±0.00	0.06±0.02

Table 3: Comparison of bias mitigation on *CivilComments* dataset with sensitive attribute *muslim*. We report balanced accuracy (BA), EO, and DEO across three runs for different choices of the number of virtual tokens $|V_t|$. The error bars report two standard errors over five runs.

the second equality follows from our choice of encoding a ternary vector using $\{0, 1, 2\}$. Following arguments analogous to the ones made in proving Proposition C.1, we arrive at the following minimization problem,

$$\min_{\mathbf{w}\in\{0,1,2\}^N} \sum_{n=1}^{N} \mathbb{1}[\mathcal{I}_{\mathcal{F},n} > 0]\big(\mathcal{I}_{\mathcal{F},n}(w_n - 1) + \frac{1}{\lambda}|w_n - 1|\big) + \sum_{n=1}^{N} \mathbb{1}[\mathcal{I}_{\mathcal{F},n} < 0]\big(\mathcal{I}_{\mathcal{F},n}(w_n - 1) + \frac{1}{\lambda}|w_n - 1|\big),$$

where $\mathbb{1}[a > b]$ is an indicator function that takes a value of one when $a > b$ and zero otherwise. Next, observe that that setting $w_n = 0$ when $\mathcal{I}_{\mathcal{F},n} > 0$ reduces the objective by $-\mathcal{I}_{\mathcal{F},n}$ at the cost of $\frac{1}{\lambda}$, while setting $w_n = 2$ when $\mathcal{I}_{\mathcal{F},n} < 0$ reduces the objective by $-|\mathcal{I}_{\mathcal{F},n}|$ also at the cost of $\frac{1}{\lambda}$. The objective is thus minimized by sorting the influences by their magnitudes and proceeding in descending order, setting $w_n = 0$, if $\mathcal{I}_{\mathcal{F},n} > 0$ and $w_n = 2$ if $\mathcal{I}_{\mathcal{F},n} < 0$ until $\frac{K}{\lambda} + \frac{M}{\lambda} > \sum_{n=1}^{K+M} |\mathcal{I}_{\mathcal{F},n}|$, where K is the number of entries of $\mathbf{w}$ set to zero and M is the number of entries of $\mathbf{w}$ set to 2. □

Proposition C.2 (**`I-JACK` weighting rule**). *Let $\mathcal{W} = \mathbb{R}^N$ and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the squared ℓ_2 norm, $\frac{1}{2}||\mathbf{w} - \mathbf{1}||_2^2$, then Equation 9 is minimized by setting,*

$$\mathbf{w}^* = -\lambda \nabla_{\mathbf{w}} \mathcal{F}^{\Delta}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w})\Big|_{\mathbf{w}=\mathbf{1}, \boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} + \mathbf{1} = -\lambda \mathcal{I}_{\mathcal{F}} + \mathbf{1}.$$

Proof. We arrive at the result by setting the gradient $\nabla_{\mathbf{w}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) \frac{1}{2\lambda}(\mathbf{w} - \mathbf{1})^T(\mathbf{w} - \mathbf{1})$ to zero and rearranging terms,

$$\nabla_{\mathbf{w}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) \frac{1}{2\lambda}(\mathbf{w} - \mathbf{1})^T(\mathbf{w} - \mathbf{1}) = 0 \implies \mathcal{I}_{\mathcal{F}} + \frac{1}{2\lambda} 2\mathbf{w} - \frac{1}{\lambda}\mathbf{1} = 0$$
$$\implies \mathbf{w} = -\lambda \mathcal{I}_{\mathcal{F}} + \mathbf{1}.$$

□

Proposition C.3 (**`SIM-JACK` weighting rule**). *Let $\mathcal{W} \in \mathbb{R}^N$ and $\mathbb{D}(\mathbf{w}, \mathbf{1})$ be the squared Mahalanobis distance $\frac{1}{2}(\mathbf{w} - \mathbf{1})^T C^{-1}(\mathbf{w} - \mathbf{1})$, where $C \in \mathcal{S}_+^{N\times N}$ is a symmetric positive definite matrix, then Equation 9 is minimized by setting,*

$$\mathbf{w}^* = -\lambda C \mathcal{I}_{\mathcal{F}} + \mathbf{1}.$$

Proof. We arrive at the result by setting the gradient $\nabla_{\mathbf{w}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) \frac{1}{2\lambda}(\mathbf{w} - \mathbf{1})^T C^{-1}(\mathbf{w} - \mathbf{1})$ to zero and rearranging terms,

$$\begin{aligned} &\nabla_{\mathbf{w}} \bar{\mathcal{F}}^{\Delta+}_{\mathcal{D}_{\text{val}}}(\boldsymbol{\theta}^*(\mathbf{w}), \mathbf{w}) \frac{1}{2\lambda}(\mathbf{w} - \mathbf{1})^T C^{-1}(\mathbf{w} - \mathbf{1}) = 0 \\ \Longrightarrow\ &\mathcal{I}_{\mathcal{F}} + \frac{1}{2\lambda} 2C^{-1}\mathbf{w} - \frac{1}{\lambda} C^{-1}\mathbf{1} = 0 \\ \Longrightarrow\ &\mathbf{w} = -\lambda C \mathcal{I}_{\mathcal{F}} + \mathbf{1}. \end{aligned}$$

□