Optimisation for Fairness

Sachit Gaudi
Department of Computer Science
Michigan State University
gaudisac@msu.edu

Abstract

In this document, I tried to list down the different techniques for solving fairness problem as mentioned in Lahoti et al.. To solve this task, I tried to look for approaches used in the general fairness task - invariant with respective label - Sadeghi et al., Sadeghi et al. - with an idea that we can relax the label constraint going forward. The general approach is to play an adversarial game, In which adversary tries to gain information of sensitive label and encoder tries to make it difficult for the adversary. These games are generally zero-sum games / min-max problems. So to understand, I have studied min-max optimisation. Some general comments are made how zero-sum games with probability constraints lead to the re-weighting approaches which are widely used in solving the problem - Chai et al., Chai and Wang, Lahoti et al., Setlur et al..

1 Introduction

We want to move from an unfair space to a fair space. The question we are interested in exploring is How do we construct a fair space?

2 Mathematical Preliminaries

We define $\mathbb{W}, \mathbb{X}$ as the column space of W, X respectively. where W is the fair space and X as the input space. We want to learn the transformation $\mathbf{H}$ such that any vector in the subspace of $\mathbb{X}$ can be transformed to $\mathbb{W}$. $\mathbb{Z}, \mathbb{Y}$ denotes latent space and prediction space. and we use $\mathbb{S}$ for sampling and sensitive space appropriately.

2.1 Problem definition

Find the equation such that reconstruction error is minimum. To put formally,

$$\min_{H} \max_{||x|| \leq L} ||WHSX - X||^2 \tag{1}$$

If we construct know the fair space then this equation will give the optimal solution, but how to construct a fair space, We look at Class conditional permutation Idea in next section.

2.2 Class conditional permutation (CCP):

Construct a permutation matrix by interchanging the subset of features by other data point. For example, Initial dataset X and y are transformed to $\hat{X}, y$

$$X = \begin{bmatrix} x_1^0 & x_2^0 & x_3^0 \\ x_1^1 & x_2^1 & x_3^1 \\ x_1^2 & x_2^2 & x_3^3 \\ x_1^3 & x_2^2 & x_3^3 \end{bmatrix} \; y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\hat{X} = \begin{bmatrix} x_1^0 & \mathbf{x_2^1} & x_3^0 \\ x_1^1 & \mathbf{x_2^0} & x_3^1 \\ \mathbf{x_1^3} & x_2^2 & x_3^3 \\ \mathbf{x_1^2} & x_2^2 & x_3^3 \end{bmatrix} y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

The bold features pointed out are interchanged, the first 2 rows belong to class 0 and the rest belongs to class 1.

Does the basis formed after class conditional permutation form a fair space? No. For example, points [3,2] and [5,4] belongs to one class separated by linear boundary, y = x, Initially both points belong to one class but after CCP, new points [3,4] , [2,4] would lie on either side of the boundary and we are losing information on y classification.

From this conclusions, we cannot apply the CCP idea directly on the input space, rather we investigate assumptions for which CCP idea is valid. $z_i|y \perp z_j|y$ for all $i, j \in 1, 2, \cdots, r$. Also $z_l \perp\!\!\!\perp y$ for all $l \in k, k+1 \cdots, r$. Two ways we can ensure causal Independence.

- CCP to work, we need to have latent space should have class conditional independence. out of the 7 causal diagrams for 3 variable scenario identified in the Wang and Boddeti , only $z_1 \leftarrow y \rightarrow z_2$, All the other will leak information when conditioned on y.
- Remove r-k dimensions or Randomize r-k dimensions, it is equivalent to randomised experiment. **Is class conditional permutation equivalent to randomised experiment?**

Ideas to solve the above 2 problems

- Explore making the latent features independent. One idea was implemented in Sadeghi et al.
- One Idea to solve the problem with permutation is Re-bias mentioned in the section 4

Another central assumption with the permutation is $s \perp\!\!\!\perp Y$, i.e we cannot remove the dependence if it is correlated with Y.

2.3 Causality

The validity of the idea should be explored from the perspective of all the possible tools available to us. The idea should be consistent with along on the dimensions.

2.4 Duality

Solving any min max problems with constraints. Generally min-max problems are solved by analysing the equilibrium point. Generally this equilibrium point is a saddle point.

2.4.1 Theory of Duality

Principles of Duality is used to solve constrained optimisation involving multiple players. Lets formulate a multi-player games in Game theory, We then apply the duality to solve a multiplayer game, So what strategy - probability distribution across rows - should u and v decide on the given constraints.

Where players u and v play a game and P is an expected cost matrix, play u selects rows and player v selects columns. both players try to minimize the cost irrespective of other player choices.

$$\min_{\mathbf{u}; u \in S} \max_{\mathbf{v}; v \in S} \mathbf{u}^T \mathbf{P} \mathbf{v}$$

where S, is probability simplex. As a player, the strategy to have better performance for the worst choice by the other player, other player will choose worst possible outcome.

$$\min_{\mathbf{u}; u \in S} \max_{i \in 1, 2, \cdots, n} ((\mathbf{u}^T \mathbf{P})_i) \tag{2}$$

The duality of the equation is written as,

$$\begin{aligned} &\textbf{minimize } \epsilon \\ &\textbf{subject to } u^T\mathbf{P} \preceq \epsilon \\ &\qquad\qquad u \succeq 0 \\ &\qquad\qquad \Sigma u = 1 \end{aligned} \tag{3}$$

$$\begin{aligned} &\textbf{maxmize } \epsilon \\ &\textbf{subject to } \mathbf{P}v \succeq \epsilon \\ &\qquad\qquad v \succeq 0 \\ &\qquad\qquad \Sigma v = 1 \end{aligned} \tag{4}$$

The duality gap is given by $\epsilon_6 - \epsilon_5 \geq 0$, This can be seen as second player always has the advantage. In case of strong duality $\epsilon_6 - \epsilon_5 = 0$.

The conditions required for using sub-gradient methods are constraints and functions should be convex, convexity of the constraints is proven by the closed set condition, $u_1, u_2 \in C \implies \lambda u_1 + (1-\lambda)u_2 \in C$, given $0 \leq \lambda \leq 1$

Solving 3 using sub-gradient method and we apply projection techniques to improve the convergence as suggested in EE364b lets write Lagrangian and try to solve it.

$$\begin{aligned} &\textbf{minimize } \epsilon + \lambda^T(u^T\mathbf{P} - \epsilon) + \mu^T(-u - 0) + \gamma(\Sigma u - 1) \\ &\textbf{subject to } \lambda \succeq 0; \mu \succeq 0 \end{aligned}$$

The objective is to solve $\max_{\lambda,\mu} g(\lambda, \mu) = \inf_u[\epsilon + \lambda^T(u^T\mathbf{P} - \epsilon) + \mu^T(-u - 0) + \gamma(\Sigma u - 1)]$. with respective to the above constraints on λ, μ. For example, This EE364b suggests that constraints are also enforced in the projection space. One idea is to have gradients only towards the direction of equality constraints and not in the direction of it. So we nullify the component in the direction of constraint subspace, or project gradients in the orthogonal subspace. Suppose $Ax = b$ is the constraint, $(I - A(A^TA)^\dagger A^T)\Delta z$ is the modified gradient. and In our example, $\Sigma u = 1 \implies [1, 1, \cdots, 1]u = 1$, Similarly to project onto $\lambda \succeq 0 \implies \lambda = [\lambda]_+$

Algorithm 1 Constraint Sub gradient Optimisation

Require: $\lambda \succeq 0; \mu \succeq 0$
  $e_o \leftarrow 0$
  $\epsilon_{best} \leftarrow \infty$
  $u_{best} \leftarrow None$
  $\gamma_i, \gamma_o, lr_o \leftarrow 0.1$
  while $e_o \leq n$ **do**
    $e_i \leftarrow 0$
    while $e_i \leq n$ **do**
      $loss = \epsilon + \lambda^T(u^T\mathbf{P} - \epsilon) + \mu^T(-u + 0) + \alpha(\Sigma u - 1)$
      $A = [1, 1, \cdots, 1]$
      $u = u - \gamma_i(I - A(A^TA)^\dagger A^T)\frac{\partial \text{loss}}{u}$ # projection idea
      $e_i += 1$
    end while
      $g(\lambda, \mu, \alpha) = \epsilon + \lambda^T({u_*}^T\mathbf{P} - \epsilon) + \mu^T(-u_* - 0) + \alpha(\Sigma u_* - 1)$
      $\lambda_{e_{o+1}} = [\lambda_{e_o} + lr_o\frac{\partial \text{g}}{\lambda}]_+$; $\mu_{e_{o+1}} = [\mu_{e_o} + lr_o\frac{\partial \text{g}}{\mu}]_+$; $\alpha_{e_{o+1}} = \alpha_{e_o} + lr_o\frac{\partial \text{g}}{\alpha}$ #max g
      $lr_o = \frac{\gamma_i}{\sqrt{e_o}}$ # decreasing lr for convergence EE364b
      $e_o += 1$
    if $\epsilon <= \epsilon_{best}$ **then** $u_{best} = u; \epsilon_{best} = \epsilon$ # Not a descent algo, so need to store best value
    end if
  end while

Geometric perspective of the algorithm

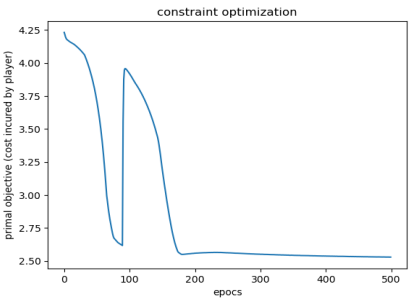
The Lagrangian constraint optimisation, In case of equality constraints, Geometric interpretation can be looked as to find a point on a level curve g(x) = k, where the gradient is flat to level surface or to put in a mathematical way, if there was a component of gradient in the direction level surface subspace, then that point is not a max/min, because we can move in the direction of gradient in the level surface and optimise the objective function further with satisfying the constraint. So but this argument, gradients on both surface are parallel to each other.$\Delta g(x) = \lambda \Delta f(x)$. To extend the same principle to the inequality constraints, we consider to enforce an additional constraint, $\lambda \geq 0$. Another perspective to look at it is Suppose your constraints are satisfied, then $\lambda < 0$, and by projection idea we will nullify it and if $\lambda > 0$, we need to optimise the main equation with $\min_x f(x) + \lambda g(x)$, maximising the dual can be interpreted as maximising the weight of unsatisfied constraint on surface of lower bound of primal.

Remember that this approach is not descent approach, where we end up reaching the minima, but these methods optimise the search space, with guarantees for convergence of the best solution which is given by the primal objective. more proof on convergence is provided in EE364b. General ideas for convergence are scheduling the learning rate.

2.4.2 Solving a Game theory problem

Solving a linear game theory problem with the above technique Suppose matrix cost matrix P in the min-max games is given below.

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 2 & 2 & 3 & 3 & 2 & 2 & 1 \\ 1 & 2 & 1 & 0 & 1 & 2 & 3 & 2 & 3 & 3 & 2 \\ 2 & 3 & 2 & 1 & 0 & 3 & 4 & 1 & 2 & 3 & 3 \\ 1 & 2 & 2 & 2 & 3 & 0 & 1 & 4 & 3 & 4 & 3 \\ 2 & 3 & 3 & 3 & 4 & 1 & 0 & 5 & 4 & 5 & 4 \\ 3 & 2 & 3 & 2 & 1 & 4 & 5 & 0 & 1 & 2 & 3 \\ 2 & 1 & 2 & 3 & 2 & 3 & 4 & 1 & 0 & 1 & 2 \\ 3 & 2 & 2 & 3 & 3 & 4 & 5 & 2 & 1 & 0 & 1 \\ 2 & 2 & 1 & 2 & 3 & 3 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$



As we can see the objective of the row player is to find optimal distribution on 11 choices, for any distribution chosen by col player. As you can see that there are multiple solutions exists to the above problem, but 2.5 is minimum that can be obtained, one solution is 1/2 , 1/2 bets in row 1 and 2. The above algorithm resulted in a solution with a cost of 2.53, 1.2%, off from the optimum value.

$$\begin{aligned} u_*^T &= [\ 0.22 \quad 0.21 \quad 0.057 \quad 0.0 \quad 0.0 \quad 0.15 \quad 0.078 \quad 0.0 \quad 0.28 \quad 0.0 \quad 0.0\ ] \\ \lambda_*^T &= [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.037 \quad 0.0 \quad 0.46 \quad 0.0 \quad 0.0 \quad 0.32 \quad 0.0\ ] \\ \mu_*^T &= [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.022 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0\ ] \\ \alpha_* &= -13.53 \end{aligned}$$

As we are utilising the projection idea all the constraints will be satisfied by the optimal. $\mu_*[3] \neq 0$, indicates that row 3 probability is slightly negative. and $\lambda_*[6], \lambda_*[9] \neq 0$, which indicates that 6,9 rows are optimising to find minima for itself in the case of max constraint.

2.4.3 Solving a multi player multi objective convex-concave problem

We solve the same problem as in EE364b, in convex-concave game setup. This problem is different from the previous problem in a way that, players control 2 different variables and operation on their on constraints. So to put it mathematically.

$$\begin{gathered} \max_u \min_v f(u, v) \\ \textbf{subject to } \Sigma u = U; \Sigma v = V; u, v \succeq 0 \end{gathered}$$

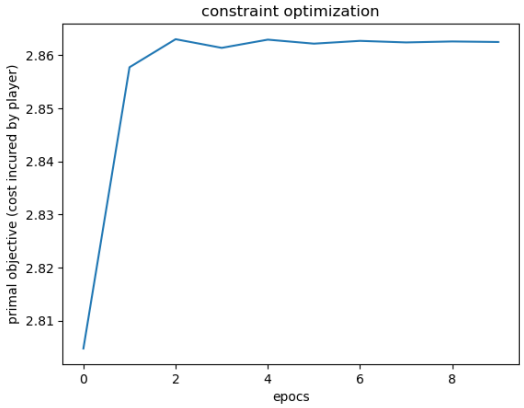
The above equation can be decomposed in to min and max with its constraints.

$$\begin{gathered} \max_u \left[\min_v f(u, v) \right] \\ \textbf{subject to } \Sigma u = U; u \succeq 0 \, [\textbf{subject to } \Sigma v = V; v \succeq 0] \end{gathered} \tag{5}$$

We can optimize the min and max constraints in an adversarial fashion, each convex optimization be it min or max can be solved by the above mentioned ideas. The 5 can be future written as

$$\begin{aligned} &\max_{u} f(u, v_*) && \min_{u} f(u_*, v) \\ &\textbf{subject to } \Sigma u = U; u \succeq 0 && \textbf{subject to } \Sigma v = V; v \succeq 0 \end{aligned} \tag{6}$$

The results obtained are similar to the ones obtained in the EE364b, and similar insights can be



drawn from the result.[1]

$$\begin{aligned} p_*^T &= [\ 2.8 \quad 2.3 \quad 2.6 \quad 0.38 \quad 2.9 \quad 0.0 \quad 2.6 \quad 2.3 \quad 1.4 \quad 2.7\ ] \\ n_*^T &= [\ 3.3 \quad 0.0 \quad 0.85 \quad 0.0 \quad 2.6 \quad 0.0 \quad 0.89 \quad 0.0 \quad 0.0 \quad 2.4\ ] \\ \lambda_{1*}^T &= [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0\ ] \\ \lambda_{2*}^T &= [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.03 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.023 \quad 0.0\ ] \end{aligned}$$

2.4.4 How can we solve Domain shift or adaptation problem?

Generally models are good at minimising the error on the training set. So the model performs well if the testing distribution is close to training distribution. This domain adaptation problem is more serious in semantic segmentation as pointed by the Truong et al.. If we train the model on the object detection task, smaller objects have less impact on the gradient loss and larger objects have huge loss in the semantic segmentation space. To fix this issue, Truong et al. have formulated as mentioned in 8. Idea is to re-balance the weights of pixels as well as use surrounding pixel information for semantic segmentation. However, math provided in the paper is not convincing argument to use transformer based auto regression for using surrounding pixel information.

To fix this problem in a robust way we can model it as a there should be a uniform loss suffered by all the classes / groups, which brings us to the techniques in the fairness domain, where the constraint is similar to demographic parity. Consider the below equation

$$\begin{aligned} &\min_{f,\epsilon} \quad \epsilon \\ &\textbf{subject to } L(f(x|s), y) - L(f_s(x|s), y) \leq \epsilon \end{aligned} \tag{7}$$

where $f_s(x|s)$ is the best model that fits for the group s. and L is the convex loss function. ϵ is the gap between the general model, which is trained on all groups data and the model trained specifically on group s. Always f_s will be the upper bound performance on who the general model performs.

The 7 can be geometrically looked as a zero sum game between all groups trying to minimize its error, and also the global agent f trying to minimise the training error.

[1] Duality experiments code

As we write the duality of the above equation, we see how it is equivalent to reweighing ideas.

$$\begin{aligned}
&\min_{f,\epsilon} \quad \epsilon + \int \lambda(s)[L(f(x|s),y) - L(f_s(x|s),y) - \epsilon] \\
&\textbf{subject to } \lambda(s) \geq 0 \quad \forall s \\
&\min_{f,\epsilon} \quad \epsilon[1 - \int \lambda(s)] + \int \lambda(s)[L(f(x|s),y) - L(f_s(x|s),y)] \\
&\textbf{subject to } \lambda(s) \geq 0 \quad \forall s
\end{aligned}$$

$$\begin{aligned}
&\min_{f,\epsilon} \quad \int \lambda(s)[L(f(x|s),y) - L(f_s(x|s),y)] \\
&\textbf{subject to } \lambda(s) \in P_s
\end{aligned} \tag{8}$$

Where P_s is the probability simplex, indicating $\lambda(s)$ is a probability function. As we can see $\epsilon[1 - \int \lambda(s)]; \quad \lambda(s) \geq 0 \quad \forall s$ optimisation is nothing but enforcing λ as probability over s. Solving 8, using the projected sub gradient methods lead us to the above mentioned iterative approach.

$$\begin{aligned}
f^* &= \inf \int \lambda(s) L(f(x|s),y) \\
g(\lambda(s)) &= \int \lambda(s)[L(f^*(x|s),y) - L(f_s(x|s),y)] \\
\lambda(s)_{t+1} &= \pi([\lambda(s)_t + \alpha \Delta g(\lambda(s))]_+)
\end{aligned} \tag{9}$$

Where π is a projection onto probability simplex. We start with training distribution,$\lambda(s)$ and then we find the function that minimizes with the initial distribution of data. but as the iteration progress, the distribution shifts adding more weight to the groups g, where the gap to the best performing model on the group is highest. This indicates that all the adversarial re-weighting approaches mentioned in Lahoti et al., Chai et al., can be linked to solving the zero-sum game with uniform errors across s, which is having a better demographic parity.

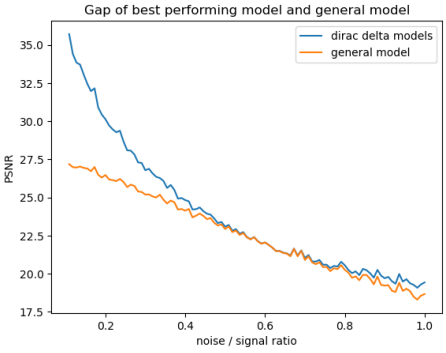


Figure 1: Problem: non uniform gap

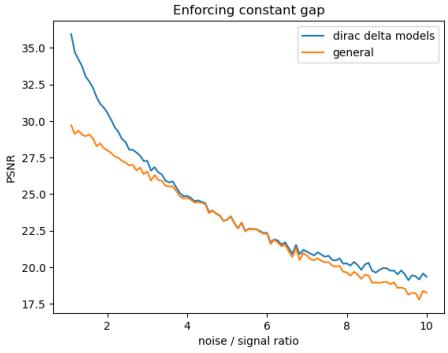


Figure 2: Solution: near uniform gap

To illustrate the above phenomenon, we perform the same experiment as in Gnanasambandam and Chan, Linear de-noising, $x = y + \sigma$ where the model task is to denoise σ, We choose a linear model, f(x) = a*x, and loss is simple reconstruction loss $||f(x) - y||^2$. From the results we observe, 3 db increase in reconstructed PSNR and the gap for the low noise groups seems to be decreasing. As Figure 3, indicates the game between the zero noise samples, and high noise samples to minimise the gap, if we minimize error for low noise samples, gap increases in the high noise samples. and finally equity is achieved, we find a saddle point of distribution which re-weights easy samples more compared to the hard samples, so that they have contribute similar amount in the gradient space, assuming large noise samples are hard samples, and contribute more to the loss.

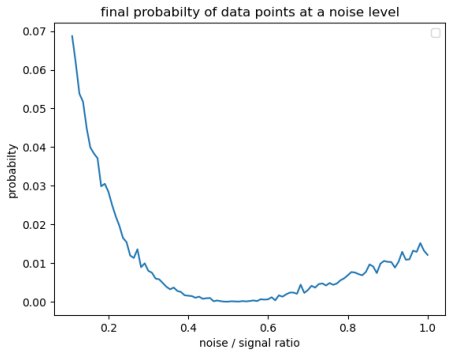


Figure 3: weights of different noise samples

Can we apply the uniform gap technique mentioned in Gnanasambandam and Chan to semantic segmentation problem of Truong et al.

2.4.5 With sensitive information (s) available:

Suppose we know the s, and s is linearly related to X, then the paper Ravfogel et al., paper formulated the problem as min max zero sum game between the projector (P) to in-variance space of s and θ which try to get sensitive information after separation.

$$\min_{\theta} \max_{P;P^TP=I} ||s - XP\theta||^2 \tag{10}$$

The sketch of proof the paper have followed is at the saddle point, $\frac{\partial .}{\partial \theta} = 0$ and the optimal θ, If for this θ, P can't optimise further then the equilibrium is achieved. i.e component of P should equal to zero, which they prove is P = Null space of $X^T s$. However this paper did not provide a closed form solution if s is a categorical features. The paper Ravfogel et al. applied multi objective convex, concave game, to apply the projection ideas to constraints, constraints need to be closed in convex space. but consider $P^TP = I$ is not closed under convex assumptions, 11.2, Solving with the 2.4.3, doesn't always guarantee a convergence.

Ignoring the convexity, To take full advantage of constraint optimisation, we need to find the projection function to enforce the constraints.

How to enforce $\mathbf{P^TP = I}$? To formulate the problem, we need to find matrix P such that $||P'^{-}P||^2$, is minimum, where P' is the optimal matrix obtained from max game. To put in the equation form, $\pi(P') = P \ni \min_{P:P^TP=I} ||P'^{-}P||^2$. To understand the geometric perspective, Imagine the eigen vectors of matrix P, the best reconstruction is squishing them onto unit hyper sphere, if there was some other component then the error will increase by the angle of rotation of eigen vectors, So, Suppose $P' = \Sigma\lambda_i v_i v_i^T$ then $P = \Sigma v_i v_i^T$ and the minimum error is $P = \Sigma(1-\lambda_i)^2$, This true only for $\lambda_i \geq 0.5$,$\lambda_i \in R$ To ensure this, $P = \frac{(P^T+P)}{2}$, P is symmetric matrix and have real eigen values. Same trick is applied in the paper Ravfogel et al..

11.1 derives the closed form solution to the above equation Complete the proof: 3^{rd} degree solution for P, prove the solution maximises (second derivative ≥ 0) and satisfies constraints. (θ_*, P_*) is a saddle point

2.4.6 Invariant representation

As enforcing independent constraints on latents used by Sadeghi et al., We can model as latents should have identity covariance matrix, Suppose $\mathbf{Z}$, denotes latents. the drawback of the below approach is that, it will only ensure there is no linear dependency.

$$(\mathbf{Z} - \bar{\mathbf{Z}})^T(\mathbf{Z} - \bar{\mathbf{Z}}) = \mathbf{I} \tag{11}$$

This equation can be enforced as hard constraint, which can be solved by a Lagrangian multiplier. This equation can also be enforced as a soft constraint, and tune the hyper parameters to obtain the

trade-off. Similar idea of enforcing constraint in the loss function is explored in the Zbontar et al. Suppose you pass the features X, to a encoder that gives latent Z. The equation we try to solve from the paper Sadeghi et al.

$$\begin{aligned} \min_{\theta_E, W_y, b_y} \quad & ||W_y\theta_E X + B_y - Y||^2 \\ \textbf{such that} \quad & \inf_{W_s, B_s} ||W_s\theta_E X + B_s - S||^2 \geq \epsilon \end{aligned}$$

This is very hard constraint optimisation to solve as $g(\lambda) = \lambda(\epsilon - f(\theta_E, W_y, b_y, W_s, B_s))$ the λ be either 0, ∞, which is unstable, So we can modify the equation as

$$\min_{\theta_E, W_y, b_y, W_s, b_s} \quad \lambda||W_y\theta_E X + B_y - Y||^2 + (1-\lambda)||W_s\theta_E X + B_s - S||^2 \tag{12}$$

We can solve the above equation, with either the matrix methods, or we can also solve with the eign sum notion as done for the **??** both give the same result and the convexity proofs are also similar to 11.2. We can remove W_y, b_y, W_s, b_s from the closed form solution of linear regression model, where the irremovable loss $W_y X$ any vector in W_y space can be represented, So irremovable error is orthogonal to W_y. The above equation can be modified as Assume $\theta_E X = M$ and $P_{M\perp} = (I - M(M^TM)^\dagger M^T)$

$$\begin{aligned} \min_{\theta_E} \quad & \lambda||(I - M(M^TM)^\dagger M^T)Y||^2 - (1-\lambda)||(I - M(M^TM)^\dagger M^T)S||^2 \\ \min_{\theta_E} \quad & trace(P_{M\perp}(\lambda YY^T - (1-\lambda)SS^T)P_{M\perp}) \\ \min_{\theta_E} \quad & trace(P_M((1-\lambda)SS^T - \lambda YY^T)P_M) \end{aligned}$$

trace(AB) = trace(BA) if dim A is mxn and dim B is nxm and $P^2 = P$ as P is a projection matrix

$$\min_{\theta_E} \quad trace(P_M[(1-\lambda)SS^T - \lambda YY^T])$$

As we know that $M = \theta_E X \in$ rowspace(X), lets construct orthonormal row space of X by Gram-Schmidt method let it be L_x, $M = L_x G_E = \theta_E X$, there exists some combination in orthonormal space of L_x, from properties of L_x, $L_X^T L_x = I$

$$\min_{\theta_E} \quad trace(L_x G_E (G_E^T G_E)^{-1} G_E^T L_x^T [(1-\lambda)SS^T - \lambda YY^T])$$

We don't want any G_E but orthonormal vectors which are subspace of rowspace of X / subspace of some rotation of L_X, which we can assume one solution of all the solutions is $G_E^T G_E = I$

$$\min_{\theta_E; G_E^T G_E = I} \quad trace(L_x G_E G_E^T L_x^T [(1-\lambda)SS^T - \lambda YY^T])$$

from the above property of trace,

$$\min_{\theta_E; G_E^T G_E = I} \quad trace(G_E^T L_x^T [(1-\lambda)SS^T - \lambda YY^T] L_x G_E)$$

This optimisation is similar to the min version of PCA so the solution of G_E is negative eigen values of $B = L_x^T[(1-\lambda)SS^T - \lambda YY^T]L_x$

Extending this approach to handle multiple modes of dependency. Sadeghi et al..

If the dependency is captured by covariance then it is nothing but capturing the linear dependency, equivalent to fitting a line. Gaussian kernel can be seen as graph, with nodes connected by the Inverse distance metric (either Gaussian or any inv dist metric). why cant we work with the same formulation as the earlier work to fit linear model to $\tilde{X}$? .

Assume we want to reformulate 12 to change the formulation with Dep function

$$\max_{\theta_E} \quad \lambda Dep(f_{\theta_E}(\tilde{X}), Y) - (1-\lambda) Dep(f_{\theta_E}(\tilde{X}), S) \tag{13}$$

Where $\tilde{X}_{ij} = K(x_i, x_j)$.

Same derivations apply to this scenario, but one additional regulariser is independence of $\theta_i X \perp\!\!\!\perp \theta_j X$ and $\theta_i \perp\!\!\!\perp \theta_j$ Once we expand the Dep equation in terms X,Y,S we end up with a Raleigh coefficient equation that can be solved with closed form like 11.3 or can use Rayleigh Quotient Iteration algorithm mentioned in TrefethenBau.

Can I solve the above equation with iterative approach, To impose the constraints as $G^T G = I$, we can have independent constraint on latent as done in Zbontar et al.. But it is common practice to use the ARL with no closed form solutions and still work with convergence as used by Ravfogel et al.

Try this experiment on non-linear functions to prove that iterative algorithms also show convergence and if the dep is linear then the non linear function also gives the same solution as above.

2.4.7 Duality in Fairness Problems without sensitive information

From this understanding lets solve 1 as already solved by Eldar and Dvorkind but here I give out a different version of proof.

$$\min_H \max_{||x|| \leq L} ||WHS^* x - x||^2$$

$$\min_H \max_{||x|| \leq L} ((WHS^* - I)x)^T ((WHS^* - I)x)$$

$$\min_H \max_{||x|| \leq L} x^T (WHS^* - I)^T (WHS^* - I)x$$

let $(WHS^* - I) = A$, the max equation becomes maximum Eigen value of A

$$\min_A \max_{eigval} A^T A$$

As the matrix $A^T A$ is symmetric positive definite the condition can be is equal to solving,

$$A = 0$$

Only this matrix has smallest largest positive eigen values which is 0, Eigen values of

$$WHS^* = I$$
$$H = W^\dagger S^{*\dagger}$$
$$H = (W^T W)^{-1} W^T S (S^T S)^{-1}$$

So

$$WHS^T = W(W^T W)^{-1} W^T S (S^T S)^{-1} S^T \tag{14}$$

Therefore, 14 becomes the closed form solution to the min max problem, Assuming W,S are full column rank matrices. The challenge is how do we construct fair space.

Approach 2: We can also think about this problem in a different way, $\mathbf{Wy}$ will always lie in the subspace of W, so if $W \perp$ is irreducible error in x, So solving similarly 15 will give the same result. We can also observe that we can remove $P_{W\perp} x$, out of the norm as it is projection matrix.

$$\min_H \max_{||x|| \leq L} ||WHS^* x - P_W x||^2 \tag{15}$$

.

2.5 Projection Ideas

Minimum projection error is given as the error for the projected vector should be orthogonal to that subspace. Vector that is present in a subspace is written by Wy, this gives, any vector in W.

$$W^T (Wy - y) = 0 \tag{16}$$

$$Wy = W(W^T W)^\dagger W^T y$$

So $W(W^TW)^\dagger W^T$ is the orthogonal projection. Instead for the constrained sampling as mentioned in Sadeghi et al. We want all the error to be orthogonal to the input space, S, this way you are maximising when projected to fair subspace W.

$$S^T(Wy - y) = 0 \tag{17}$$

$$Wy = W(S^TW)^\dagger S^T y$$

If we take the previous notation of projections in duality,

$$WHS^* = W(S^TW)^\dagger S^T$$

3 Max margin: SVM, connection to graphs

Suppose you have a graph G(V,E), for the problems such as max-flow, travelling salesman problems, requires solving constraint optimisation. For solving these NP-Hard and NP-complete problems, we need to formulate the problem as a constraint optimisation and use the techniques mentioned below. In optimisation problem we show that any primal optimisation can be written as max flow and the dual as min-cut.

Questions we are interested in exploring are

- Can I solve max flow and min cut problem with optimisation?
- Can I extend it to the TSP algorithm and see how far off the optimisation leads to?
- How can I interpret kernel SVM as a fully connected graph.
- How can show that kernel SVM as a max flow and min-cut?

3.1 Optimisation Algorithms

4 Re-biasing: One method to debias without s

The paper Bahng et al. proposes a new method based on the nature of bias in the models, Suppose M_1, M_2 are multiple models and these models capture the bias in X, i.e learn spurious features. Then we can make our model M independent to learned representation M_1, M_2. This idea is further extended to video by shuffling frames as spurious learnt model. We can further extend this idea to any model M_1 which learns spurious attributes it can be $M_1(X) = S$, which will make the problem as if solving with sensitive label information.

How to create $\mathbf{M_1}, \mathbf{M_2}$? The way they modeled in the Image classification models is based on the assumption that features learnt small receptive fields are biased $M_1(X) = y$,where M_1 is modeled as kernel of size 1x1, which will learn pixel level features like color. and M is modeled as traditional 3x3 kernel. Similarly for CelebA they modeled M_1 as 3x3 and M as 7x7 kernel.

The objective function is

$$\begin{aligned} \max_{\alpha} \min_{\theta} \quad & Dep(f_\theta(X), g_\alpha(X)) \\ \textbf{such that} \quad & L(f_\theta(X), y) \leq \epsilon \\ & L(g_\alpha(X), y) \leq \epsilon \end{aligned}$$

This can be seen as multi objective optimisation as mentioned in 2.4.3 and can be solved with alternating gradient decent.

One drawback of this approach is to come up with $M_1, M_2 \cdots$ functions and it is not always guaranteed that these features are completely independent to the task and this not a complete concave-convex optimisation. As $L(g_\alpha(X), y)$ is convex.

5 UDA: Unsupervised Domain Adaptation

- Can I take any ideas from UDA to apply for fairness.

Gaps in understand duality and projection ideas

- Does dual ascent algorithm mentioned in Gnanasambandam and Chan give the same optimal result?
- Does the primal and dual transformation applicable only in the case of strong duality? What happens when there is weak duality?
- How do we know if a problem has a strong / weak duality?
- Understand max volume min cut problem of duality
- Study EE364b barrier method for convex - concave games.
- If s is known and W is ortho-normal basis to the decision boundary of s

6 Related Works[Incomplete]

6.1 Pre Processing

Remove the component of X from the subspace of S. This is achieved by fitting linear regression between every feature of X and S, this way of implementation is available in fairlearn library. Orthogonal projection onto S space, or Least squares projection of X on S, or gram Schmidt give have the same impact. We stick with least squares because its gives us the parametric form, which can be applied at test time. The $\hat{X}$ is the feature space free from S information, and it is orthogonal to S

$$\hat{X} = (\mathbf{I} - \mathbf{S}(\mathbf{S^T S})^{-1}\mathbf{S^T})\mathbf{X}$$

Suppose the solution to least squares is given by W

$$W = (\mathbf{S^T S})^{-1}\mathbf{S^T})\mathbf{X}$$
$$\hat{X} = \mathbf{X} - \mathbf{SW}$$

6.2 In Processing

The big idea is to train model in an adversarial way to remove the S information. This can be achieved by running a zero-sum game between predictor which tries to maximise predictive capabilities of latent Z, and the adversary tries to predict sensitive attribute. We will examine Adversarial technique as proposed by Zhang et al..

6.3 Post Processing

One of the popular post processing ideas is to set the treshold based on the objective function we want to look

6.4 Metrics

Demographic Parity (DP):

Equalised odds (EOD):

Equalised opportunity (EOP) :

6.5 Analysis

We want understand the impact of Pre, In, and Post Processing, So we will run experiments on Folktables dataset Ding et al.. We study how each of the methods work.

Table 1: Classical Fairness (Folktables)

Method	Accuracy	Worst group Accuracy	DP	EOD	EOP
Invariant Rep Sadeghi et al.	0	0	0	0	0
Invariant Alternating gradient decent	0	0	0	0	0
Linear Adv (LA)	0	0	0	0	0
Pre-Process + LA	0	0	0	0	0

7 experiments

8 Enforcing Priors [CVPR 2023]

The problem of out-of-distribution sampling is equivalent to having a prior of data distribution. Suppose if you have a prior distribution of data then How can you enforce it in a neural network. Truong et al., paper have explored the idea of enforcing constraint on How can smaller object, which have less prediction pixels, likewise pointing to the problem of class imbalance in the pixel space.

8.1 Mathematical formulation

$$\hat{y} = f(X;\theta)$$
$$min_\theta \mathop{\mathbb{E}}_{P_{XY}\sim X\times Y}(L(\hat{y},y))$$

We can say that sampling x is equivalent to sampling $\hat{y}$. So $P_{XY} \sim P_{\hat{Y}Y}$

To eliminate the class imbalance, we can sample the loss from the balanced class data. Let Q_{XY} is the true prior distribution we want to enforce, In case of class-imbalance or Out-of-distribution models, $Q_{XY} \sim U$, where U is a uniform distribution.

$$min_\theta \mathop{\mathbb{E}}_{Q_{\hat{y}y}\sim \hat{Y}\times Y}(L(\hat{y},y))$$

$$min_\theta \mathop{\mathbb{E}}_{P_{\hat{y}y}\sim \hat{Y}\times Y}(L(\hat{y},y) * \frac{Q_{\hat{y}y}}{P_{\hat{y}y}}) \tag{18}$$
$$min_\theta \mathop{\mathbb{E}}_{P_{\hat{y}y}\sim \hat{Y}\times Y}(L(\hat{y},y)) + KL(Q_{\hat{y}y}, P_{\hat{y}y}) \tag{19}$$

Key assumption made in the paper Truong et al. is $\hat{y} \perp\!\!\!\perp y$ and also we assume the labels in the data are balanced, i.e In every batch of training we sample one from every class.

$$P_{\hat{y}y} = P_{\hat{y}} * P_y; Q_{\hat{y}y} = Q_{\hat{y}} * P_y; \frac{Q_y}{P_y} = 1$$

If we make such assumption then, equation 18 will reduce to

$$min_\theta \mathop{\mathbb{E}}_{P_{\hat{y}y}\sim \hat{Y}\times Y}(L(\hat{y},y) * \frac{Q_{\hat{y}}}{P_{\hat{y}}})$$

Applying log or 19 and by using the convex property of log, will give us upper bound

$$\geq min_\theta \mathop{\mathbb{E}}_{P_{\hat{y}y}\sim \hat{Y}\times Y}(L(\hat{y},y) + \mathop{\mathbb{E}}_{P_{\hat{y}y}\sim \hat{Y}\times Y}(\ln\frac{Q_{\hat{y}}}{P_{\hat{y}}}))$$

If we assume $Q \sim U$ then the second term is equivalent to maximising the entropy. This property seems to be trivial and cannot be used in general settings, suppose we have situation where we can

balance the segmentation pixels, smaller objects have smaller pixel prediction, In such situation this equation will be helpful. First term in the equation is minimising the supervised loss, second term is the regularizer / enforcing prior constraints.

$$P(\hat{y}) = \Sigma_{k=0}^{224\times224} P(\hat{y}^k) P(\hat{y}^{\setminus k} | \hat{y}^k)) \tag{20}$$

where k is each pixel classification in semantic segmentation. Similar to log convexity we will have upper bound.

$$\mathop{\mathbb{E}}_{P_{\hat{y}y} \sim \hat{Y} \times Y} (\ln \frac{Q_{\hat{y}}}{P_{\hat{y}}})) \geq \mathop{\mathbb{E}}_{P_{\hat{y}y} \sim \hat{Y} \times Y} (\Sigma_{k=0}^{224\times224} ln(\frac{Q}{P}(\hat{y}^k)) + \Sigma_{k=0}^{224\times224} ln(\frac{Q}{P}(\hat{y}^{\setminus k} | \hat{y}^k))) \tag{21}$$

Similar to our earlier assumptions $Q \sim U$ the first term is maximising the entropy for each pixel, Second term is regressive loss, which is equivalent to predict only from context.

9 Zero-sum games

9.1 General solution to zero sum games

Generally the solution to zero-sum games are formalized as min-max problems, this interpreatation dates back to the Adaboost paper. Can we use inspiration from the Adaboost, and dynamically re-weight the samples and formalise the conditions and bounds from the original Adaboost paper.

- Read Adaboost paper and formalise the problem statement, look at the bounds and conditions they are using and take inspiration from them.

10 Ideas

10.1 Actionable Items and next steps

- Mathematically formulate the problem.
- How do we combine Permutation idea 2.2 with 2.4.4 and use multi objective techniques mentioned in the section 2.4 to tame the trade-off curve.
- Can I implement and test the alternating gradient decent as mentioned in 2.4 to the Sadeghi et al.. Does this guarantee a convergence with right set of constraint?
- What happens to the accuracy if there is a shift in training and testing? Can we apply the Gnanasambandam and Chan to balance this? How do we quantify the shift?
- Compare min max equation performance against using an hinge loss in all the re weighting algorithms, As we can see that using an hinge loss is equivalent to solving a dual problem in SVM.
- Image generation suffers from bias, some prompts include, people with drug overdose and terrorists, How do we account for such bias? Can we quantify the bias?

10.2 Combined models trained on different data

Can we use these techniques and combine smaller models into a large model using One size fits all paper Gnanasambandam and Chan Suppose we Combine models trained on MNIST and Image Net and achieve better accuracy compared to the model trained on combined MNIST and Image Net data? Does this paper solve the class imbalance problem by equally weighting all the classes? As you are training with the objective of equally loss across all the classes?

References

P. D. Auroux. Proof of legragian multipliers, 2010. URL https://ocw.mit.edu/courses/18-02sc-multivariable-calculus-fall-2010/ebbeb8e61827a8058d2c45b674d003b3_MIT18_02SC_notes_22.pdf.

H. Bahng, S. Chun, S. Yun, J. Choo, and S. J. Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning (ICML)*, 2020.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011.

J. Chai and X. Wang. Self-supervised fair representation learning without demographics. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=7TGpLKADODE.

J. Chai, T. Jang, and X. Wang. Fairness without demographics through knowledge distillation. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=8gjwWnN5pfy.

F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

B. EE364b. Ee364b stanford constarint subgrad, a. URL https://web.stanford.edu/class/ee364b/lectures/constr_subgrad_slides.pdf.

B. EE364b. Ee364b stanford subgrad, b. URL https://web.stanford.edu/class/ee392o/subgrad_method.pdf.

B. EE364b. Game theory min-max problem, c. URL https://web.stanford.edu/class/ee364b/lectures/cvxccv.pdf.

Y. Eldar and T. Dvorkind. A minimum squared-error framework for generalized sampling. *IEEE Transactions on Signal Processing*, 54(6):2155–2167, 2006. doi: 10.1109/TSP.2006.873488.

A. Gnanasambandam and S. H. Chan. One size fits all: Can we train one denoiser for all noise levels? *CoRR*, abs/2005.09627, 2020. URL https://arxiv.org/abs/2005.09627.

P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. Chi. Fairness without demographics through adversarially reweighted learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 728–740. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/07fc15c9d169ee48573edd749d25945d-Paper.pdf.

P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. H. Chi. Fairness without demographics through adversarially reweighted learning, 2020b.

S. Ravfogel, M. Twiton, Y. Goldberg, and R. Cotterell. Linear adversarial concept erasure. *CoRR*, abs/2201.12091, 2022. URL https://arxiv.org/abs/2201.12091.

B. Sadeghi, R. Yu, and V. Boddeti. On the global optima of kernelized adversarial representation learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7970–7978, 2019a. doi: 10.1109/ICCV.2019.00806.

B. Sadeghi, R. Yu, and V. N. Boddeti. Constrained sampling: Optimum reconstruction in subspace with minimax regret constraint. *IEEE Transactions on Signal Processing*, 67(16):4218–4230, aug 2019b. doi: 10.1109/tsp.2019.2925608. URL https://doi.org/10.1109%2Ftsp.2019.2925608.

B. Sadeghi, S. Dehdashtian, and V. Boddeti. On characterizing the trade-off in invariant representation learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=3gfpBR1ncr. Featured Certification.

A. Setlur, D. Dennis, B. Eysenbach, A. Raghunathan, C. Finn, V. Smith, and S. Levine. Bitrate-constrained dro: Beyond worst case robustness to unknown group shifts, 2023.

J. Tian, Z. He, X. Dai, C.-Y. Ma, Y.-C. Liu, and Z. Kira. Trainable projected gradient method for robust fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7836–7845, 2023.

TrefethenBau. Rayleighquotient iteration. URL https://www.cs.cmu.edu/afs/cs/academic/class/15859n-f16/Handouts/TrefethenBau/RayleighQuotient-27.pdf.

T.-D. Truong, N. Le, B. Raj, J. Cothren, and K. Luu. Fredom: Fairness domain adaptation approach to semantic scene understanding. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023.

L. Wang and V. N. Boddeti. Do learned representations respect causal relationships?, 2022.

J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. *CoRR*, abs/2103.03230, 2021. URL https://arxiv.org/abs/2103.03230.

B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating unwanted biases with adversarial learning. *CoRR*, abs/1801.07593, 2018. URL http://arxiv.org/abs/1801.07593.

11 Additional Math

11.1 Matrix Differentiation

Matrix differentiation with eignsum method is intuitive and easy to solve complex differentiation. We can also leverage chain rule, and convert complex matrices into simple simple know derivatives.
Example 1

$$\frac{\partial(||s-\theta PP^TX||^2)}{\partial P} \tag{22}$$

Lets break the above equation into smaller know differentiable terms.

$$\begin{aligned}
K &= PP^T \\
U &= \theta KX \\
\frac{\partial(||s-\theta PP^TX||^2)}{\partial P} &= \frac{\partial(||s-U||^2)}{\partial P} \\
&= \frac{\partial trace((s-U)(s-U)^T)}{\partial P} \\
&= \frac{\partial trace(UU^T-2s^TU)}{\partial P} \\
&= \frac{\partial trace(UU^T)}{\partial P} - 2\frac{\partial trace(s^TU)}{\partial P}
\end{aligned}$$

Lets solve 2 differentiation separately, using eignsum notation.

$$\begin{aligned}
\frac{\partial trace(s^TU)}{\partial P} &= \frac{\partial s_{1i}U_{1i}}{\partial U_{1j}} * \frac{\partial U_{1j}}{\partial K_{kl}} * \frac{\partial K_{kl}}{\partial P_{mn}} \\
\frac{\partial s_{1i}U_{1i}}{\partial U_{1j}} &= s_{1i}\delta_{ij} = s_{1j}
\end{aligned}$$

where,

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

and similarly for the second chain.

$$\begin{aligned}
U &= \theta KX \\
U_{1j} &= \theta_{1p}K_{pq}X_{qj} \\
\frac{\partial U_{1j}}{\partial K_{kl}} &= \frac{\partial \theta_{1p}K_{pq}X_{qj}}{\partial K_{kl}} \\
&= \theta_{1p}X_{qj}\frac{\partial K_{pq}}{\partial K_{kl}} \\
&= \theta_{1p}X_{qj}\delta_{kp}\delta_{lq} \\
&= \theta_{1k}X_{lj}
\end{aligned}$$

and similarly for the third chain.

$$\begin{aligned}
K &= PP^T \\
K_{kl} &= P_{kz}Plz \\
\frac{\partial K_{kl}}{\partial P_{mn}} &= \frac{\partial P_{kz}Plz}{\partial P_{mn}} \\
&= P_{kz}\frac{\partial P_{lz}}{\partial P_{mn}} + P_{lz}\frac{\partial P_{kz}}{\partial P_{mn}} \\
&= P_{kn}\delta_{lm} + P_{ln}\delta_{km}
\end{aligned}$$

Combining all three

$$\begin{aligned}
\frac{\partial trace(s^T U)}{\partial P_{mn}} &= s_{1j}\theta_{1k}X_{lj}[P_{kn}\delta_{lm} + P_{ln}\delta_{km}] \\
&= s_{1j}\theta_{1k}X_{lj}P_{kn}\delta_{lm} + s_{1j}\theta_{1k}X_{lj}P_{ln}\delta_{km} \\
&= s_{1j}\theta_{1k}X_{mj}P_{kn} + s_{1j}\theta_{1m}X_{lj}P_{ln} \\
&= X_{mj}s_{j1}^T\theta_{1k}P_{kn} + \theta_{m1}^T s_{1j}X_{jl}^T P_{ln} \\
\frac{\partial trace(s^T U)}{\partial P} &= Xs^T\theta P + \theta^T sX^T P \\
\frac{\partial trace(s^T U)}{\partial P} &= [(Xs^T\theta) + (Xs^T\theta)^T]P
\end{aligned}$$

Moving onto the Second term

$$\begin{aligned}
\frac{\partial trace(UU^T)}{\partial P} &= \frac{\partial U_{1i}U_{1i}}{\partial U_{1j}} * \frac{\partial U_{1j}}{\partial K_{kl}} * \frac{\partial K_{kl}}{\partial P_{mn}} \\
&= 2U_{1i}\delta_{ij} * \theta_{1k}X_{lj}[P_{kn}\delta_{lm} + P_{ln}\delta_{km}] \\
&= 2U_{1j} * \theta_{1k}X_{lj}P_{kn}\delta_{lm} + 2U_{1j} * \theta_{1k}X_{lj}P_{ln}\delta_{km} \\
&= 2U_{1j} * \theta_{1k}X_{mj}P_{kn} + 2U_{1j} * \theta_{1m}X_{lj}P_{ln} \\
&= 2 * X_{mj}U_{j1}^T\theta_{1k}P_{kn} + 2\theta_{m1}^T U_{1j} * X_{jl}^T P_{ln} \\
&= 2 * XU^T\theta P + 2\theta^T UX^T P \\
&= 2 * [X(\theta PP^T X)^T\theta + (X(\theta PP^T X)^T\theta)^T]P
\end{aligned}$$

Solving the matrix equation for max P

$$\begin{aligned}
2 * [X(\theta PP^T X)^T\theta + (X(\theta PP^T X)^T\theta)^T]P &= 2 * [(Xs^T\theta) + (Xs^T\theta)^T]P \\
[X(\theta PP^T X)^T\theta + (X(\theta PP^T X)^T\theta)^T]P &= [(Xs^T\theta) + (Xs^T\theta)^T]P
\end{aligned}$$

Solving for θ

$$\begin{aligned}
\frac{\partial(||s - \theta PP^T X||^2)}{\partial\theta} &= \frac{\partial(||s - U||^2)}{\partial\theta} \\
&= \frac{\partial trace(UU^T)}{\partial\theta} - 2\frac{\partial trace(s^T U)}{\partial\theta} \\
\frac{\partial trace(s^T U)}{\partial\theta} &= s_{1j}\frac{\partial U_{1j}}{\partial\theta_{1k}} \\
U_{1j} &= \theta_{1p}K_{pq}X_{qj} \\
\frac{\partial U_{1j}}{\partial\theta_{1k}} &= K_{kq}X_{qj} \\
\frac{\partial trace(s^T U)}{\partial\theta} &= s_{1j}K_{kq}X_{qj} \\
&= s_{1j}X_{jq}^T K_{qk}^T \\
&= sX^T K^T \\
\frac{\partial trace(UU^T)}{\partial\theta} &= \frac{\partial U_{1i}U_{1i}}{\partial U_{1j}} * \frac{\partial U_{1j}}{\partial\theta_{1k}} \\
&= 2U_{1j}K_{kq}X_{qj} \\
&= 2U_{1j}X_{jq}^T K_{qk}^T \\
&= 2UX^T K^T \\
\frac{\partial(||s - \theta PP^T X||^2)}{\partial\theta} &= 2UX^T K^T - 2sX^T K^T
\end{aligned}$$

Equating to 0 to find the optimum

$$UX^TK^T = sX^TK^T \implies \theta KXX^TK^T = sX^TK^T \implies \theta = s(KX)^\dagger$$

Substituting θ in the optimum equation to find saddle point P_*, θ_*

$$X^TK\theta^T = s^T \implies X^TK\theta^T = s^T$$

11.2 Convexity of Linear concept Erasure

Assume,

$$P_1^TP_1 = I$$
$$P_2^TP_2 = I$$

Disproving if $\lambda P_1 + (1-\lambda)P_2$

$$\begin{aligned}(\lambda P_1 + (1-\lambda)P_2))^T(\lambda P_1 + (1-\lambda)P_2)) &= \lambda^2 I + (1-\lambda)^2 I + ((1-\lambda))\lambda P_1^TP_2 + (\lambda)(1-\lambda)P_2^TP_1 \\ &= \lambda^2 I + (1-\lambda)^2 I + 2(1-\lambda)\lambda\frac{[P_1^TP_2 + P_2^TP_1]}{2} \\ &\neq I \quad \forall P_1, P_2\end{aligned}$$

11.3 Math Raleigh Coefficient Proof

Appendix D, Theorem 3 proof in TMLR Sadeghi et al..

$$\max_{X; X^TCX=I} trace(X^TBX)$$

where B,C are symmetric matrices, we use Lagrangian similar to the section 2

$$\begin{aligned}g(X,\lambda) &= trace(X^TBX) - <\lambda, X^TCX - I> \\ \frac{\partial g(X,\lambda)}{X_{pq}} &= \partial[X_{ji}B_{jk}X_{ki} - \lambda_{il}[X_{ji}C_{jk}X_{kl} - \delta_{il}]] \\ &= X_{ji}B_{jk}\delta_{kp}\delta_{iq} + X_{ki}B_{jk}\delta_{jp}\delta_{iq} - \lambda_{il}X_{ji}C_{jk}\delta_{kp}\delta_{lq} - \lambda_{il}C_{jk}X_{kl}\delta_{jp}\delta_{iq} \\ &= X_{jq}B_{jp} + X_{kq}B_{pk} - \lambda_{iq}X_{ji}C_{jp} - \lambda_{ql}C_{pk}X_{kl} \\ &= (B+B^T)X - C^TX\lambda - CX\lambda^T \\ &= 2BX - 2CX\lambda \\ 0 &= 2BX - 2CX\lambda \\ BX &= CX\lambda\end{aligned}$$

This is equivalent to solving generalised eigen value equation. and solving the dual equation of $\max g(\lambda)$

$$\begin{aligned}\max g(X,\lambda) &= trace(X^TBX) - <\lambda, X^TCX - I> \\ &= trace(X^TCX\lambda) - 0 \\ &= trace(I\lambda) \\ &= \max_\lambda \Sigma_\lambda \lambda_{ii}\end{aligned}$$

The last line implies all the positive eigen values of generalised eigen value equation. So solving $Bx_i = \lambda_{ii}Cx_i$

12 Optimisation Preliminaries

Most of the preliminaries of the optimisation are taken from the Boyd and Vandenberghe. Important optimisation techniques are Constrained Optimisation, sub-gradient methods, and ADMM (Boyd et al.). These constraint optimisation is used by Tian et al. to constraint the fine tuned weights around the sphere of pre-training network weights. First we need to understand why does Lagrangian multipliers work. The sketch of proof will be from the geometric perspective is given in Auroux. One way we can also think as If there is a gradient in the plane of constraint, We can still achieve the minima, so the gradient in the plane of constraint is 0. More on the second statement.

12.1 Constraint Optimisation

Preliminary equation in constrained optimisation is given by the following equation

$$\begin{aligned} &\min_{\vec{x}} && f(\vec{x}) \\ &\textit{subject to } A\vec{x} = b \end{aligned} \tag{23}$$

where f is a convex and differentiable function, $f : R^n \to R$. Using method of Lagrangian multipliers we can modify the above equation assuming strong duality.

$$\max_{y} \inf_{x} f(x, y)$$

We can write $g(y) = \inf_x f(x, y)$. This can be solved by using gradient ascent.

Algorithm 2 Constrained Optimisation

Ensure: $Ax = b$
  $x \leftarrow RAND_{1\times N}$
  $y \leftarrow RAND_{1\times N}$
  $\alpha_k \leftarrow lr_0$
  for *epoch* $\leftarrow 1$ to N **do**
    $x_{k+1} = \inf_x g(x, y_k)$
    $y_{k+1} = y_k + \alpha_k(Ax_{k+1} - b)$
    $\alpha_{k+1} = \frac{\alpha_k}{\sqrt{k}}$ ▷ We can have different variations, constant, linear decay, step decay.
  end for

To improve the speed of the algorithm we can use alternating optimisation rather than calculating inf every iteration. cite the paper that proved alternating optimisation can yield similar results as to the calculating inf. However, Tian et al. has used alternating optimisation, alternating every $\mathbf{f_a}$ epochs. Therefore we can modify the above algorithm as

  for *epoch* $\leftarrow 1$ to N **do**
    $x_{k+1} = x_k - \gamma_k \nabla_x g(x, y_k)$
    if *epoch* $\%\mathbf{f_a} = 0$ **then**
      $y_{k+1} = y_k + \alpha_k(Ax_{k+1} - b)$
    end if
  end for

We can further relax the constraint of f being differentiable and using sub-gradients, we define sub-gradient as any function h such that

$$g(y) \geq g(x) + h_x^T(y - x) \tag{24}$$

we can observe that when f is differentiable we can replace $h_x = \nabla_x f(x_k)$, the proof can be obtained from the Taylor expansion. $f(x + \delta x) = f(x) + \delta x f'(x) + \frac{(\delta x)^2}{2!} f''(x) + \cdots$. If we use the sub gradient methods it is not guaranteed to converge to the global optima, but we can find x^* for $f_{best} - f_{optim} \leq \epsilon$. The proofs for convergence is completed in the EE364b.

Solving equation 23, We can formulate the problem as

$$\begin{aligned} &\min_{\vec{x}} && f(\vec{x}) \\ &\textit{subject to } \vec{x} \in \mathcal{C} \end{aligned}$$

$$\begin{aligned} &\min_{\vec{x}} && f(\vec{x}) + y^T(Ax - b) + \frac{\rho}{2}||Ax - b||^2 \\ &\textit{subject to } \vec{x} \in \mathcal{C} \end{aligned}$$

Where $\mathcal{C}$, is the subspace denoted by $\mathcal{C} = \{x \in R^n | Ax = b\}$ We can modify the above equation and call it Augmented Lagrangians by adding $\frac{\rho}{2}||Ax - b||^2$ to the minimisation equation as the

objective does not change, only leads to a faster convergence cite the reference. We can also combine $y^T(Ax-b)+\frac{\rho}{2}||Ax-b||^2$ into

$$r = Ax - b, \frac{y}{\rho} = u$$
$$y^T r + \frac{\rho}{2} r^T r = \frac{\rho}{2}(||r + \frac{y}{\rho}||^2 - ||\frac{y}{\rho}||^2)$$
$$= \frac{\rho}{2}(||r-u||^2 + ||u||^2)$$

Another form of writing the above equation with disentangled equations and constraints and bounded by the equality $\vec{x} = \vec{z}$.

$$\min_{\vec{x}=\vec{z}} f(\vec{x}) + g_{\mathcal{C}}(\vec{z})$$
$$\min_{\vec{x}=\vec{z}} f(\vec{x}) + g_{\mathcal{C}}(\vec{z}) + \frac{\rho}{2}||x - z + u||^2$$

Algorithm 3 ADMM

Ensure: $Ax = b$
  $x \leftarrow RAND_{1\times N}$
  $y \leftarrow RAND_{1\times N}$
  $\alpha_k \leftarrow lr_0$
  for *epoch* $\leftarrow 1$ to N **do**
    $x_{k+1} = \inf_x [f(x) + \frac{\rho}{2}||x - z + u||^2]$
    $z_{k+1} = \inf_z [g_{\mathcal{C}}(\vec{z}) + \frac{\rho}{2}||x - z + u||^2] \implies \prod_{\mathcal{C}}(x_{k+1} + u_k) \triangleright g_{\mathcal{C}}(\vec{z})$ *sub gradient* $= 0 z \in \mathcal{C}$
    $u_{k+1} = u_k + x_{k+1} - z_{k+1}$
    $\alpha_{k+1} = \frac{\alpha_k}{\sqrt{k}}$ ▷ We can have different variations, constant, linear decay, step decay.
  end for

12.2 Applications

In the Application section we target are simple linear models, and try to incorporate above mentioned sub gradient and the ADMM methods and come with convergent algorithms.

12.2.1 Regression with absolute deviations

Objective is to minimise the L1 norm for the regression problem. We can write this problem as

$$\min_{\mathbf{z}} \quad ||\mathbf{z}||_1$$
$$\textit{subject to } \mathbf{Ax} - \mathbf{z} = \mathbf{b}$$

Combining all the above properties we can perform alternating gradient decent as

$$x_{k+1} = \inf_x F(x) = y^T(Ax - z_k - b) + \frac{\rho}{2}||Ax - z_k - b||_2^2$$
$$\implies \inf_x [\frac{\rho}{2}||Ax - z_k - b + \frac{u_k}{\rho}||^2]$$
$$\implies A^T Ax = z_k + A^T b - \frac{u_k}{\rho} + \mathcal{N}(A)$$
$$= (A^T A)^{-1}(z_k + A^T b - \frac{u_k}{\rho} + \mathcal{N}(A))$$
$$= (A^T A)^{-1}(z_k + A^T b - \frac{u_k}{\rho})$$

The derivative of $||\mathbf{z}||_1$ does not exist at z = 0, we can use the sub gradient method to account for the gradient. This g satisfies the properties of sub gradients 24. This means there is no closed form that exists but gradient decent can be applied.

$$g(z) = \begin{cases} -1 & z < 0 \\ 1 \textit{ or } -1 & z = 0 \\ 1 & z > 0 \end{cases}$$

$$
\begin{aligned}
z_{k+1} = \inf_x F(z) &= ||z||_1 + y^T(Ax_{k+1} - z - b) + \frac{\rho}{2}||Ax_{k+1} - z - b||_2^2 \\
&\implies \inf_x [||z||_1 + \frac{\rho}{2}||Ax_{k+1} - z - b + u_k||^2] \\
&\implies \Delta z'_{k+1} = (b + z_k - Ax_{k+1} - \frac{u_k}{\rho}) + g(z_k) \\
&= z_k - \alpha\rho(b + z_k - Ax_{k+1} - \frac{u_k}{\rho}) - g(z_k)
\end{aligned}
$$

The latent variable u can be updated according to 3

$$
u_{k+1} = u_k + Ax_{k+1} - z_{k+1} - b
$$

---

for *epoch* $\leftarrow 1$ to N **do**
  $x_{k+1} = (A^TA)^{-1}(z_k + A^Tb - \frac{u_k}{\rho})$
  $z_{k+1} = z_k$
  for *zconvergence* $\leftarrow 1$ to K **do**
    $z_{k+1} = z_k - \alpha\rho(b + z_k - Ax_{k+1} - \frac{u_k}{\rho}) - \alpha g(z_k)$
  end for ▷ closed form exists as $S_{\frac{1}{\rho}(*)}$, but its complicated to derive
  $u_{k+1} = u_k + Ax_{k+1} - z_{k+1} - b$ ▷ gradient ascent
end for

12.2.2 Lasso Regression

We also want to add a constraint of restricting the weights to a fixed radii hyper sphere. **Weight constraint for finetuning** is a generalised form of Lasso regression where $\mathbf{x_0} = 0$. We will derive the weight restrictions to the Linear model and extend it to the complex deep learning ResNet. These equations are tried out in Tian et al..

$$
\begin{aligned}
&\min_{\mathbf{z}} \quad ||\mathbf{z}||_2^2 \\
&\textit{subject to } \mathbf{Ax} - \mathbf{z} = \mathbf{b} \\
&\quad ||\mathbf{x} - \mathbf{x_0}||_2 \leq \epsilon
\end{aligned}
$$

$$
\begin{aligned}
&\min_{\mathbf{z}} \quad ||\mathbf{z}||_2^2 \\
&\textit{subject to } \mathbf{Ax} - \mathbf{z} = \mathbf{b} \\
&\quad ||\mathbf{y} - \mathbf{x_0}||_2 \leq \epsilon \\
&\quad \mathbf{x} = \mathbf{y}
\end{aligned}
$$

$$
\max_{\mu_1, \mu_2, \mu_3} \min_{x,y,z} ||\mathbf{z}||_2^2 + \mu_1^T(\mathbf{Ax} - \mathbf{z} - \mathbf{b}) + \mu_2^T(||\mathbf{y} - \mathbf{x_0}||_2 - \epsilon) + \mu_3^T(\mathbf{x} - \mathbf{y})
$$
$$
\mu_2 \geq 0
$$

---

for *epoch* $\leftarrow 1$ to N **do**
  $x_{k+1} = (A^TA)^{-1}A^T(z_k + b - u_k)$
  $z_{k+1} = \rho(Ax_{k+1} - b + u_k)$
  $u_{k+1} = u_k + Ax_{k+1} - z_{k+1} - b$ ▷ gradient ascent
end for
