CSE 528 Project

Daniel Billmann

December 18, 2022

1 Proposal

I will attempt a gradient-descent based approach to solving a 3D line symmetry problem. The idea is to find a plane $\in \mathbb{R}^2$ that can be embedded in a manifold $F \in \mathbb{R}^3$ such that the plane lies along a symmetry line for the manifold. We denote this plane as Ω_{sym}The null hypothesis h_0 states that Ω_{sym}will be at the bilateral facial line, or $\frac{\Delta}{2}$ where $\Delta = x_{max} - x_{min}$. For the purposes of our application, we constrain our dataset to only include images of human faces with the straight-on or eye-level angle. We will use the central points from Mediapipe's 468 landmark facial mesh predictions as our h_0 for Ω_{sym}. Unlike most other problems in machine learning where we learn general parameters to fit to a population distribution, we are training and testing weights on a single specified image at a time. Our approach will be general and our solutions will be customized. We define L and R to be a sub-manifolds such that

$$\begin{gathered} L, R \in F, \\ L = \{\forall p \in F, p < \Omega_{sym}\} \\ R = \{\forall p \in F, p \geq \Omega_{sym}\} \end{gathered} \tag{1}$$

The reason we define two submanifolds is because we will use the points and locations of one to predict the locations and points in the other. Any differences will be considered error and calculable with the loss function. For our experiments, L will be the training set and R will be the test set.

In these terms, perfect bilateral facial symmetry, or h_0, is merely a reflection across the x axis plus a set of transpose coordinates, t, which indicate the starting position of h_0. Using our terms, this solution would be

$$\begin{gathered} h_0 = \begin{bmatrix} -1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \end{bmatrix} \\ R = h_0^T \begin{bmatrix} L \\ 1 \end{bmatrix} \end{gathered} \tag{2}$$

We will consider $\Omega_{sym}$to be equivalent to the learnable parameter Θ. It is well documented that Θ can be directly computed with a closed-form (CF) solution:

$$\theta = (X^T X)^{-1} X^T Y \tag{3}$$

which, substituting in symbols related to our problem would be

$$\Omega_{sym} = (\mathrm{L}^T L)^{-1} L^T R \tag{4}$$

2 Approach

The first step is to establish corresponding points between the L and R sub-manifolds. Once that is established, the rest of the approach should fall into place.

Given the computational difficulty of computing the CF solution, we also will consider an iterative approach computing the loss with gradient descent (GD).

Here, we define the loss function, $J(R, L,\ \Omega_{sym})$. We use the sum of squared errors (SSE) function to calculate error.

$$\begin{aligned} & L, R, \Omega_{sym} \in \mathbb{R}^3 \\ J(L, R,\ \Omega_{sym}) = \quad & \textstyle\sum_i^{|L|} ({\omega_{sym}}^T l_i - r_i)^2 \end{aligned} \tag{5}$$

We then adjust Ω_{sym} using the GD optimization method. This will find and calculate the gradient of Ω_{sym} in the x, y, and z directions. This will move the plane's vertices in accordance with the image we are learning.

$$\Omega_{sym} := \Omega_{sym} \text{-}\ \alpha \tfrac{dJ(L,R,\Omega)}{dL} \tag{6}$$

Note to the reader: I couldn't cleanly put the Ω_{sym} symbol in the derivative part of the equation, so I substituted it with Ω

3 Results and Analysis

Our original hypothesis h_0 was that facial symmetry would be a simple computation flipping the L values across the Y axis: $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. This experiment demonstrated that this hypothesis is not an accurate portrayal of symmetry.

We did show that the GD and CF solutions return similar matrices

3.1 Data set-up

We used the diagram of mediapipe's canonical face with 468 landmarks in two dimensions to identify corresponding points across the y-axis. The 28 central points, denoted as **IDX_SET** were identified manually using the diagram. The other 440 were calculated using a variety of algorithms. The first observation was that landmarks on the left side of the face had values less than 268. Values on the right side of the face had values greater than 268. We horizontally matched the points across the **IDX_SET** iteratively searching for the minimum distance between the y value for the left point and each landmark on the right. For 217 of 220 pairs ($\frac{440}{2}$), this approach was successful. For 4 landmarks, we needed to reassign the values manually. For the four anomalous points the difference between the predicted landmark on the right and the actual landmark on the right was at most 0.0001. Given the size of the issue, we assumed this discrepancy was due to image resolution constraints and prioritized solving other problems.

Once we had the corresponding landmark indices, we wrote a function to build the L and R datasets and imported the function in the main module.

3.2 Code

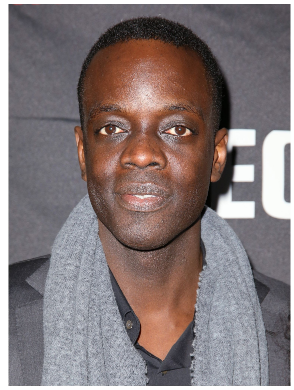
Our CF approach was exactly as we described in our above formulation. Given the size of the parameter space, computation time was nearly instantaneous, even in Python. We made one change to the gradient descent optimization method in that we added a momentum hyperparameter η. This hyperparameter allowed our function to converge to the same error minimum in 25% to 90% of the time that simple gradient descent allowed. The updated gradient descent algorithm is as follows, designating Δ_n as the change with momentum at epoch n and η as the momentum hyperparameter:

$$\begin{gathered}\Delta_n = \alpha \frac{dJ(L, R, \Omega)}{dL} + \eta * \Delta_{n-1} \\ \Omega_{symm_{n+1}} := \ \Omega_{symm_n} - \Delta_n\end{gathered} \tag{7}$$

Total compute time using GD without momentum was fewer than two seconds at worst. The difference between using GD with momentum and GD without momentum for each of our images was less than 0.001. If in the future we choose to prioritize convergence speed over accuracy, this may be a solution, acknowledging that we may get caught in local minima.

Below are some examples of our inputs based on the image constraints we set for ourselves and the h_0 we predicted.

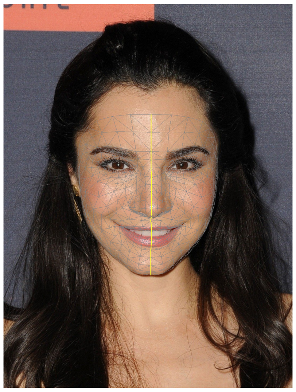
Martha Higadera - Original **Ato Essandoh - Original**

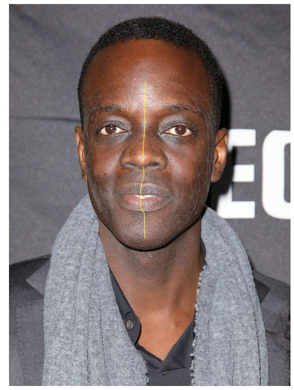


Null Hypothesis matrices

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Martha Higadera - Annotated **Ato Essandoh - Annotated**





CF parameter matrices

$$\begin{bmatrix} 1.49246785 & 0.00811304 & 0.03728997 \\ -0.00936537 & 0.9940122 & -0.01611439 \\ 1.18919534 & 0.00872882 & 1.05794324 \end{bmatrix} \qquad \begin{bmatrix} 1.42325736 & -0.00622343 & -0.02159802 \\ 0.05631605 & 1.01482869 & -0.0089678 \\ 1.9773857 & 0.03074522 & 0.86091489 \end{bmatrix} \tag{9}$$

Gradient Descent parameter matrices

$$\begin{bmatrix} 1.38084999 & 0.00745683 & 0.03399044 \\ 0.07959525 & 0.99453521 & -0.01348463 \\ 1.07278983 & 0.00804446 & 1.05450219 \end{bmatrix} \qquad \begin{bmatrix} 1.404241 & -0.00627849 & -0.02090888 \\ 0.07413646 & 1.01488031 & -0.00961366 \\ 1.95673204 & 0.03068358 & 0.86166838 \end{bmatrix} \tag{10}$$

These results led me to ask how similar are the weight matrices for the two different approaches. Below I show the elementwise absolute difference between the matrices in equation 9 and equation 10.

Absolute Difference between GD and CF

$$\begin{bmatrix} 0.11161786 & 0.00065621 & 0.00329953 \\ 0.08896062 & 0.00052301 & 0.00262976 \\ 0.11640551 & 0.00068436 & 0.00344106 \end{bmatrix} \qquad \begin{bmatrix} 0.0190163614 & 0.000055055 & 0.689141568 \\ 0.01782041 & 0.0000516141 & 0.0006458574 \\ 0.020653659 & 0.00006164 & 0.000753483 \end{bmatrix} \tag{11}$$

Sum of differences: **0.32821790744927**

Sum of differences: **0.05974723192668**

This shows there are small differences between the approaches on a per-image basis.

4 Conclusion

Our null hypothesis h_0 was proved incorrect. The weight matrices, for both CF and GD approaches, were more similar to the identity matrix $I \in \mathbb{R}^3$ than the eventual weights. In the next section, we speculate as to why this might be.

5 Next steps

There are several key questions raised by this project that are worth exploring in another project or a research :

- Was the data prepared correctly? Theoretically, the origin point should be the center value of the face. Looking at empirical image data, this value is $0.5 + \epsilon$.
- If the data was prepared correctly, does the symmetry metric properly reflect the current paradigm?
- In what situation does this solution generalize to other types of linear symmetry? Are there instances where it would be ineffective or inappropriate?
- Even though error rates are similar, there are discrepancies between GD $\Omega_{sym}$and CF Ω_{sym}matrices. How does the approach affect the final result?
- **Most importantly** How is this useful or applicable?