ECE / CS 6501: Geometry of Data **Due Thu 12/8**

Homework 4: Information Geometry

Instructions: Submit a single Jupyter notebook (.ipynb) of your work to Collab by 11:59pm on the due date. All code should be written in Python. **Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question.**

You may discuss the concepts with your classmates, but write up the answers entirely on your own. Do not look at another student's answers, do not use answers from the internet or other sources, and do not show your answers to anyone.

This is due on the final exam day, and as such, there will be no late days available and no extensions!

Pick **one** of the following two questions to answer!

1. Recall from class that the Fisher information metric for a Gaussian pdf with mean and standard deviation parameters, (μ, σ), is given by

$$g = \begin{pmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{2}{\sigma^2} \end{pmatrix}.$$

   (a) Compute the Christofell symbols for this metric.

   (b) Using your Christofell symbols, confirm that curves of the following form satisfy the geodesic equation:

$$(\mu(t), \sigma(t)) = (a,\ be^{ct}),$$

   where a, b, c are constants.

   (c) Next, confirm that curves of the following form also satisfy the geodesic equation:

$$(\mu(t), \sigma(t)) = \left(a\tanh(ct) + b,\ a\frac{\sqrt{2}}{2}\text{sech}\,(ct) \right),$$

   where a, b, c are constants.

   (d) Consider two Gaussians with (μ, σ) parameters: $(0, 1)$ and $(0, 2)$. Plot their pdf's in the same graph. What is the geodesic curve between them? Plot this curve in the Poincaré half plane. What is the geodesic distance between them?

   Answer the same questions between $(0, 4)$ and $(0, 8)$. Can you conjecture a general rule about geodesic distances under scaling both standard deviations by the same factor? **Hint:** You may use $t \in [0, 1]$ in one of the equations above.

   (e) Consider two Gaussians with (μ, σ) parameters: $(-1, 1)$ and $(1, 1)$. Plot their pdf's in the same graph. What is the geodesic curve between them? Plot this curve in the Poincaré half plane. What is the geodesic distance between them?

   Answer the same questions between $(-2, 2)$ and $(2, 2)$. Can you conjecture another general rule about geodesic distances? **Hint:** You will want to use $t \in [-1, 1]$ in one of the equations above.

(f) For the previous example between the Gaussians $(-1, 1)$ and $(1, 1)$, what is the Fréchet mean of these two? That is, what is the midpoint (μ, σ) on the geodesic between them? Plot the pdf's for these two Gaussians and their Fréchet mean on the same graph. What observations do you have about the mean?

2. Write two functions to compute the maximum-likelihood estimate (MLE) of a logistic regression from training data. In the first function, use standard gradient descent with a fixed step size. In the second function, use the natural gradient with an empirical estimate for the Fisher information matrix.

   You should implement the gradient descent of logistic regression yourself, rather than just use a library, to make sure that you know what you are doing. However, feel free to look up the equations for the log-likelihood (cross-entropy) gradient on the internet.

   (a) Pick a data set (suggestions: MNIST, Fashion MNIST, CIFAR-10), and pick two classes from that data set to test your binary classifier. You may use a suitable subsample of the training set to save time.

   (b) Run your fixed-step gradient descent on the training data. Use a random initialization for the weight vector, and experiment to find a step size that works reasonably well. Run for a fixed number of iterations (at least 10). Save the training loss for each iteration.

   (c) Using the same random initialization for the weight vector that you used above, run your natural gradient descent method for the same number of iterations. Again, save the training loss at each iteration.

   (d) Repeat the previous two steps several times (at least 5 times). Each repeat should use a new random weight initialization, but always equal for the two algorithms. Plot the average loss as a function of the iteration for both algorithms. Which algorithm converges faster?