Martinez Assignment 3
Topic 3: Statistical Learning Methods

Part 1

Given:

$$P(\mathbf{x}|y_q) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_q)^T\,\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_q)\right); q = 1, 2$$

Prove that linear discriminant functions

$$g_q(\mathbf{x}) = \boldsymbol{\mu}_q^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \tfrac{1}{2}\boldsymbol{\mu}_q^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_q + \ln P(y_q); q = 1, 2$$

And decision boundary $g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x}) = 0$ is given by

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$$

$$\mathbf{w}^T\mathbf{x} + w_0 = (\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_2^T)\boldsymbol{\Sigma}^{-1}\mathbf{x} - \tfrac{1}{2}(\boldsymbol{\mu}_1^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{\mu}_2) + \ln\frac{P(y_1)}{P(y_2)}$$

There are a few cases in which we would handle the Gaussian Discriminant Function. However, this case is for when $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ is our covariance matrix. This is a diagonal matrix that shows the variables are statistically independent.

$$p(\mathbf{x}|y_i) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma_i}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i}^T\boldsymbol{\Sigma_i}^{-1}(\mathbf{x}-\boldsymbol{\mu_i})\right] \qquad \text{(Equation 1)}$$

We also know that we can take the log of $g_i(\mathbf{x})$ and start from there:

$$\mathrm{g}_i(\mathbf{x}) = \ln p(\mathbf{x}|y_i) + \ln P(y_i)$$

$$\mathrm{g}_i(\mathbf{x}) = -\tfrac{1}{2}[(\mathbf{x}-\boldsymbol{\mu_i}^T\boldsymbol{\Sigma_i}^{-1}(\mathbf{x}-\boldsymbol{\mu_i})] - \tfrac{n}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\boldsymbol{\Sigma_i}| + \ln P(y_i) \qquad \text{(Equation 2)}$$

We can now drop the second and third terms. The second is a constant and the third will be the same for all classes (since the covariance matrix is just the diagonal).

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})\boldsymbol{\Sigma_i}^{-1}(\mathbf{x}-\boldsymbol{\mu_i}) + \ln P(y_i) \qquad \text{(Equation 3)}$$

Some rearranging of terms will lead us to the following:

$$-\frac{1}{2\sigma^2}[\mathbf{x}^T\mathbf{x} - 2\boldsymbol{\mu_i}^T\mathbf{x} + \boldsymbol{\mu_i}^T\boldsymbol{\mu_i} + \ln P(y_i)] \quad \text{(Equation 4)}$$

$$= -\frac{1}{2\sigma^2}[-2\boldsymbol{\mu_i}^T\mathbf{x} + \boldsymbol{\mu_i}^T\boldsymbol{\mu_i}] + \ln P(y_i) \quad \text{(Equation 5)}$$

With this we can now take the formula

$$\mathbf{w}^T\mathbf{x} + w_i0 \quad \text{(Equation 6)}$$

which is a linear equation.

$$\mathbf{w}_i = \tfrac{1}{2\sigma^2}\boldsymbol{\mu_i}$$

$$w_i0 = \tfrac{1}{2\sigma^2}\boldsymbol{\mu_i}^T\boldsymbol{\mu_i} + \ln P(y_i) \quad \text{(Equation 7)}$$

Now we can plug in and show the difference:

$$\mathrm{g}_i(\mathbf{x}) - g_j(\mathbf{x}) = 0$$

$$\mathrm{g}_i(\mathbf{x}) = w_i^T\mathbf{x} + w_i0$$

$$\mathrm{g}_j(\mathbf{x}) = w_j^T\mathbf{x} + w_j0 \quad \text{(Equation 8)}$$

$$(\mathrm{w}_i - w_j)^T\mathbf{x} + w_i0 - w_j = 0$$

$$= \frac{1}{\sigma^2}(\boldsymbol{\mu_i} - \boldsymbol{\mu_j})^T\mathbf{x} - \frac{\boldsymbol{\mu_i}^T\boldsymbol{\mu_j}}{2\sigma^2} + \ln P(w_i) + \frac{\boldsymbol{\mu_j}^T\boldsymbol{\mu_j}}{2\sigma^2} + \ln P(w_j) = 0$$

$$= (\boldsymbol{\mu_i} - \boldsymbol{\mu_j})^T\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}^T\boldsymbol{\mu_i}^T) - (\boldsymbol{\mu_j}^T - \boldsymbol{\mu_j}) + \sigma^2\frac{\ln P(w_i}{P(w_j)} = 0$$

$$= (\boldsymbol{\mu_i} - \boldsymbol{\mu_j})^T[\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu_i} + \boldsymbol{\mu_j}) - \frac{\sigma^2}{\|\boldsymbol{\mu_i} - \boldsymbol{\mu_j}\|^2}\ln\frac{P(w_i)}{P(w_j)}] = 0$$

$$= w^T(x - x_0) = 0$$
$$w = \boldsymbol{\mu_i} - \boldsymbol{\mu_j}$$

$$x_0 = \frac{1}{2}(\boldsymbol{\mu_i} + \boldsymbol{\mu_j} - \frac{\sigma^2}{\|\boldsymbol{\mu_i} - \boldsymbol{\mu_j}\|^2}\ln\frac{P(w_i)}{P(w_j)}(\boldsymbol{\mu_i} - \boldsymbol{\mu_j})$$

The final equation is actually orthogonal. This function will measure Euclidean distance and maximize that between the to mean vectors. The decision boundary will be orthogonal to the center of that distance. This will be the decision boundary.

Part 2

Perform two iterations of the gradient algorithm to find the minima of $E(w)=2w_1^2+2w_1w_2+5w_2^2$

The starting point is $w=\begin{bmatrix}2 & -2\end{bmatrix}^T$

Draw the contours and show your learning path graphically.

So we begin with the original function and starting vector:

$$\begin{aligned}\mathrm{E}(\mathbf{w}) &= 2\ \mathrm{w}_1^2 + 2w_1w_2 + 5w_2^2 \\ \mathbf{w} &= \begin{bmatrix}2 & -2\end{bmatrix}^T\end{aligned} \qquad \text{(Equation 9)}$$

The gradient is just a set of partial derivatives of the vector. So we can define our gradient as:

$$\nabla E(\mathbf{w}) = \begin{bmatrix}\frac{\partial E(\mathbf{w})}{\partial w_1} & \frac{\partial E(\mathbf{w})}{\partial w_2}\end{bmatrix} \qquad \text{(Equation 10)}$$

With this, we can plug in the functions above and differentiate with respect to :

$$\frac{\partial E(\mathbf{w})}{\partial w_1} = 4w_1 + 2w_2 = 4(2) + 2(-2) = 4 \qquad \text{(Equation 11)}$$

and :

$$\frac{\partial E(\mathbf{w})}{\partial w_2} = 2w_1 + 10w_2 = 2(2) + 10(-2) = -16 \qquad \text{(Equation 12)}$$

And our gradient is then defined as:

$$\nabla E(\mathbf{w}) = \begin{bmatrix}4 & -16\end{bmatrix}^T \qquad \text{(Equation 13)}$$

Gradient descent is an assignment function that updates the value based on the gradient and learning rate. The default function:

$$w_{k+1} = w_k - \alpha \nabla E(\mathbf{w}) \qquad \text{(Equation 14)}$$

where alpha is the learning rate and the gradient is defined above. For now we will set it as 0.1.

Now we will do two iterations of the function and gather the coordinates of two points in the direction of the gradient:

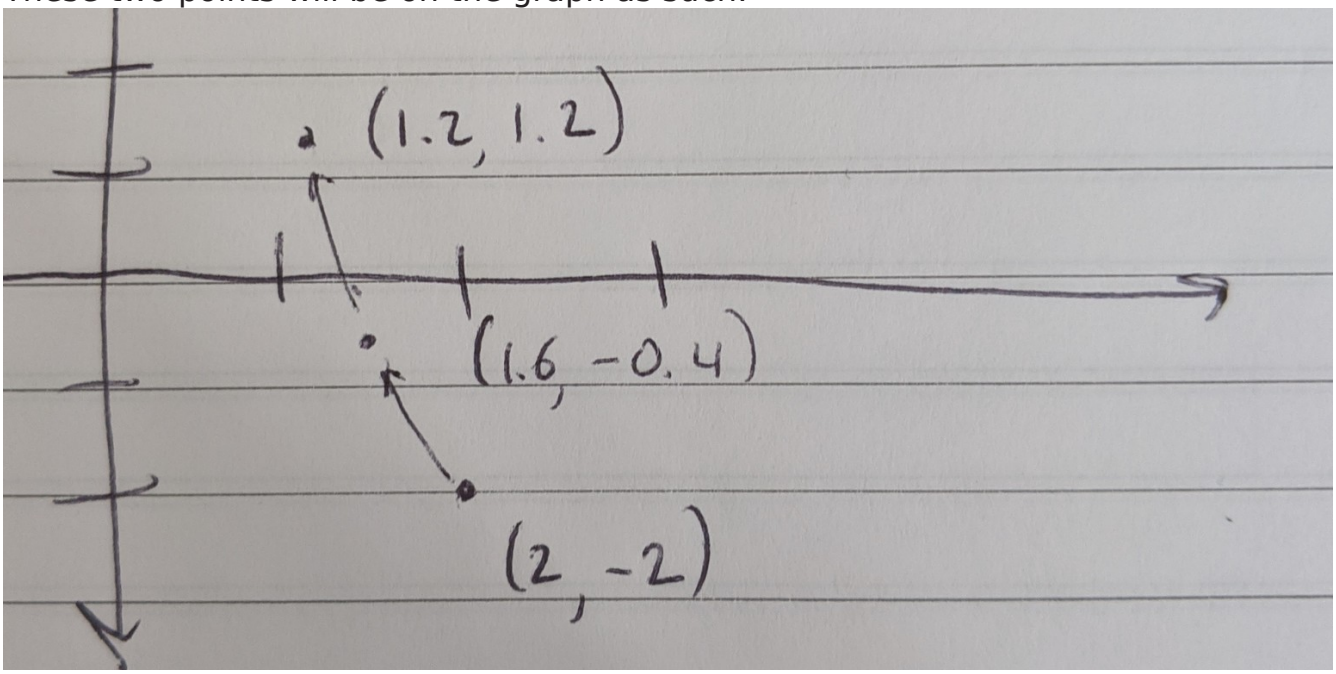
$$\begin{aligned}\mathrm{w}_{1+1} &= 2 - 0.1(4) \\ w_{2+1} &= -2 - 0.1(-16) \\ \mathbf{w}' &= \begin{bmatrix}1.6 & -0.4\end{bmatrix}^T\end{aligned} \qquad \text{(Equation 15)}$$

So we've moved from $\begin{bmatrix}2 & -2\end{bmatrix}^T$ to $\begin{bmatrix}1.6 & -0.4\end{bmatrix}^T$

Now starting at we will go down another step:

$$\begin{aligned} \mathrm{w}_{1+2} &= 1.6 - 0.1(4) \\ w_{2+2} &= -0.4 - 0.1(-16) \\ \mathbf{w}' &= \begin{bmatrix} 1.2 & 1.2 \end{bmatrix}^T \end{aligned} \quad \text{(Equation 16)}$$

These two points will be on the graph as such:



Part 3

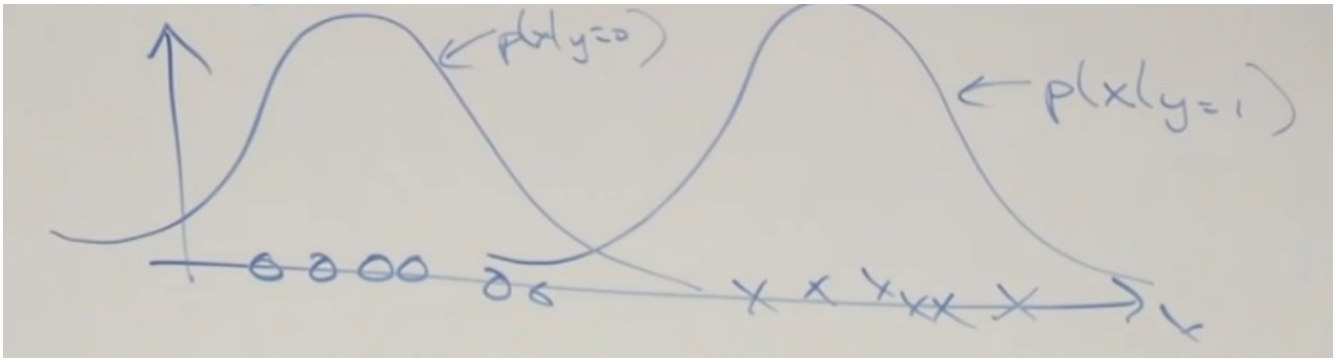
Show that logistic regression is a nonlinear regression problem. Is it possible to treat logistic discrimination in terms of an equivalent linear regression problem? Justify your answer.

Let's say we have two classes. There exists some probability of each glass given y and some training examples . For argument's sake, both and are normally distributed:

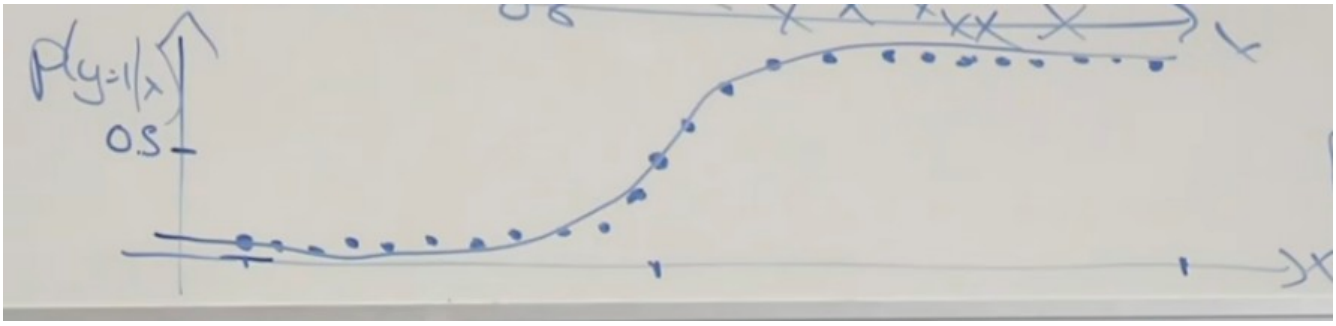
(Equation 17)

Say we had some data that was either 0 or 1 (yes/no, benign/malignant, etc) and say we had just one feature

We could then draw Gaussian curves over each space to see the distribution. As can be seen, if we have a point, say then we can see there is a high probability of it being from the left hand distribution. Same for a point that would be to the right.



If we were to choose one class and then draw these probabilities out on a graph...



And we can see that as the data progresses in x, the probability of y being 1 increases, creating a sigmoid function.

For the second question, if the covariance matrices are the same, that is $\Sigma_i = \Sigma_j$ then yes. Otherwise if the input variables are statistically independent, then also yes. There will be two Gaussian distributions in this case. In the first, even though there will be covariance, they will be simply two ellipses. In the case of the diagonal variance matrix, then (as shown above) they will be statistically independent and two circles with an orthogonal decision boundary.
It would not work in the case of a quadratic discriminant analysis when covariances are not equal. Then the decision boundary would be nonlinear.

References:

Sicotte, X. (2018). Linear and Quadratic Discriminant Analysis. Data Blog. https://xavierbourretsicotte.github.io/LDA_QDA.html

Xiaozhou, Y. (2020). Linear Discriminant Analysis, Explained. Towards Data Science. https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Caihong, L. (n.d.). Logistic Regression and Discriminant Analysis. University of Kentucky. https://education.uky.edu/edp/wp-content/uploads/sites/4/2018/04/Logistic-Regression-and-Discriminant-Analysis.pdf

Rakotomalala, R. (n.d.). Tanagra. http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_LDA_and_Regression.pdf

UC Business Analytics R Programming Guide. Github. http://uc-r.github.io/discriminant_analysis

Mujtaba, H. (2020). An Easy Guide to Gradient Descent in Machine Learning. Greatlearningblog. https://www.mygreatlearning.com/blog/gradient-descent/

Pandey, P. (2019). Understanding the Mathematics Behind Gradient Descent. Towards Data Science. https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e

Wang, C. (2018). How do you find the partial derivative of a function? Towards Data Science. https://towardsdatascience.com/step-by-step-the-math-behind-neural-networks-ac15e178bbd

The Coding Train. (2017). *3.5 Mathematics and Gradient Descent – Intelligence and Machine* Learning. YouTube. https://www.youtube.com/watch?v=jc2IthslyzM

MITOpenCourseWare. (2019). *22. Gradient Descent: Downhill to a Minimum*. YouTube. https://www.youtube.com/watch?v=AeRwohPuUHQ

3Blue1Brown. (2017). *Gradient Descent, how neural networks learn | Deep learning, chapter 2*. YouTube. https://www.youtube.com/watch?v=IHZwWFHWa-w&t=209s

Project Rhea. (2013). Discriminant Functions For The Normal Density - Part 2. Rhea. https://www.projectrhea.org/rhea/index.php/Discriminant_Functions_For_The_Normal(Gaussian)_Density_-_Part_2

giuseppe (https://stats.stackexchange.com/users/36773/giuseppe), What is a Gaussian Discriminant Analysis (GDA)?, URL (version: 2018-10-01): https://stats.stackexchange.com/q/80979

dfhgfh (https://stats.stackexchange.com/users/18287/dfhgfh), Why are Gaussian "discriminant" analysis models called so?, URL (version: 2014-04-23): https://stats.stackexchange.com/q/47167

Jain, P. (2019). *Linear Discriminant Functions: Basics to Perceptron [E20]*. YouTube. https://www.youtube.com/watch?v=inzM1o-uQms

Stanford. (2008). *Lecture 5 | Machine Learning (Stanford).* YouTube. https://www.youtube.com/watch?v=qRJ3GKMOFrE

UCSD. (n.d.). CSE 151 Decision Boundary Equations for Binary Gaussian Generative Model. https://cseweb.ucsd.edu/classes/fa19/cse151-a/cse151_gaussian_decision_boundary.pdf