Machine Learning Assignment 06

313652008 黃睿帆

October 10, 2025

Unanswered question in the class

- When using regression model to approximate periodic functions, will **the Fourier Series** give the best approximation? How complicated that the neuron network of Fourier Series looks like?

Programming assignments (本週 (Week 06) 作業)

1. Classification by using GDA (Quadratic)

Recall that in assignment in week 04, our classfication model performed not so good. This time, we try to use gaussian (quadratic) discriminant analysis. (Note: Python codes here are generated by ChatGPT, and some results part are discussing with ChatGPT.)

(1) How the GDA model works and why it can be used for classification, in particular this data set: Recall that for classification, Gaussian Discriminant Analysis (GDA) is a generative probabilistic model. It assumes that the conditional distribution of the feature vector $x \in \mathbb{R}^n$ given the class label $y \in \{0, 1\}$ follows a multivariate normal (Gaussian) distribution:

$$p(x|y = 0) = \mathcal{N}(\mu_0, \Sigma_0), \quad p(x|y = 1) = \mathcal{N}(\mu_1, \Sigma_1).$$

The prior probabilities of the classes are denoted by $\phi = p(y = 1)$ and $(1 - \phi) = p(y = 0)$. Using Bayes' theorem, we can compute the posterior probability of a class given an observation x:

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 1)p(y = 1) + p(x|y = 0)p(y = 0)}.$$

A new observation x is assigned to the class with the higher posterior probability:

$$\hat{y} = \begin{cases} 1, & \text{if } p(y = 1|x) > p(y = 0|x), \\ 0, & \text{otherwise.} \end{cases}$$

(GDA) 是一種生成式分類模型 (Generative Model)。並不是直接學習「輸入 $x \to$ 類別 y」的關係，而是先學習「每個類別的輸入資料是如何分布的」，然後再使用貝氏定理，來決定一個新的資料點屬於哪個類別。模型假設（Hypothesis function）：每個類別的資料分布都是一個多變量常態分布 (multivariate Gaussian)：

$$p(x|y = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

其中:

1. μ_k = 類別 k 的平均向量 (平均中心位置)
2. Σ_k = 類別 k 的 covariance matrix.
3. d= 特徵維度 (這裡是經度和緯度,$d = 2$)

然後對每個類別計算: $-\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log \pi_k$, π_k 是類別的先驗機率（在訓練集中出現的比例）。模型最後預測：$\hat{y} = \arg\max_k \delta_k(x)$. 同時如同課堂上舉例, 兩個類別的 Covariance matrix(Σ_0, Σ_1) 不同, 決策邊界（兩類分布交界處）會是二次曲線（橢圓形或曲線）。因此也被稱爲 Quadratic Discriminant Analysis (QDA)。如果假設所有類別共變異矩陣相同 ($\Sigma_0 = \Sigma_1 = \Sigma$) 則決策邊界會變成線性直線，爲 Linear Discriminant Analysis (LDA). (此處不適用 LDA 就是因爲 data set 是全台灣的氣象觀測資料, 台灣的邊界並非一直線,LDA Classification 的程度應該有限.)

Why GDA Works for This Dataset ?

GDA is suitable for this dataset because: The features in weather dataset (temperature) can be reasonably modeled by continuous Gaussian distributions within each class. Each class (Like ”High temperature” vs. ”low temperature”) tends to form clusters in feature space, which fits the Gaussian assumption. GDA captures not only the mean differences between classes but also their covariance structure, allowing for non-linear decision boundaries (in the quadratic version).

Dataset 裡的氣象資料是：

1. Longitude (經度): 水平方向的地理位置
2. Latitude (緯度): 垂直方向的地理位置
3. Label (0 or 1): 該位置的溫度是否為有效觀測值

改用 GDA 比較適合這種問題，原因也包括：地理資料通常有空間分布特性 (相近地點溫度成連續分布), 而溫度觀測值缺失（-999）往往集中在某些區域，像是海上 (無氣象觀測值)。而這樣溫度的分布可以用 GDA 的「高斯分布」自然地表示（例如區域與擴散範圍）。

(ChatGPT 建議:) 樣本維度較低 (二維),GDA 對低維資料表現穩定，能畫出清晰的「有效/無效」區域邊界。因此易於可視化與解釋, 我們能清楚看到不同類別的均值 μ_0, μ_1 對應的地理中心點，以及決策邊界 (Decision Boundary) 如何分隔有效與無效區域。

Training method of the GDA Model

The model parameters $\mu_0, \mu_1, \Sigma_0, \Sigma_1$, and ϕ are estimated directly from the training data:

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i, \quad \hat{\Sigma}_k = \frac{1}{N_k} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T, \quad \hat{\phi} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{y_i = 1\}.$$

Once these parameters are estimated, the model can classify any new sample by computing $p(y|x)$ using the learned Gaussian distributions.

(在程式中，我們：將資料分成 80% 訓練集、20% 測試集；在訓練集中估計每個類別的平均向量、Covariance Matrix、與先驗機率；用上面的貝氏分類規則在測試集上預測；並用 Accuracy (準確率) 評估模型表現。準確率：Accuracy= (Number of correct predictions)/(Total predictions).)

Why It Is Effective for Classification

GDA leverages the entire probability distribution of each class rather than just separating them by a simple boundary. This means that even when the class distributions overlap or have different variances, GDA can still find an optimal decision boundary that minimizes misclassification. (程式中畫出的 decision boundary 顯示了模型認為「有效區域」與「無效區域」的分界線。如果邊界曲線較平滑且能分隔兩群點，代表 GDA 成功捕捉到地理位置與有效值之間的統計關係。)

預測結果

以下為模型預測結果:

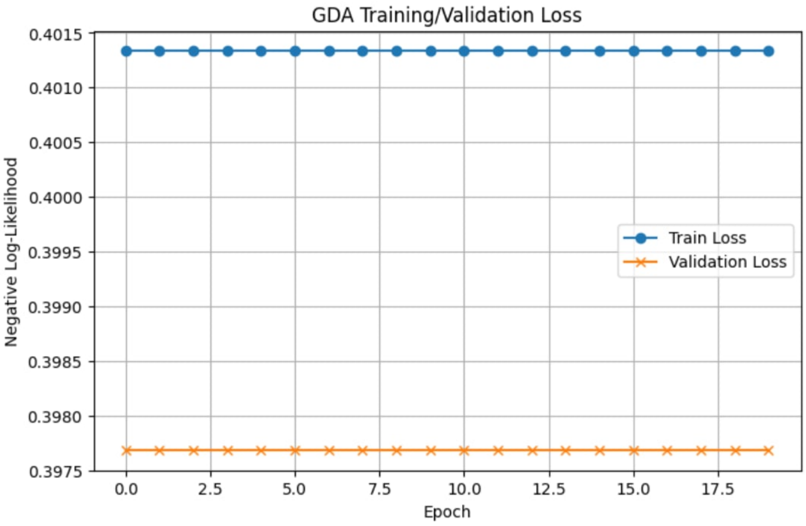
```
Epoch 1/20: train_loss=0.4013, val_loss=0.3977
...(中間皆相同,故省略)
Epoch 20/20: train_loss=0.4013, val_loss=0.3977
=== GDA Classification Test Results ===
Accuracy: 0.8400
MSE: 0.1600
Max Error: 1.0000
```

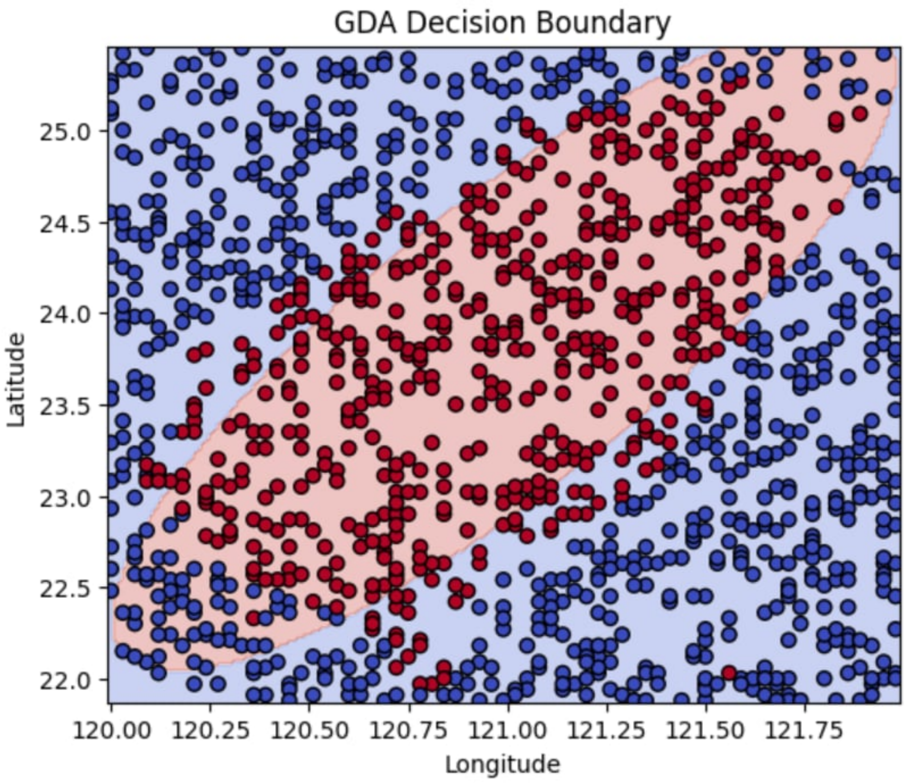
注意到 loss 每次都一樣, 主要原因是程式的這段：

```
for epoch in range(epochs):
    gda.fit(X_train, y_train)
```

的行爲是「每一回合都重新估計平均值與共變異矩陣」，但 GDA（Gaussian Discriminant Analysis）本身沒有 iterative learning（不是像之前的 neural network 一樣逐步更新參數），是個 closed-form analytical model：訓練的時候只要把 $\mu_k, \Sigma_k, P(y=k)$ 三個量算出來就結束，不需要梯度下降或多次迭代。所以每次 fit 都會得到一模一樣的參數，loss 自然也相同。

Accuracy=0.84, 表示模型實際上是運作良好的。GDA 是一種基於「類別條件分布爲 Gaussian」的貝氏分類器，它學到每個類別的高斯分布後，再根據最大後驗概率（MAP rule）分類：$\hat{y} = \arg\max_k P(y=k|x) = \arg\max_k P(x|y=k)P(y=k)$. 資料中，有效與無效點（0 or 1）在空間上應該是有差異的，而這次的 GDA 利用這些分布差異成功分類出了 84% 的測試點。





2. New Regression model

build a regression model that represents a piecewise smooth function. By combining the two models (Classification/Regression model) into a single function. Specifically, let C(x) be your classification model, and R(x) be your regression model. Then construct a model h(x) defined as h(x)= R(x), if C(x)=1; h(x)=-999, if C(x)=0.

Explanation:How we built the combined function? We combined the classification and regression models into one piecewise regression function $h(x)$ defined as:

$$h(x) = \begin{cases} R(x), & if \quad C(x) \quad = 1. \\ -999, & if \quad C(x) \quad = 0. \end{cases}$$

where $C(x)$ is the output of the classification model, trained to predict whether a location (lon, lat) has an observed temperature. $R(x)$ is the output of the regression model, trained to predict the actual temperature value (or -999 when missing).

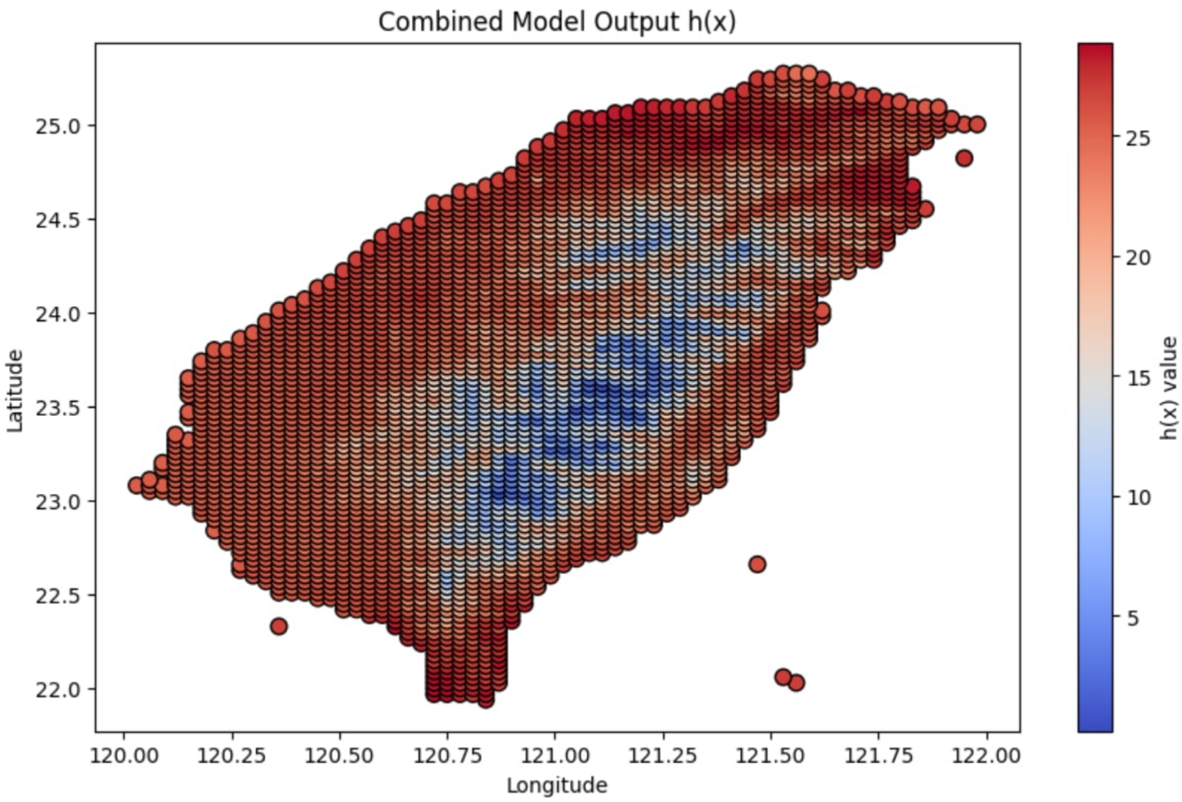
The classifier identifies which points should have valid temperature predictions. The regressor estimates the temperature values for those valid points. The combined function h(x) merges them — producing a "piecewise-smooth" surface that outputs real temperature where data exists, and a missing indicator (-999) where data does not exist. This structure allows the model to represent discontinuities (sharp jumps between valid and missing data) while remaining smooth within valid regions —hence the term piecewise smooth function.

(我們建構組合函數的方式如下: 將分類模型與迴歸模型合併成一個「分段平滑函數（piecewise regression function)」$h(x)$, $C(x)$ 是分類模型的輸出，用來預測該地點（lon, lat）是否有溫度觀測值。$R(x)$ 是迴歸模型的輸出，用來預測實際的溫度值（或缺值時爲 -999）。

因此分類模型用來判斷哪些地點「有有效溫度值」。迴歸模型對那些有效地點進行溫度預測。組合函數 $h(x)$ 將兩者結合，對有觀測的地點輸出實際溫度，對無觀測地點輸出 -999。這樣的模型能夠在「有資料的區域」保持平滑連續，而在「缺資料的區域」明確斷開，因此稱爲「分段平滑函數」.)

模型結果如下:

```
檢查：
C=1 時是否 h=R: True
C=0 時是否 h=-999: True
```



	lon	lat	C_pred	R_pred	h_pred
0	120.84	21.94	1	28.123	28.123
1	120.72	21.97	1	28.574	28.574
2	120.75	21.97	1	28.527	28.527
3	120.78	21.97	1	27.692	27.692
4	120.81	21.97	1	27.059	27.059
5	120.84	21.97	1	28.323	28.323
6	120.72	22.00	1	28.574	28.574
7	120.75	22.00	1	28.592	28.592
8	120.78	22.00	1	28.116	28.116
9	120.81	22.00	1	28.005	28.005
10	120.84	22.00	1	28.532	28.532
11	120.72	22.03	1	28.519	28.519
12	120.75	22.03	1	28.443	28.443
13	120.78	22.03	1	27.210	27.210
14	120.81	22.03	1	27.357	27.357
15	120.84	22.03	1	27.642	27.642
16	120.87	22.03	1	26.857	26.857
17	121.56	22.03	1	26.962	26.962
18	120.72	22.06	1	28.622	28.622
19	120.75	22.06	1	28.535	28.535