ROB313 Assignment 2 – Generalized Linear Models

Objective

The objective of this assignment is to deepen our knowledge of the derivation and software implementation of generalized linear models (GLMs). We use singular value decomposition (SVD) and Cholesky decomposition to create models for both regression and classification datasets, and then analyze model accuracy further on testing sets. We explore standard and kernelized GLMs, proving that if implemented correctly, the results produced are identical. We also notice that the use of basis functions as opposed to simply data points (as in Assignment 1) allows us to approximate trends much more closely.

Code Structure and Strategy

The approach to implementing the generalized linear models was to create different functions to complete each task. For each of the problems, functions were defined to compute common operations required, for example RMSE, the kernel or the gaussian RBF. The 3 GLM implementations are:

1. Standard GLM with singular value decomposition of Φ
   Functions: glm_svd_validation () and glm_svd_test ()
2. Kernelized GLM with Cholesky decomposition of $K + \lambda I$
   Functions: glm_kernelized () and visualize_kernel ()
3. GLM with radial basis function (RBF) and Cholesky decomposition of $K + \lambda I$
   Functions: glm_rbf_validation () and glm_rbf_test ()

For each of the problems, the code was vectorized using Python's numpy module for efficiency in computation. The code could've been slightly better modularized, for example: creating a function for the gram matrix (K) for any standard set of x and z vectors and basis functions. This would've ensured a standardized implementation of the matrices and decreased number of bugs in the code. More information about how to run the code can be found in the README.md file.

Another strategy to help tune the selected basis functions was to determine the frequency of oscillation of the Mauna Loa training data by analyzing a plot of the training points over time. By determining the average period of oscillation, I determined the frequency to be approximately 111.2067 rad/s as can be seen in the global variables of the code.

Discussion

Q1: Least-squares GLM with SVD

The basis function was originally selected to be $[1, x, x^2, xsin(\omega x), xcos(\omega x)]^T$. We know that the first element of the basis functions vector (i.e. ϕ_0) must be 1. The following two elements were selected as x and x^2 because x approximates the height of the data well, and at some points (i.e. the peaks and troughs) the system is close to a parabola. We choose the $xsinx$ and $xcosx$ as a part of the basis function because it is clear that the data is increasing in a sinusoidal pattern.

Upon realizing that in the following problem (q2), we would have to kernelize (i.e. take the dot product) of our vectors of basis functions, I added an $\sqrt{2}$ multiplier to the x basis function. This is so that the polynomial part of the kernel function can simplify to $k_p(x,z) = (1 + xz)^2$. The final basis function, where ω is the frequency of oscillation in the Mauna Loa data, is:

$$\left[1, \sqrt{2}x, x^2, xsin(\omega x), xcos(\omega x)\right]^T$$

We construct a Φ matrix (N x M) where N is the number of data points and M is the number of basis functions, that will be decomposed using singular value decomposition. The computational cost of the SVD of Φ is approximately $O(NM^2 + NM^3)$. Upon implementing this, we get the result of an optimal λ value of **14**, and a test RMSE of **0.103939**. Figure 1 below shows the plot of the actual test values and predictions based on $\lambda = 14$.

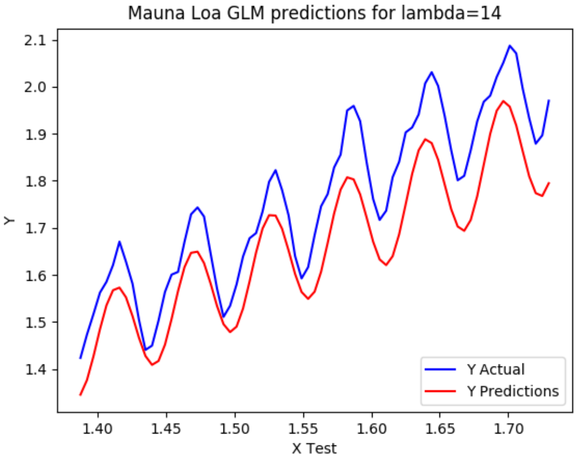


Figure 1, Mauna Loa least-squares GLM prediction with SVD

Q2: Kernelized GLM with Cholesky

Using the same basis function vector as determined for the first problem, we can derive the kernel function by taking the dot product of two basis function vectors for arbitrary vectors x and z as follows:

$$k(x,z) = 1 * 1 + \sqrt{2}x * \sqrt{2}z + xz + xz * \sin(\omega x) * \sin(\omega z) + xz * \cos(\omega x) * \cos(\omega z)$$

$$k(x,z) = (1 + xz)^2 + xz * \cos\big(\omega(x - z)\big)$$

The process for determining the predictions is similar to in question 1, except for when we use the dual representation, our prediction function is dependant on the gram matrix, K, and a variable α instead of the matrix of basis functions, Φ, and w, respectively. The gram matrix and the predictions $(\hat{y})$ are defined as follows:

$$K[i,:] = \left\{k\big(x_{test}, x_{train[1]}\big), \dots, k\big(x_{test}, x_{train[N]}\big)\right\} \in R^N$$

$$\hat{y} = K\alpha$$

$\alpha = (K + \lambda 1)^{-1}y$, where $K + \lambda 1$ is symmetric positive definite, so can be decomposed using Cholesky decomposition in $O(N^3)$. We noticed that this is more computationally complex than decomposing the Φ matrix with SVD in question 1, which can be done in $O(NM^2)$ time, were M = 5. The space complexity of the Cholesky of K here and the SVD of Φ in question 1 are both $O(N^2)$ since our matrix K is (N x N), and U is (N x N) in full SVD.

The results found when using the kernelized GLM with Cholesky decomposition of K are the same, as expected, as the result when using the standard GLM with SVD of Φ. The result for the optimal λ is **14**, with a test RMSE of **0.103939.**

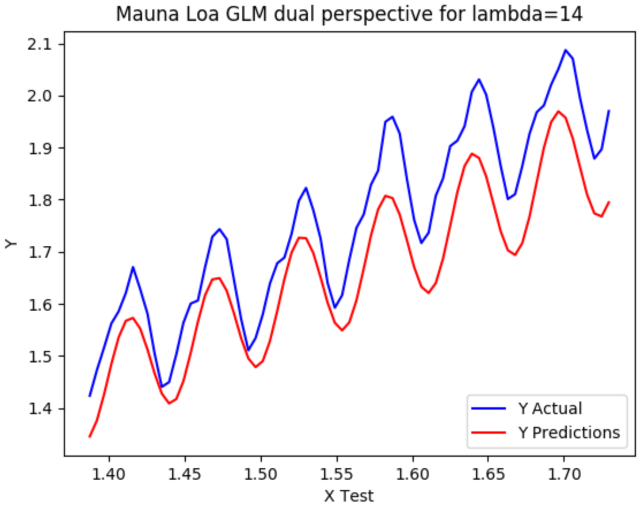


Figure 2, Mauna Loa dual perspective GLM prediction with Cholesky

Visualizing the kernel, we see that it's a constant at value 1 for $k(0, z)$ since all the elements of the kernel (and basis functions) depend on variable x, other than the 1 in $(1 + xz)^2$, due to $\phi_0 = 1$. Once we take a look at $k(1, 1 + z)$, we begin to see the effect of the $xsinx(\omega x)$ and $xcos(\omega x)$ terms in the kernel. The basis function is clearly not translationally-invariant because as we proportionally increase the x and z values in $k(x, z)$ the result of the kernel changes.

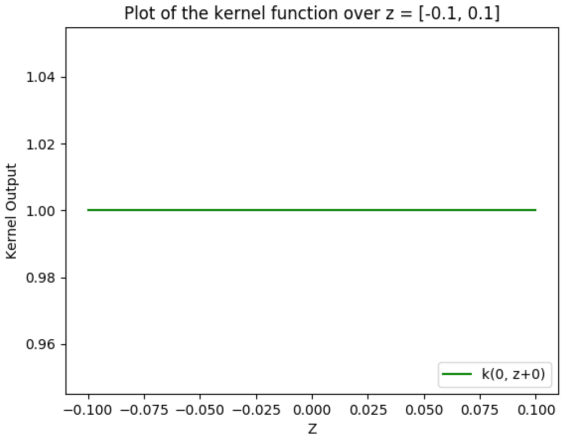


Figure 3, Kernel function $k(0, z)$

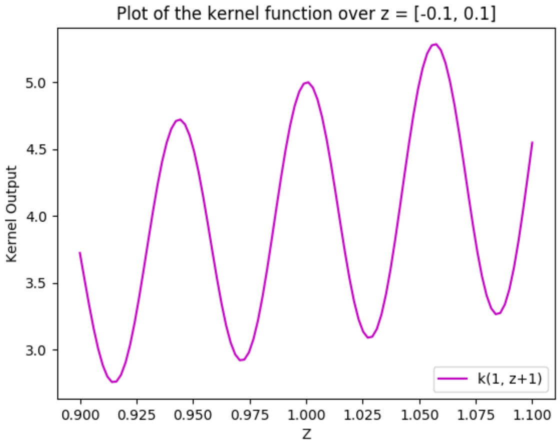


Figure 4, Kernel function $k(1, 1 + z)$

Q3: Kernelized GLM with Gaussian RBF

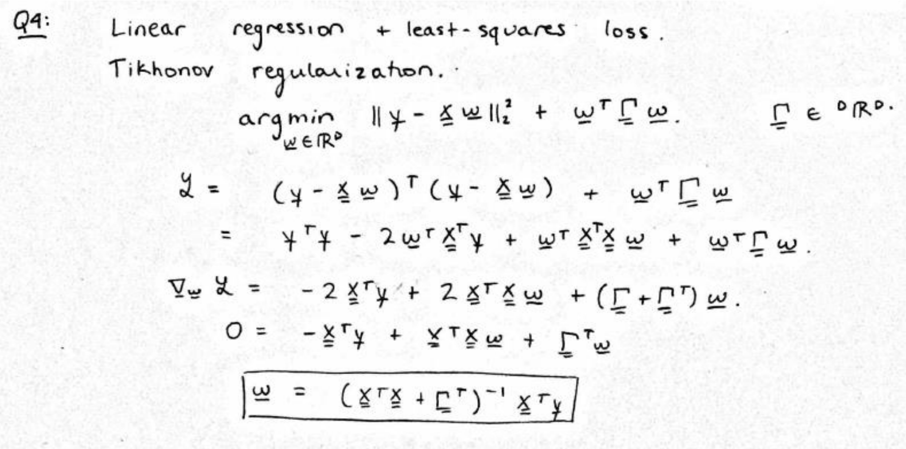
The results for the two regression datasets and one classification set using the Gaussian radial basis function (RBF) are listed in table 1 below. We notice that for the only classification dataset, the results on the testing data are very accurate, whereas on the Mauna Loa set, the test RMSE is actually higher than when created our model with the standard and kernelized GLMs with the basis functions selected in Q1. Note that the optimal values of theta and lambda were selected from the options $\theta = \{0.05, 0.1, 0.5, 1, 2\}$ and $\lambda = \{0.001, 0.01, 0.1, 1\}$.

Table 1, Results for kernelized GLM with Gaussian RBF

Dataset	Optimal θ	Optimal λ	Validation RMSE/Accuracy	Validation RMSE/Accuracy
Mauna Loa (R)	1	0.001	0.12448	0.14977
Rosenbrock (R)	2	0.001	0.19324	0.14812
Iris (C)	0.5	1	87.90%	100%

Q4: Tikhonov regularization

See figure 5 below for the derivation of the minimizer for the GLM with a least-squares loss function with Tikhonov regularization.



Q4: Linear regression + least-squares loss.

Tikhonov regularization.

$$\underset{\underline{w} \in \mathbb{R}^D}{\operatorname{argmin}} \; \|\underline{y} - \underline{\underline{X}}\,\underline{w}\|_2^2 + \underline{w}^T \underline{\underline{\Gamma}}\,\underline{w}. \qquad \underline{\underline{\Gamma}} \in {}^{D}\mathbb{R}^{D}.$$

$$\mathcal{L} = (\underline{y} - \underline{\underline{X}}\,\underline{w})^T (\underline{y} - \underline{\underline{X}}\,\underline{w}) + \underline{w}^T \underline{\underline{\Gamma}}\,\underline{w}$$

$$= \underline{y}^T \underline{y} - 2\underline{w}^T \underline{\underline{X}}^T \underline{y} + \underline{w}^T \underline{\underline{X}}^T \underline{\underline{X}}\,\underline{w} + \underline{w}^T \underline{\underline{\Gamma}}\,\underline{w}.$$

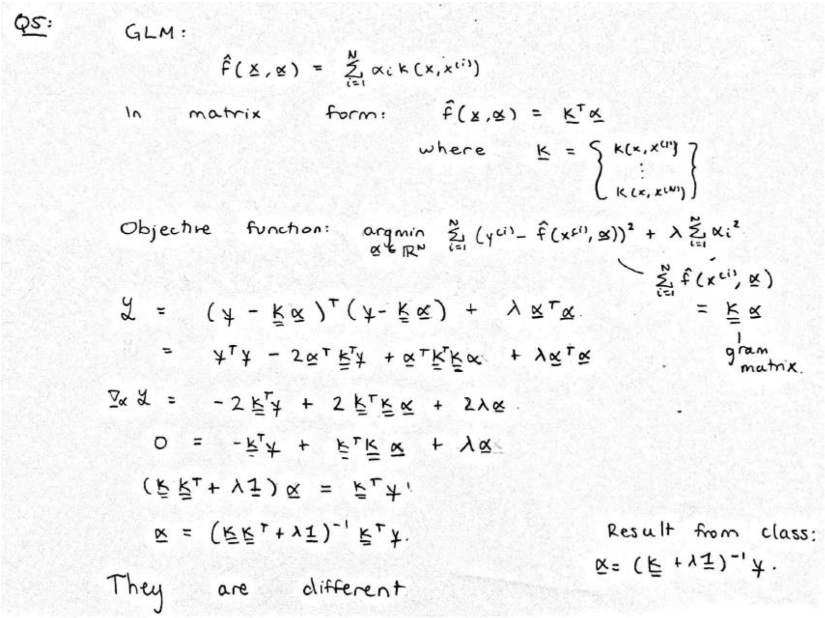
$$\nabla_{\underline{w}} \mathcal{L} = -2\underline{\underline{X}}^T \underline{y} + 2\underline{\underline{X}}^T \underline{\underline{X}}\,\underline{w} + (\underline{\underline{\Gamma}} + \underline{\underline{\Gamma}}^T)\,\underline{w}.$$

$$0 = -\underline{\underline{X}}^T \underline{y} + \underline{\underline{X}}^T \underline{\underline{X}}\,\underline{w} + \underline{\underline{\Gamma}}^T \underline{w}$$

$$\boxed{\underline{w} = (\underline{\underline{X}}^T \underline{\underline{X}} + \underline{\underline{\Gamma}}^T)^{-1} \underline{\underline{X}}^T \underline{y}}$$

Figure 5, Least-squares loss and Tikhonov regularization minimizer

Q5: Dual representation minimization



Q5: GLM:

$$\hat{f}(\underline{x}, \underline{\alpha}) = \sum_{i=1}^{N} \alpha_i k(x, x^{(i)})$$

In matrix form: $\hat{f}(\underline{x}, \underline{\alpha}) = \underline{k}^T \underline{\alpha}$

where $\underline{k} = \begin{Bmatrix} k(x, x^{(1)}) \\ \vdots \\ k(x, x^{(N)}) \end{Bmatrix}$

Objective function: $\underset{\underline{\alpha} \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{i=1}^{N} (y^{(i)} - \hat{f}(x^{(i)}, \underline{\alpha}))^2 + \lambda \sum_{i=1}^{N} \alpha_i^2$

$\sum_{i=1}^{N} \hat{f}(x^{(i)}, \underline{\alpha}) = \underline{\underline{K}}\,\underline{\alpha}$

gram matrix.

$$\mathcal{L} = (\underline{y} - \underline{\underline{K}}\,\underline{\alpha})^T (\underline{y} - \underline{\underline{K}}\,\underline{\alpha}) + \lambda \underline{\alpha}^T \underline{\alpha}.$$

$$= \underline{y}^T \underline{y} - 2\underline{\alpha}^T \underline{\underline{K}}^T \underline{y} + \underline{\alpha}^T \underline{\underline{K}}^T \underline{\underline{K}}\,\alpha + \lambda \underline{\alpha}^T \underline{\alpha}$$

$$\nabla_{\alpha} \mathcal{L} = -2\underline{\underline{K}}^T \underline{y} + 2\underline{\underline{K}}^T \underline{\underline{K}}\,\underline{\alpha} + 2\lambda\underline{\alpha}.$$

$$0 = -\underline{\underline{K}}^T \underline{y} + \underline{\underline{K}}^T \underline{\underline{K}}\,\underline{\alpha} + \lambda \underline{\alpha}$$

$$(\underline{\underline{K}}\,\underline{\underline{K}}^T + \lambda \underline{\underline{1}})\,\underline{\alpha} = \underline{\underline{K}}^T \underline{y}$$

$$\underline{\alpha} = (\underline{\underline{K}}\,\underline{\underline{K}}^T + \lambda \underline{\underline{1}})^{-1} \underline{\underline{K}}^T \underline{y}.$$

Result from class:

$$\underline{\alpha} = (\underline{\underline{K}} + \lambda \underline{\underline{1}})^{-1} \underline{y}.$$

They are different

Figure 6, Least squares loss minimization with alpha regularizer

As you can see in Figure 6, the values of α are not the same from the minimization previously derived as the one in class. This is because when α is derived in class, the starting minimization function uses $w^T w$ as the regularization function and Φw as the estimation vector. Now we are beginning with assuming that some alpha exists and solving for that. Therefore, we use $\alpha^T \alpha$ as the regularization function and $K\alpha$ as the prediction vector.

What we actually did there was replace the w for α and the Φ for K in the normal GLM with ridge regression problem, so we got the same equation format, but they will not be same w/α values.