HW5: Kernels, SVMs, PCA, and Recommender Systems

Part I: HW5 Concept Questions

Problem 0: Conceptual Questions about SVMs

0a: A combination of very high gamma and C values will lead to overfitting on the training set and result in zero training error. Smaller gamma allows a broader consideration of points, yielding smoother decision boundaries, while larger gamma values focus on nearby points, leading to more complex decision boundaries. In the case of C, smaller values prioritize a large margin but allow more misclassifications, while larger C values prioritize reducing misclassifications with a smaller margin. The interplay of high gamma and C values results in a model that fits the training data too closely, leading to poor generalization on unseen data.

0b: False. First, the L2 penalty minimizes the sum of the squared values of the weight coefficients, which encourages smaller weights, but it does not force the weights to be zero. Second, the sparsity of α in the dual formulation of SVM does not imply the sparsity of the weight vector w in the primal formulation of SVM. Find the weight vector, w, below:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

Although the set of α values is sparse (many of them are zero), each non-zero α is associated with a feature vector x, and when these are summed to find w, the result is typically not sparse.

.

Problem 1: Conceptual Questions about Principal Components Analysis (PCA)

1a. The optimal number of components would be K = F, where F is the number of features in the original data. This PCA setup essentially retains all the principal components, and allows you to perfectly reconstruct the training data because no information is discarded. However, this is undesirable because the primary goal of PCA is to reduce the dimensionality while preserving most of the data. Thus, choosing K = F defeats this purpose, leading to overfitting and increasing the computational cost.

1b. Yes, Stella is correct. She appropriately follows the necessary steps to project the test dataset X’ using matrix W learned from the training set. These steps include calculating the mean vector by averaging the test vectors along each feature dimension, centering the test data by subtracting the mean from each vector, and then projecting the centered test vectors onto the K-dimensional space defined by the basis vectors of W.

HW5: Kernels, SVMs, PCA, and Recommender Systems

Problem 2: Conceptual Questions about Recommender Systems

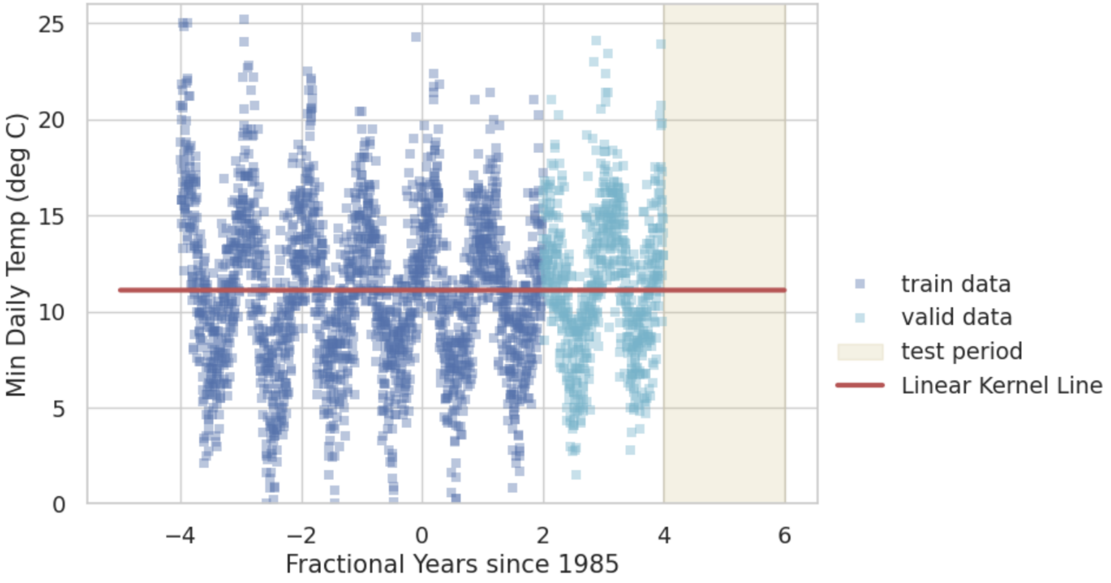
2a: If an item has a negative bias, it indicates that the item tends to be rated lower than the overall rating across all items. It might suggest that the item is of lower quality, less popular, or less suited to the tastes of the majority of the users in the system. Conversely, if the bias value increases, it implies that the item is not as poorly rated compared to the average. And, if the bias value were to become positive, it would mean the item is generally liked more than the average item in the system.

2b: False. The code does not correctly compute the squared error on the validation set for two reasons: Firstly, it includes all the ratings in the matrix, not just those in the validation set. Secondly, it does not exclude the placeholder value of -1 that represents missing ratings. The calculation of squared error should be limited to actual, observed ratings in the validation set.

Part II: HW5 Case Study: Kernelized Linear Regression for Temperature Forecasting

Problem 4: Linear Kernel + Ridge Regression

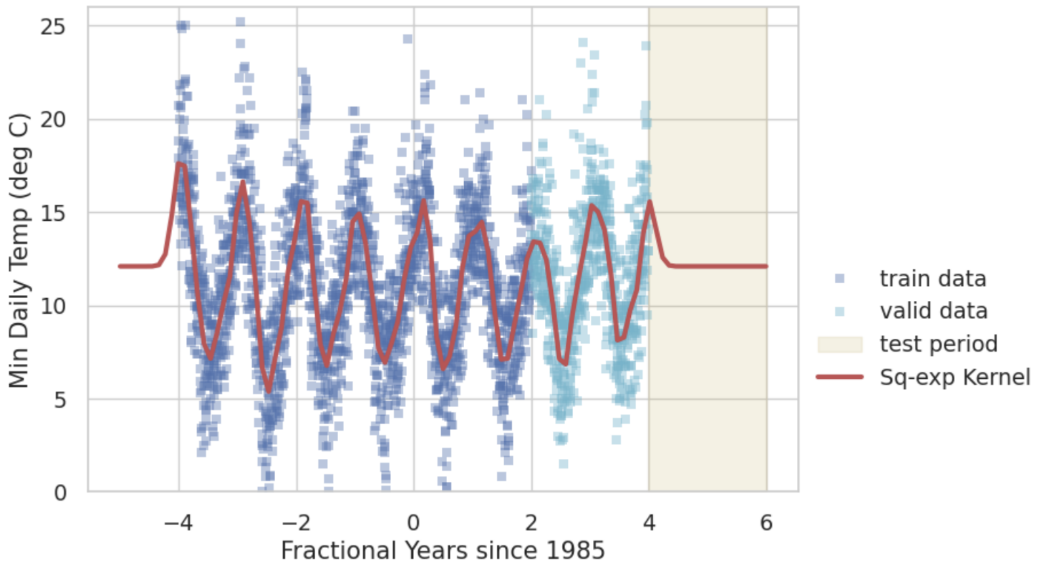
Figure 4:



Short Answer 4: Figure 4 demonstrates that Linear Kernel doesn't capture the relationship between the min temperature and time well. The model plots a horizontal line that goes through the center of the training data. It does a poor job in both interpolating and extrapolating because the variables have a much more complex relationship than a linear one, which is evident from the fact that the overall trend is a horizontal zigzag.

Problem 5: Squared-Exponential Kernel + Ridge Regression

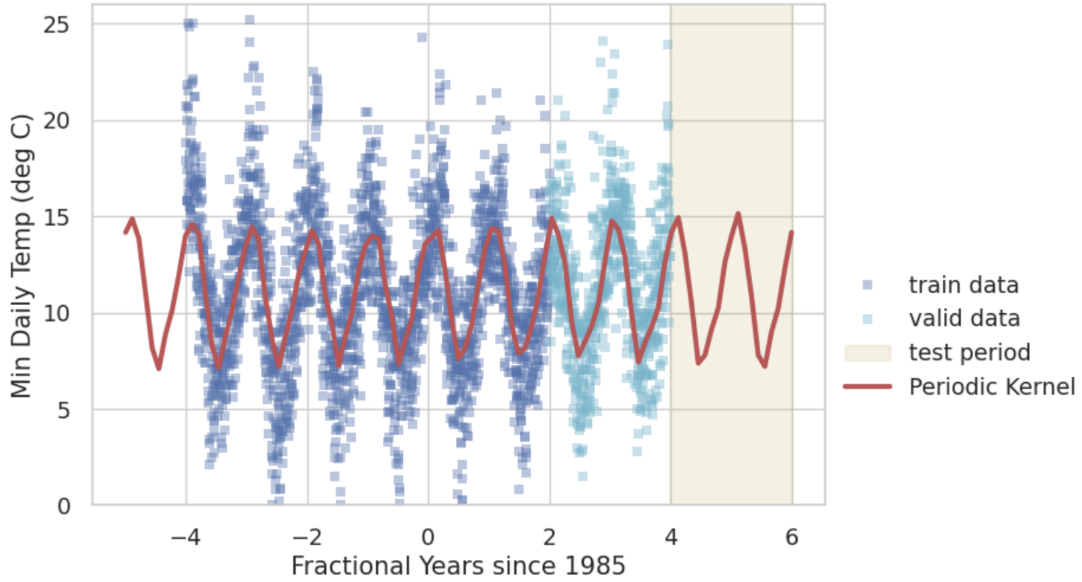
Figure 5:



Short Answer 5: In Figure 5, we observe that the model interpolates well within the range of the training and the validation data. Yet, it is not perfect as it fails to capture the peaks and troughs on the zigzag shape of the data. In comparison, the model fails to extrapolate the unseen data since the regression line flattens out on the training data. This happens because of rapid decay in the kernel with distance from the training points in the squared exponential kernel. This leads to the kernel to become 0, which implies that yhat equals to the bias, which is essentially a horizontal line.

Problem 6: Periodic Kernel + Ridge Regression

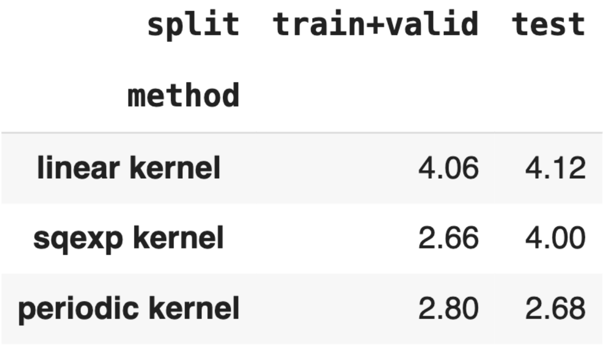
Figure 6:



Short Answer 6: In Figure 6, we see that the model with periodic kernel regression does a similar job in interpolating the training and validation data. Though, the plot of the periodic kernel is a little bit underfitting compared to the plot of the squared-exponential kernel since the peaks are slightly lower and throughs are slightly higher than in Figure 5. However, the model seems to be extrapolating much better than the other two models, maintaining the zigzag plot/periodic cyclic pattern over the test set. This plot indicates that the periodic kernel is particularly suited for datasets where the target variable exhibits a cyclical/periodic nature, which seems to be the case in the data for Figure 6.

Problem 7: Final Showdown

Table 7:



split	train+valid	test
method		
linear kernel	4.06	4.12
sqexp kernel	2.66	4.00
periodic kernel	2.80	2.68

Short Answer 7a: As seen in Table 7, the model with periodic kernel has the best performance, with the lowest error at 2.68. Following it, comes the sqexp kernel model with an error of 4.00 and linear kernel model with an error of 4.12. Based on the course concepts, the performance ordering of the different kernel type models make sense. The periodic kernel has the best performance due to its ability to capture cyclic/periodic patterns present in the data, extrapolating very well. Whereas, the sqlexp kernel model performs poorly on the test set due to its fast decaying nature of the kernel for unseen data, and the linear kernel is too simple to describe the trend in the data. This table implies that picking a kernel compatible with the underlying pattern in the dataset is very important for building a high performing model.

Short Answer 7b: The leave-some-years-out method is more preferable for this task and dataset. With this method, the model is trained on data from earlier years and validated on data from later years. This mimics how we would use the model in real life, which is predicting the future based on the past information. In an at-random method, the model is trained on data randomly picked from training and validation data, which implies that some future data could be used to predict the past. This contradicting how the model would be used in real life.

: