Names: Jake Stamell
UNI: jfs2167

Homework 2

1. Problem 1

   (a) When solving for $\hat{\pi}$, the only term that includes π is the first one. Also, call the objective function L. We will see that the estimate for π is the proportion of times in our data that we observe class 1.

$$\begin{aligned}
\hat{\pi} &= \arg\max_{\pi} \sum_{i=1}^{n} ln\left(p(y_i|\pi)\right) \\
&= \arg\max_{\pi} \sum_{i=1}^{n} y_i ln(\pi) + (1-y_i) ln(1-\pi) \\
\frac{\partial}{\partial \pi} L &= \sum_{i=1}^{n} \frac{y_i}{\pi} + \frac{1-y_i}{1-\pi} = 0 \\
&= \frac{1}{\pi} \sum_{i=1}^{n} y_i - \frac{n}{1-\pi} + \frac{1}{1-\pi} \sum_{i=1}^{n} y_i = 0 \\
&= (1-\pi) \sum_{i=1}^{n} y_i = n\pi - \pi \sum_{i=1}^{n} y_i = 0 \\
\hat{\pi} &= \frac{1}{n} \sum_{i=1}^{n} y_i
\end{aligned}$$

   (b) To solve for $\hat{\lambda}_{y,d}$ in the general case, we first notice that the prior is symmetric for all λ. Next, we notice that when we take the derivative to solve for each individual λ, the only x terms that matter are the ones for that class of y. Lastly, it does not depend on the first term in L since that only involves π. (I will also ignore terms in the λ and x distributions that do not include λ as they will drop out when we take the derivative anyways.) For notational convenience, define: $\sum_{i:y_i=y} 1 = N_y$. Therefore, we can solve the following:

$$\begin{aligned}
\hat{\lambda}_{y,d} &= \arg\max_{\lambda_{y,d}} \; ln\; p(\lambda_{y,d}) + \sum_{i:y_i=y} ln\; p(x_{i,d}|\lambda_{y,d}) \\
&= \arg\max_{\lambda_{y,d}} (\alpha-1) ln(\lambda_{y,d}) - \beta\lambda_{y,d} + \sum_{i:y_i=y} x_{i,d} ln(\lambda_{y,d}) - \lambda_{y,d} \\
&= \arg\max_{\lambda_{y,d}} (\alpha - 1 + \sum_{i:y_i=y} x_{i,d}) ln(\lambda_{y,d}) - (\beta + N_y)\lambda_{y,d} \\
\frac{\partial}{\partial \lambda_{y,d}} L &= \frac{\alpha - 1 + \sum_{i:y_i=y} x_{i,d}}{\lambda_{y,d}} - (\beta + N_y) = 0 \\
\hat{\lambda}_{y,d} &= \frac{\alpha - 1 + \sum_{i:y_i=y} x_{i,d}}{\beta + N_y} = \frac{1 + \sum_{i:y_i=y} x_{i,d}}{1 + N_y}
\end{aligned}$$

   We now see that the MAP estimate for $\lambda_{y,d}$ is simply a balance between the prior and the average number of times that x_d is seen in class y.

2. Problem 2

(a) Precision = 87%

		model prediction	
		0	1
ground truth	0	2295	492
	1	99	1714

(b) The stem plot below shows the values of λ_1 relative to λ_0 for all features. (I found this plot easier to interpret than placing them side by side, but have also included that for completeness.) Features 16 and 52, which correspond to 'free' and '!' respectively, are highlighted in red. From this chart, we can see that spam emails have these words/punctuation at a higher rate than non-spam emails. This matches our intuition about words that would appear more frequently in spam. We can also compare these two with features 19 and 21 (corresponding to 'you' and 'your'), which interestingly have much higher λ values for spam emails.

Figure 1: Comparison of λ values for spam vs. non-spam emails

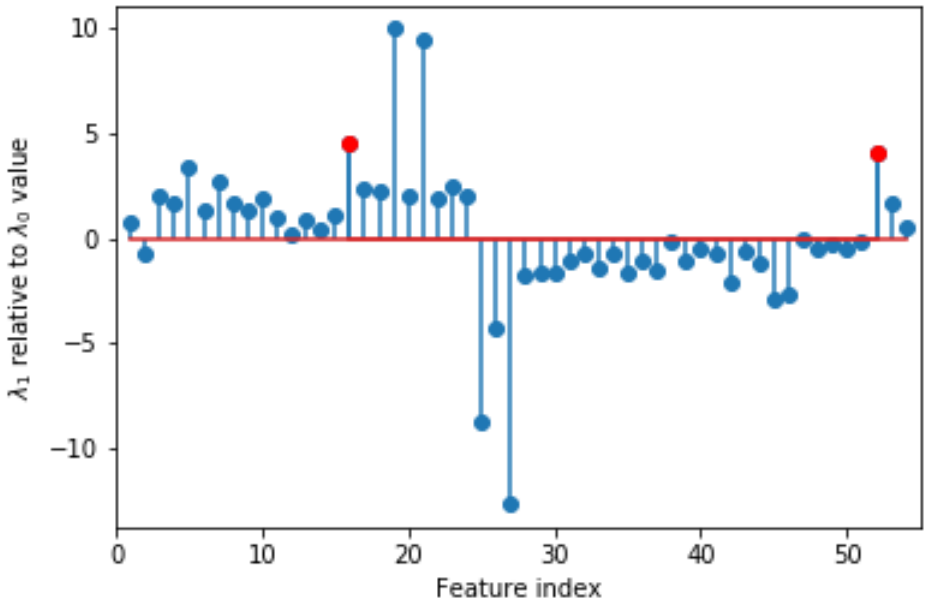
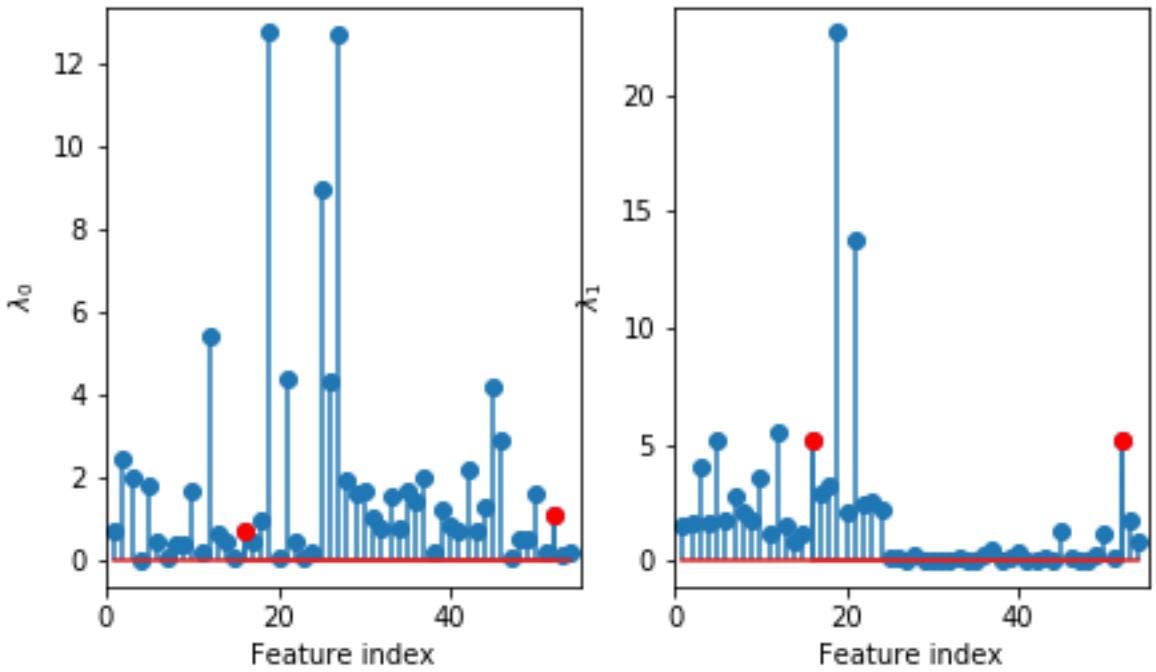
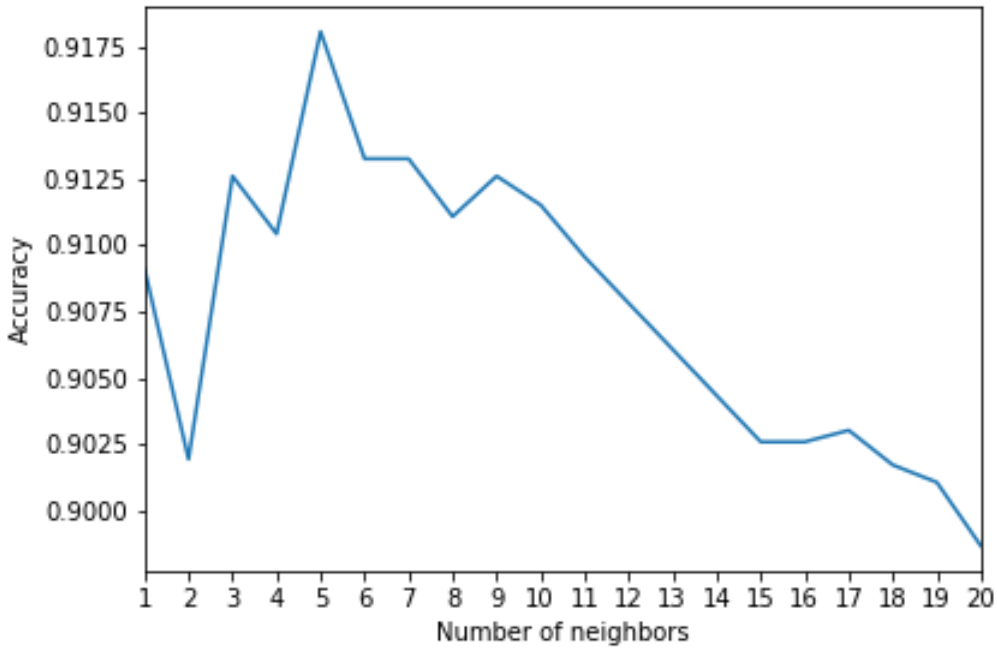


Figure 2: Raw λ values for spam vs. non-spam emails



(c) The k-NN classifier performs best with $k = 5$, although it performs better than the naive bayes model for all choices of k. Interestingly, the even values of k (where ties are decided by random choice in my algorithm) typically perform worse than the odd values (which are always determined by voting).

Figure 3: k-NN prediction accuracy by choice of k



3. Problem 3

(a) See attached code.

(b) See table below. The strongest test set RMSE is seen around low values of σ^2 and higher values of b.

		σ^2									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
b	5	1.966	1.933	1.923	1.922	1.925	1.929	1.935	1.941	1.947	1.953
	7	1.920	1.905	1.908	1.916	1.925	1.934	1.942	1.950	1.958	1.965
	9	1.898	1.903	1.918	1.933	1.946	1.957	1.967	1.976	1.985	1.992
	11	1.891	1.915	1.939	1.958	1.973	1.986	1.996	2.006	2.014	2.021
	13	1.896	1.936	1.965	1.986	2.001	2.014	2.024	2.033	2.041	2.049
	15	1.910	1.960	1.991	2.012	2.027	2.039	2.049	2.058	2.066	2.073

(c) The best value is 1.891 and occurs at $b = 11$ and $\sigma^2 = 0.1$, which is better than the ~ 2.1 found across the polynomial regularized regression in homework 1. While the Gaussian process does provide improved RMSE performance, it is more difficult to interpret and explain, particularly to a non-technical audience. Regression has the advantage of being able to easily lay out the features with weights corresponding to how important they are in predicting the response. Furthermore, the computation time for prediction with linear regression is faster once you have learned the weights (i.e. you just need to take a dot product of your new input vector with the weights vs. calculating the kernel between the new input vector and all training vectors as in the Gaussian process).

(d) From this plot, we can see that the predictive mean tracks (what looks like) the average of the data for that value of x. Because of the Gaussian kernel, this does not appear to be linear in this space.

Figure 4: Gaussian process predictive mean vs. data for training set