1 The growth function

Question 4: Unfortunately, I have not had the time to complete this part of the assignment.

2 Airline Revisited

As I understand the question, we should assume that the event of whether a passenger with a flight reservation shows up or not is drawn from a bernoulli distribution with a given probability p of showing up, and a probability $1 - p$ of not showing up. Using this assumption, we have to bound the probability that the following two events both happen:

- Event A: We i.i.d sample 1000 flight reservations from the distribution, and the passengers show up in exactly in 95 percent of them.
- Event B: We i.i.d sample 100 flight reservations from the distribution, and the passengers show up in all of them.

Since event A and event B are independent, then $P_p(A \wedge B) = P_p(A)P_p(B)$, where p is the probability of a single passenger showing up for a reservation. $P_p(A)P_p(B)$ probability is going to be very low, however, since the probability of observing *exactly* 95 percent of the passengers in the 10000 reservations not showing up is very low, no matter what p is. To be precise, this probability is equal to the value of the binomial distribution of 9500 success in 10000 experiments with a success probability p on each experiment. I denote this value by $\text{bin}(9500, 10000, p)$. In other words, we have:

$$P_p(A) = \text{bin}(9500, 10000, p) \tag{1}$$

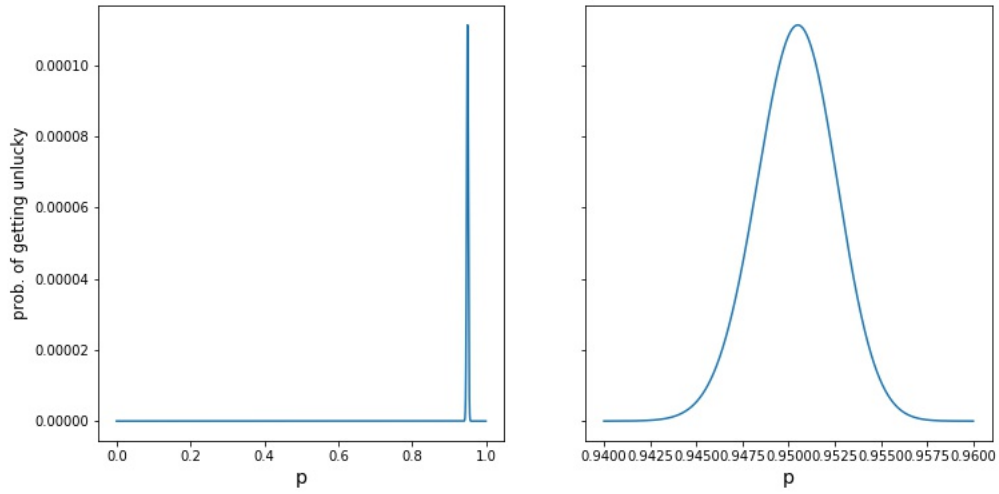
As in question 1, we also have that:

$$P_p(B) = p^{100} \tag{2}$$

All in all, we therefore have that:

$$P_p(A \wedge B) = P_p(A)P_p(B) = \text{bin}(9500, 10000, p)p^{100} \tag{3}$$

Here are two plots of the probability of the flight company getting unlucky, that is $P_p(A \wedge B)$, against the value of p. The plot on the right is just a zoomed in version of the plot on the left, only focusing on p from 0.94 to 0.96:



The value of $P_p(A \wedge B)$ is maximized for $\tilde{p} = 0.95005$ with a value of $P_{\tilde{p}}(A \wedge B) = 0.00011$. This means that for all $p \in [0, 1]$, we have that $P_p(A \wedge B) \leq 0.00011$. However, this says very little about anything interesting, since the low probability just comes from the fact that we are taking A as the event of observing *exactly* 95 percent of the passengers not showing up. A flight company would be completely uninterested in this probability, since they would never have a decision procedure of implementing the reservation policy of overbooking with 1 passenger, if and only if they observe exactly event A happening.

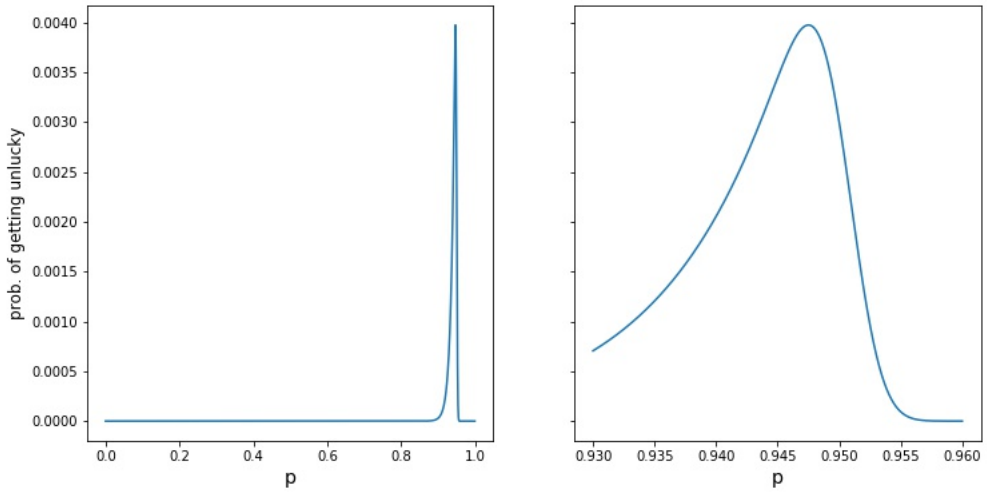
Another slighty more interesting, but still not very realistic scenario, is to ask for the probability of observing both event $\hat{A}$ and event B, where

- Event $\hat{A}$: We i.i.d sample 1000 flight reservations from the distribution, and the passengers show up in exactly in less than or exactly 95 percent of them.
- Event B: We i.i.d sample 100 flight reservations from the distribution, and the passengers show up in all of them.

Here we have that $P_p(\hat{A})$ is just the value of the cumulative binomial distribution of 9500 success in 10000 experiments with a success probability p on each experiment. I denote this value by $\text{cumbin}(9500, 10000, p)$. We therefore have:

$$P_p(\hat{A} \wedge B) = P_p(\hat{A})P_p(B) = \text{cumbin}(9500, 10000, p)p^{100} \tag{4}$$

Here are two plots of the probability of the flight company getting unlucky in this setting against the value of p:



The value of $P_p(\tilde{A} \wedge B)$ is maximized for $\tilde{p} = 0.94475$ with a value of $P_{\tilde{p}}(\hat{A} \wedge B) = 0.0039$. This means that for all $p \in [0, 1]$, we have that $P_p(\hat{A} \wedge B) \leq 0.0039$.

3 SVMs

3.1 Data normalization

Let μ_i and σ_i be the empirical mean and standard deviation of the 98 measurements of the i^{th} feature in the training data. I define my normalization function $f_{norm} : \mathbb{R}^{22} \to \mathbb{R}^{22}$ by

$$f_{norm}(x) = \left(f^1_{norm}(x_1), ..., f^{22}_{norm}(x_{22})\right) \tag{5}$$

where

$$f^i_{norm}(x_i) = \frac{x_i - \mu_i}{\sigma_i} \tag{6}$$

I implement this function with *StandardScaler* object in the *sklearn.preprocessing* module of the *scikitlearn* library in Python.

Here is a table of the mean and standard deviation of each of the features in the training data, before and after the normalization. Each entrance is rounded to the 4^{th} decimal:

Feat.	mean	std	norm. mean	norm. std
1	155.9604	44.3036	0.0	1.0
2	204.8212	98.1520	0.0	1.0
3	115.0586	45.7556	0.0	1.0
4	0.0060	0.0040	0.0	1.0
5	0.0000	0.0000	0.0	1.0
6	0.0032	0.0024	0.0	1.0
7	0.0033	0.0023	0.0	1.0
8	0.0096	0.0071	0.0	1.0
9	0.0277	0.0159	0.0	1.0
10	0.2624	0.1627	0.0	1.0
11	0.0147	0.0087	0.0	1.0
12	0.0166	0.0101	0.0	1.0
13	0.0220	0.0133	0.0	1.0
14	0.0440	0.0260	0.0	1.0
15	0.0226	0.0298	0.0	1.0
16	22.0007	4.0632	0.0	1.0
17	0.4948	0.1015	0.0	1.0
18	0.7157	0.0558	0.0	1.0
19	-5.7637	1.0304	0.0	1.0
20	0.2148	0.0758	0.0	1.0
21	2.3658	0.3694	0.0	1.0
22	0.1997	0.0816	0.0	1.0

Here is the same kind of table for the test data:

Feat.	mean	std	norm. mean	norm. std
1	152.4790	37.9096	0.0786	0.8557
2	189.3091	82.9900	0.1580	0.8455
3	117.6037	40.8634	0.0556	0.8931
4	0.0064	0.0056	0.1132	1.4108
5	0.0000	0.0000	0.0716	1.2908
6	0.0034	0.0035	0.0869	1.4618
7	0.0036	0.0032	0.1157	1.3865
8	0.0102	0.0104	0.0870	1.4621
9	0.0317	0.0212	0.2490	1.3311
10	0.3023	0.2200	0.2452	1.3524
11	0.0167	0.0113	0.2296	1.3105
12	0.0192	0.0135	0.2509	1.3334
13	0.0262	0.0197	0.3166	1.4799
14	0.0500	0.0340	0.2296	1.3105
15	0.0271	0.0486	0.1491	1.6319
16	21.7701	4.7401	0.0568	1.1666
17	0.5023	0.1057	0.0736	1.0405
18	0.7205	0.0544	0.0868	0.9754
19	-5.6042	1.1365	0.1548	1.1030
20	0.2383	0.0885	0.3107	1.1674
21	2.3981	0.3933	0.0874	1.0647
22	0.2135	0.0971	0.1686	1.1894

As can be seen from the tables, each feature of the normalized training data ends up with a mean of 0 and a standard deviation of 1. Since the empirical means and standard deviations of the training data features are also used to normalize the test data features, the normalized test data features do not end up with means of exactly 0 or standard deviations of exactly 1. However, the test data features that start with means far way from 0 and standard deviations far away from 1 still end up with normalized means and standard deviations that are very much closer to 0 and 1 than the unnormalized ones.

3.2 Model selection using grid-search

Let

$$\mathcal{C} = \{0.01, 0.1, 1, 10, 100, 1000, 10000\} \tag{7}$$

$$\mathcal{Y} = \{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\} \tag{8}$$

Let $(X_1, ..., X_5)$ be a random split of the training data X into 5 subsets of as equal size as possible. Let

$$h^i_{C,\gamma} \quad i \in \{1, ..., 5\},\ C \in \mathcal{C},\ \gamma \in \mathcal{Y} \tag{9}$$

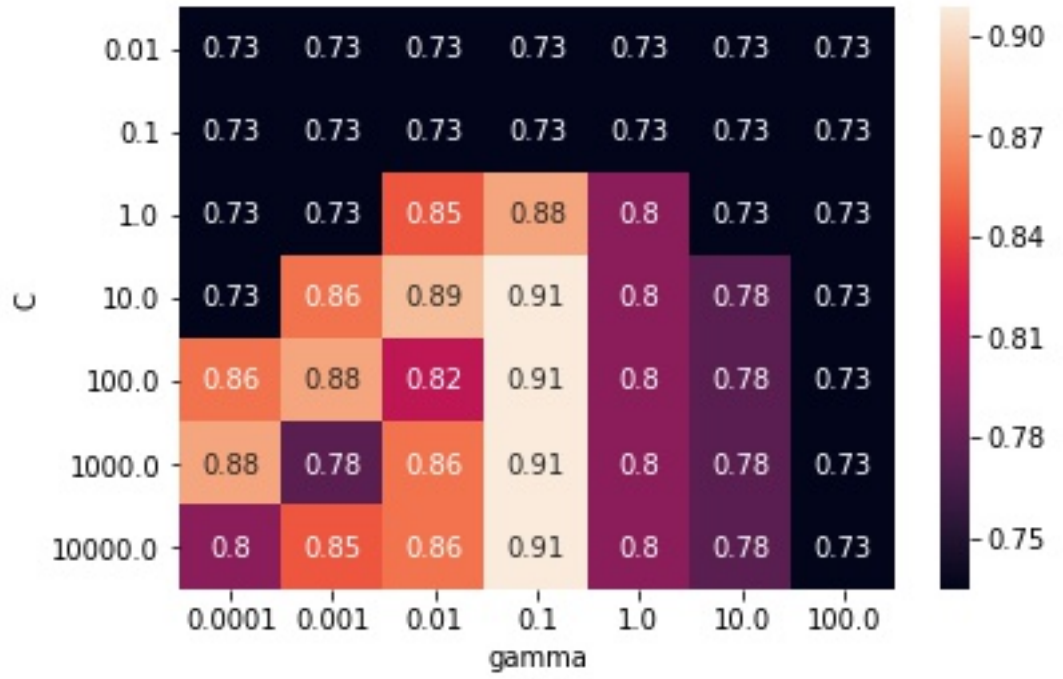
be the classifier, we get by running an svm with a radial kernel on the union of all the splits of the training data except X_i with the hyperparametres set to C and γ. We can now run the classifier $h^i_{C,\gamma}$ on the set X_i and observe the accuracy of $h^i_{C,\gamma}$ on X_i, which we can call the i^{th} validation score for the combined hyperparametres (C, γ). Let us denote this quantity by $S^i_{C,\gamma}$.

For any combination of hyperparametres $(C, \gamma) \in \mathcal{C} \times \mathcal{Y}$ we can now define the 5-fold cross validation score $S_{C,\gamma}$ on the training set X as the average of each of the i^{th} validation scores of the given pair (S, γ):

$$S_{C,\gamma} = \frac{1}{5} \sum_{i=1}^{5} S^i_{C,\gamma} \tag{10}$$

When we use 5-fold cross validation grid search to choose a pair of hyperparametres (C, γ) from the parameter grid $\mathcal{C} \times \mathcal{Y}$, then we first calculate the 5-fold cross validation score $S_{C,\gamma}$ for all $(C, \gamma) \in \mathcal{C} \times \mathcal{Y}$. We then choose the pair (C, γ) with the highest validation score.

I have implemented 5-fold cross validation grid search with the object *GridSearchCV* from the module *sklearn.model_selection* in the *scikitlearn* library in Python. Here is a heatmap of the resulting cross validation score - rounded to the nearest 2 decimals - for all $(C, \gamma) \in \mathcal{C} \times \mathcal{Y}$:



The pair with the maximum 5-fold cross validation score on the given training set is $(C = 10, \gamma = 0.1)$ with the score 0.9082. The radial kernel svm trained on the entire training set X with this pair of hyperparametres obtains an accuracy of 0.9072 on the test set.

3.3 Inspecting the kernel expansion

Unfortunately, I have not had the time to complete this part of the assignment.