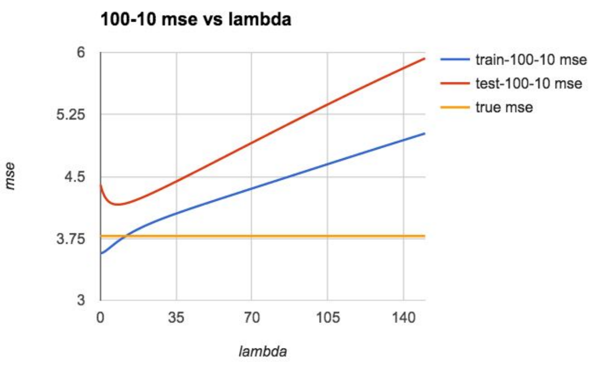
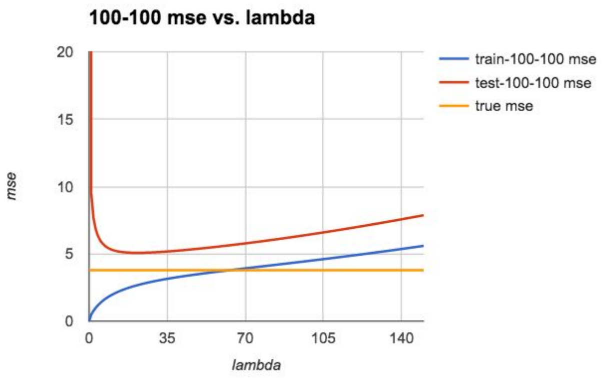
Report for Project 2

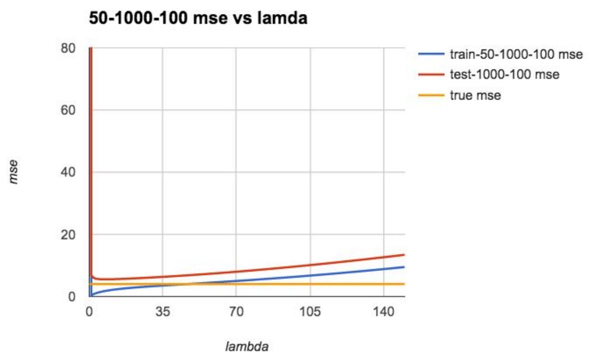
Zhaokun Xue

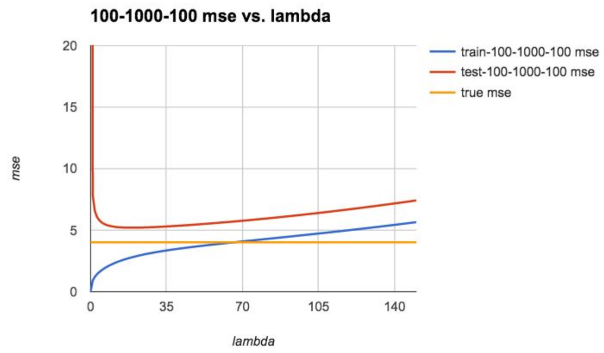
● Task 1 Regularization

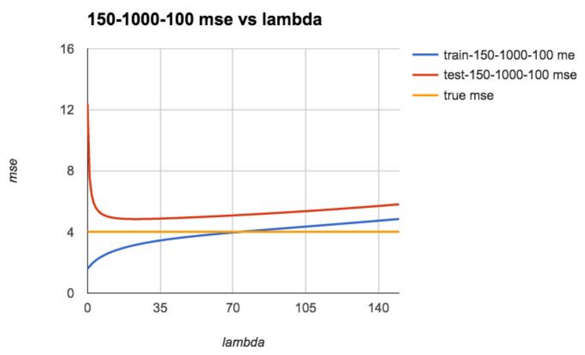
○ Training set MSE and test set MSE figures

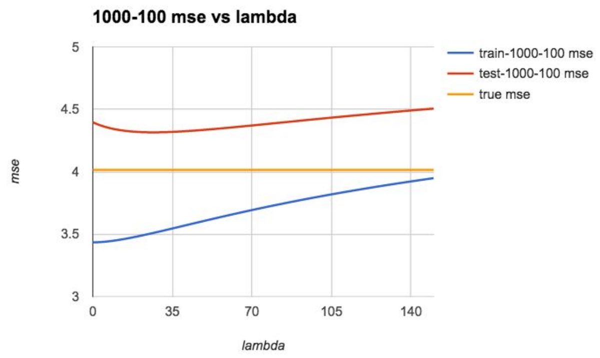


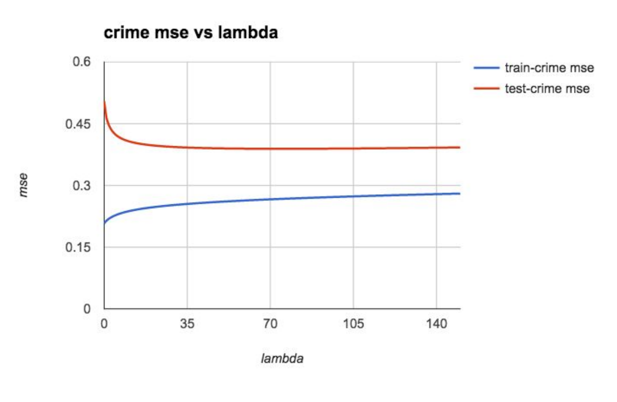


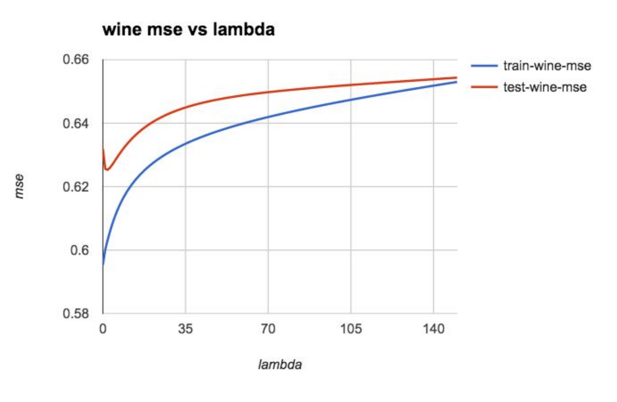












- **Why can't the training set be used to select λ?**

  From the graphs, we can conclude that the mse lines for training sets always go up, and there are no local minimum values for training sets. We can not find a reasonable optimal lambda by using training sets. Another reason is that we can always modify model to fit the training data perfect, which usually causes over-fitting problem. Therefore, we cannot use training set to select lambda.

- **How does λ affect error on the test set?**

  As we increase lambda, the error for test set drops down to a local minimum value first and then start going up. And at the local minimum value, we get the optimal solution for lambda.

- **How does the choice of the optimal λ vary with the number of features and number of examples?**

Test dataset	Optimal lambda	mse
test-100-10	8	4.159678509482881
test-100-100	22	5.0782998005938245
test-50-1000-100	8	5.540902229185348

test-100-1000-100	19	5.205911957333234
test-150-1000-100	23	4.848943053347792
test-1000-100	27	4.315570630318465
test-crime	75	0.38902338771344336
test-wine	2	0.6253088423047056

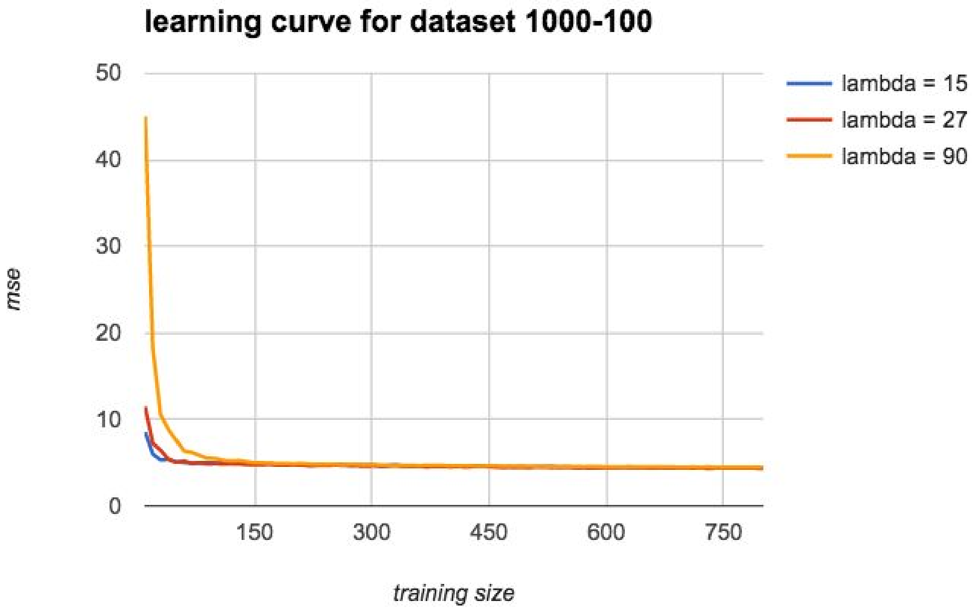
According to results I got, by comparing the results for test-100-10 and test-100-100, we can conclude that when we have fixed number of examples, if we have more features in our data, we have a larger optimal lambda. By comparing the results for test-50-1000-100, test-100-1000-100, test-150-1000-100 and test-1000-100, we can say that when we have fixed number of features, if we have more examples, the optimal lambda gets larger.

- **Consider both the cases where the number of features is fixed and where the number of examples is fixed. How do you explain these variations?**

  When we have fixed number of features and we increase the number of examples, it could cause over-fitting problem as we have more data records to be trained. Using a larger regularization coefficient lambda could make a larger impact on shrinking these effects on parameter w to zero, which implies larger lambda could help address the over-fitting issue in this case. When we have fixed number of examples and we increase the number of features, we have the same situation. The larger optimal lambda could handle the over-fitting problem caused by this change.

- **Task 2 Learning Curves**

  I use lambda = 15, 27, 90 for this task.



  - **What can you observe from the plots regarding the dependence on λ and the number of samples?**

    Based on the plots, we can conclude that when the number of samples increases from 10 to 150, we have higher mse for larger lambda. But as the number of samples increase to more than 200, we have almost the same mse for these three different lambdas. The dependence on lambda gets weaker as we have more samples.

  - **Consider both the case of small training set sizes and large training set sizes. How do you explain these variations?**

    Since we have fixed number of features, when we have smaller training set sizes, a larger lambda would have a larger and more obvious impact on parameter w. Therefore, we could get a larger mse. However, as the training size gets larger and larger, this impact by lambda could be eliminated.

- **Task 3 Cross Validation**

Optimal solutions from task 1

dataset	lambda	mse
100-10	8	4.159678509482881
100-100	22	5.0782998005938245
50-1000-100	8	5.540902229185348
100-1000-100	19	5.205911957333234
150-1000-100	23	4.848943053347792
1000-100	27	4.315570630318465
crime	75	0.38902338771344336
wine	2	0.6253088423047056

Results from cross validation

dataset	lambda	mse
100-10	12	4.175709159671147
100-100	20	5.08088881791838
50-1000-100	24	5.934465384186084
100-1000-100	30	5.259982741015501
150-1000-100	46	4.934192607760081
1000-100	39	4.322722350882442
crime	150	0.39233899203438105
wine	2	0.6253088423047056

- ○ **How do the results compare to the best test-set results from part 1 both in terms of the choice of λ and test set MSE?**

In general, cross validation method usually gives us a larger optimal lambda for all test datasets except 100-100 and wine test datasets. For 100-100, cross validation gives a slightly smaller lambda and for wine, it gives the same lambda.

In term of MSE, MSEs at optimal lambdas given by cross validation are always larger than MSEs from task 1 except for wine dataset, they two give the same result.

- **What is the run time cost of this scheme?**

  According to the algorithm of cross validation, suppose the time for each validation is k, in our case, we have to repeat it 10 times for each lambda. Therefore the time cost for each lambda is 10k, and we have 151 lambda values, [0, 150]. The total time for each one dataset is 1510k. For 8 datasets, it would be 12080k. In general, if we have n folds in enumerated rage from 1 to m, the run time would be (mnk). And it usually takes 30 to 60 seconds to finish all calculations for our task.

- **How does the quality depend on the number of examples and features?**

  By comparing the results for test-100-10 and test-100-100, we can conclude that when we have fixed number of examples, if we have more features in our data, we have a larger optimal lambda and larger mse. By comparing the results for test-50-1000-100, test-100-1000-100, test-150-1000-100, we can say that when we have fixed number of features, if we have more examples, the optimal lambda gets larger but with smaller mse. However, if we compare test-150-1000-100 with test-1000-100, we have a smaller lambda and smaller mse. And with more examples and more features, we have a longer run time for finishing this algorithm.

- **Task 4 Bayesian Model Selection**

Optimal solutions from task 1

dataset	mse
100-10	4.159678509482881
100-100	5.0782998005938245
50-1000-100	5.540902229185348
100-1000-100	5.205911957333234
150-1000-100	4.848943053347792
1000-100	4.315570630318465
crime	0.38902338771344336
wine	0.6253088423047056

Results from Bayesian Model Selection

dataset	mse
100-10	4.180101581922186
100-100	7.352631980774663
50-1000-100	5.78957529374873
100-1000-100	5.733930734524779
150-1000-100	5.242748175555804
1000-100	4.338351518207383
crime	0.3911022080620118
wine	0.626745817029432

- **How do the results compare to the best test-set results from part 1 in terms of test set MSE?**

  In general, for all test datasets, Bayesian Model Selection has larger MSEs than optimal solutions from Task 1.

- **What is the run time cost of this scheme?**

  The run time for Bayesian Model Selection is much faster than using cross validation. For my case, I set the initial value of alpha and beta to 1 and the convergence error to 10^-4. The algorithm usually terminates within 15 iterations.

- **How does the quality depend on the number of examples and features?**

  By comparing the results for test-100-10 and test-100-100, we can conclude that when we have fixed number of examples, if we have more features in our data, we have a larger mse. By comparing the results for test-50-1000-100, test-100-1000-100, test-150-1000-100 and test-1000-100, we can say that when we have fixed number of features , if we have more number of examples, we have a smaller mse.

- **Task 5 Comparison**

Results from cross validation

dataset	mse
100-10	4.175709159671147
100-100	5.08088881791838
50-1000-100	5.934465384186084
100-1000-100	5.259982741015501
150-1000-100	4.934192607760081
1000-100	4.322722350882442
crime	0.39233899203438105
wine	0.6253088423047056

Results from Bayesian Model Selection

dataset	mse
100-10	4.180101581922186
100-100	7.352631980774663
50-1000-100	5.78957529374873
100-1000-100	5.733930734524779
150-1000-100	5.242748175555804
1000-100	4.338351518207383
crime	0.3911022080620118
wine	0.626745817029432

- **How do the two model selection methods compare in terms of the test set MSE and in terms of run time?**

  In term of the test set MSE, by comparing the results for 50-1000-100, we can conclude that when we are given training datasets that have ratio of the number of examples to the number of features less than 1, i.e. the number of examples is less than the number of features. We should use Bayesian Model Selection, since it has a smaller mse. There could be another explain for this case. When we use cross validation algorithm, we always have to sacrifice a part of our training test as test data. This is really harmful when we have small number of training data. Therefore, for such case, we should consider to use Bayesian Model Selection algorithm. For other cases where the ratio of the number of examples to the number of features is equal or greater than 1, we should use cross validation which gives us smaller MSEs. From my experiments, in general, in order to have a smaller mse, we should pick cross validation.

In term of run time, based on my experiments, Bayesian Model Selection runs much faster in almost all cases. If we care more about the run time, we should always use Bayesian Model Selection.

○ **What are the important factors affecting performance for each method?**

In general, the most important factors affecting performance for these two methods are the number of examples and the number of features.

As for cross validation method, the most important factor affecting its performance is the number of examples. When we compare test datasets 50-1000-100, 100-1000-100, 150-1000-100 and 1000-100 with fixed number of features, we can conclude that MSEs for these datasets are really sensitive to the change of the number of examples. Especially when we increase the number of examples from 50 to 100. The mse changes about 0.7. In term of the run time, when we run cross validation for 1000-100, it would take a much longer time than 50-1000-100, 100-1000-100 and 150-1000-100.

As for Bayesian Model Selection, from the results we got and from the equations to implement the algorithm, we could say that the number of features has more effect on its performance. From the equations for its algorithm, we can notice that the updated functions for beta and alpha are highly dependent on the number of features. The effect reflects on the term m_n in these equations. And from the results we got from our experiments, comparing the results for 100-10 and 100-100, we can conclude, when we have fixed number of examples, Bayesian Model Selection would be really sensitive to the change of the number of features. As we increase the number of features from 10 to 100, the MSE changes around 3.2. In the term of run time, the performance for this algorithm is also highly affected by the number of features. Based on my experiments, it takes 5 iterations to finish the work for 100-10 dataset, and it takes 12 iterations for 100-100. As we fixed the number of examples, if we increase the number of features, the run time for Bayesian Model Selection would be largely increased. When we fixed the number of features, if we increase the number of examples, the run time does not change a lot. In my experiments, it takes 8 iterations to finish the work for 50-1000-100, 6 iterations for 100-1000-100, 5 iterations for 150-1000-100 and 4 iterations for 1000-100. Based on this data, we can say that when we have fixed number of features, the run time for Bayesian Model Selection does not vary to much for increased number of examples.

○ **Given these factors, what general conclusions can you make about deciding which model selection method to use?**

If we have fixed number of examples where the number is not so small and we have varied number of features and we do not really care about the run-time cost, we should choose cross validation method for getting a smaller mse.

If we have fixed number of features and we have varied number of examples, probably we should use Bayesian Model Selection method. Especially, when we have datasets that the ratio of the number of examples to the number of features is less than 1 or the number of available training data is relatively small, we should consider to use Bayesian Model Selection. And for the case that we have a really large amount of data, such as we have more than 1000 data records, if the run-time cost is our first concern, we should always use Bayesian Model Selection method. Based on the performance we got for 1000-100, wine and crime from these two methods, they two provide really closed MSEs. So in general, when we work on large dataset (more than 1000 records), we should firstly consider to use Bayesian Model Selection algorithm, as it offers a much faster run time.