Carl Glahn

SID: 998092557

ECS 171

Report for HW 3

Notes for code:

I put this in the code as well but credit for all of the m files in the MSVM pack are not mine however they are public and the authors asked to be cited thusly:

*F. Lauer and Y. Guermeur, MSVMpack: a Multi-Class Support Vector Machine Package,*
*Journal of Machine Learning Research,*
*12:2269-2272, 2011.*

To run the code you might need to change the addpath() function although it should work as written with the "addpath(genpath('MSVMpack1.5/matlab'))"

Also I had to do some work arounds to get the ROC and PR curves with the toolbox I chose, the deatails of which are further explained in #4

1. For this problem I decided to use ridge regression, which penalizes the objective function (RSS) by adding the sum of the weights squared multiplied by a weight constrained parameter (lambda) while also bounding the sum of the squared weights to be below a constant c which I chose to be one. What this does is that it creates a radius in which the weights are allowed to take values and the features that have the most effect on the outputs will take most of total value of the sum of squared weights and the other feature's weights will drop to around zero.

   To find my optimal lambda I set it to a predefined value (I started at one), generated my weights, and then use the error to adjust lambda. What I found was that lambda settled at a value of .15, where the error would bounce from positive and negative 7.0897e-06. To save time on the program I preset lambda to .15, although my program would alter lambda if the error was big enough (although after printing out some values I found this never occurred).

   Because the shape of ridge regression is circular, many of the weights never quite drop to 0 and instead settle around .001. To fix this I set a threshold where if the weight was below .001, it would be treated as if it the weight was a zero. Given this the (lowest) number of non-zero features (NNZ) in the cross validation was 2329.

   The table for the 10 fold cross validation errors are below and the generalization error (the average) was .024994

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Error | 0.012782 | 0.049608 | 0.014803 | 0.043084 | 0.003046 | 0.025071 | 0.01218 | 0.003081 | 0.068619 | 0.017663 |
| NNZ elements | 2505 | 2614 | 2553 | 2619 | 2629 | 2592 | 2453 | 2566 | 2329 | 2553 |

2. For this problem I took 100 random samplings of the original dataset (bootstraps) and ran them through my cross validation linear regression algorithm. While doing this I stored all of the weights that were generated for each bootstrap data set. After I had generate rated 100 sets of weights I then ran a sample of the original dataset through every set of weights and store the outputted predictions in a vector. I would then take the mean of the vector and use it to generate the upper and lower bounds (for a 95% confidence interval) for that sample. I would then repeat this process for each sample of the dataset. To do this I had to assume that the values of the predictions generated by the set of weights of the bootstrap dataset had a normal distribution. The results are below

| Sample Number | Lower Bound | Upper Bound |
|---|---|---|
| 1 | 0.623955276 | 0.629381937 |
| 2 | 0.636971178 | 0.64168564 |
| 3 | 0.628498858 | 0.632744783 |
| 4 | 0.725743885 | 0.728205221 |
| 5 | 0.715667235 | 0.71966343 |
| 6 | 0.723075155 | 0.725367968 |
| 7 | 0.465756879 | 0.467394772 |
| 8 | 0.467521634 | 0.469468582 |
| 9 | 0.465532523 | 0.468148572 |
| 10 | 0.477974076 | 0.479421874 |
| 11 | 0.481011655 | 0.482419862 |
| 12 | 0.477667248 | 0.479668891 |
| 13 | 0.470306588 | 0.477712149 |
| 14 | 0.470306588 | 0.477712149 |
| 15 | 0.461478801 | 0.483432247 |
| 16 | 0.462230949 | 0.479513541 |
| 17 | 0.479958529 | 0.490165307 |
| 18 | 0.464428968 | 0.479674003 |
| 19 | 0.470899709 | 0.483029836 |
| 20 | 0.47515208 | 0.486222081 |
| 21 | 0.380244577 | 0.432207253 |
| 22 | 0.681166107 | 0.791083669 |
| 23 | 0.491731483 | 0.499281829 |
| 24 | 0.314777571 | 0.40524363 |
| 25 | 0.381527591 | 0.387037198 |
| 26 | 0.376139749 | 0.381199142 |
| 27 | 0.381938634 | 0.388340175 |
| 28 | 0.406091451 | 0.415754596 |
| 29 | 0.416013544 | 0.423903978 |
| 30 | 0.420306464 | 0.429216426 |

| | | |
|---|---|---|
| 31 | 0.477818991 | 0.480758979 |
| 32 | 0.475866602 | 0.479116793 |
| 33 | 0.479274017 | 0.484096546 |
| 34 | 0.487570637 | 0.490538935 |
| 35 | 0.489471022 | 0.49445035 |
| 36 | 0.480557024 | 0.483592335 |
| 37 | 0.63155082 | 0.634559284 |
| 38 | 0.623774869 | 0.629578995 |
| 39 | 0.63095054 | 0.633855536 |
| 40 | 0.635499167 | 0.640162751 |
| 41 | 0.634655283 | 0.641379787 |
| 42 | 0.587531729 | 0.605470481 |
| 43 | 0.678004018 | 0.682241779 |
| 44 | 0.647292894 | 0.66881909 |
| 45 | 0.715262552 | 0.720814335 |
| 46 | 0.708279596 | 0.711745101 |
| 47 | 0.688873092 | 0.697052302 |
| 48 | 0.2190242 | 0.262315114 |
| 49 | 0.211931577 | 0.275076856 |
| 50 | 0.296984594 | 0.306159353 |
| 51 | 0.252727132 | 0.272825354 |
| 52 | 0.444332975 | 0.460132579 |
| 53 | 0.440594997 | 0.456560808 |
| 54 | 0.444635472 | 0.46131852 |
| 55 | 0.45034981 | 0.536307209 |
| 56 | 0.1118334 | 0.125807637 |
| 57 | 0.112826833 | 0.126884945 |
| 58 | 0.243258286 | 0.249001839 |
| 59 | 0.354575121 | 0.359800445 |
| 60 | 0.231637856 | 0.236726769 |
| 61 | 0.233449869 | 0.238727263 |
| 62 | 0.23125939 | 0.236416996 |
| 63 | 0.265059869 | 0.30562457 |
| 64 | 0.253544477 | 0.29273402 |
| 65 | 0.239069469 | 0.271184966 |
| 66 | 0.218778887 | 0.245718868 |
| 67 | 0.374779148 | 0.38173773 |
| 68 | 0.371136309 | 0.377050458 |
| 69 | 0.380740858 | 0.391625325 |
| 70 | 0.369513055 | 0.372433518 |
| 71 | 0.361576299 | 0.365623804 |

| | | |
|---|---|---|
| 72 | 0.366221606 | 0.369587031 |
| 73 | 0.373691002 | 0.377128749 |
| 74 | 0.368716902 | 0.371021683 |
| 75 | 0.369641223 | 0.372292366 |
| 76 | 0.210759875 | 0.21965342 |
| 77 | 0.239704451 | 0.247966025 |
| 78 | 0.231496912 | 0.23739093 |
| 79 | 0.221710993 | 0.227780187 |
| 80 | 0.24102472 | 0.24954876 |
| 81 | 0.246792697 | 0.258681361 |
| 82 | 0.120079633 | 0.137064388 |
| 83 | 0.131592288 | 0.145463073 |
| 84 | 0.157610525 | 0.161361202 |
| 85 | 0.095454221 | 0.116910744 |
| 86 | 0.129296772 | 0.138701447 |
| 87 | 0.100876305 | 0.116939673 |
| 88 | 0.117813094 | 0.127180649 |
| 89 | 0.109397834 | 0.115045047 |
| 90 | 0.085554879 | 0.097420064 |
| 91 | 0.090377759 | 0.10025225 |
| 92 | 0.096246111 | 0.110590907 |
| 93 | 0.083155836 | 0.089873902 |
| 94 | 0.087950965 | 0.101810994 |
| 95 | 1.215212277 | 1.236977803 |
| 96 | 1.157085236 | 1.18945105 |
| 97 | 1.160223625 | 1.201299294 |
| 98 | 1.215751999 | 1.230546365 |
| 99 | 1.23914671 | 1.262265263 |
| 100 | 0.64335863 | 0.649968401 |
| 101 | 0.619308159 | 0.63437775 |
| 102 | 0.634579485 | 0.645957108 |
| 103 | 0.647742925 | 0.655235096 |
| 104 | 0.617699643 | 0.634967292 |
| 105 | 0.640755197 | 0.649841165 |
| 106 | 0.240817178 | 0.266748358 |
| 107 | 0.218186046 | 0.254739047 |
| 108 | 0.272672534 | 0.289193668 |
| 109 | 0.151343548 | 0.162326111 |
| 110 | 0.176853198 | 0.193447736 |
| 111 | 0.182782195 | 0.202544556 |
| 112 | 0.351274982 | 0.417796756 |

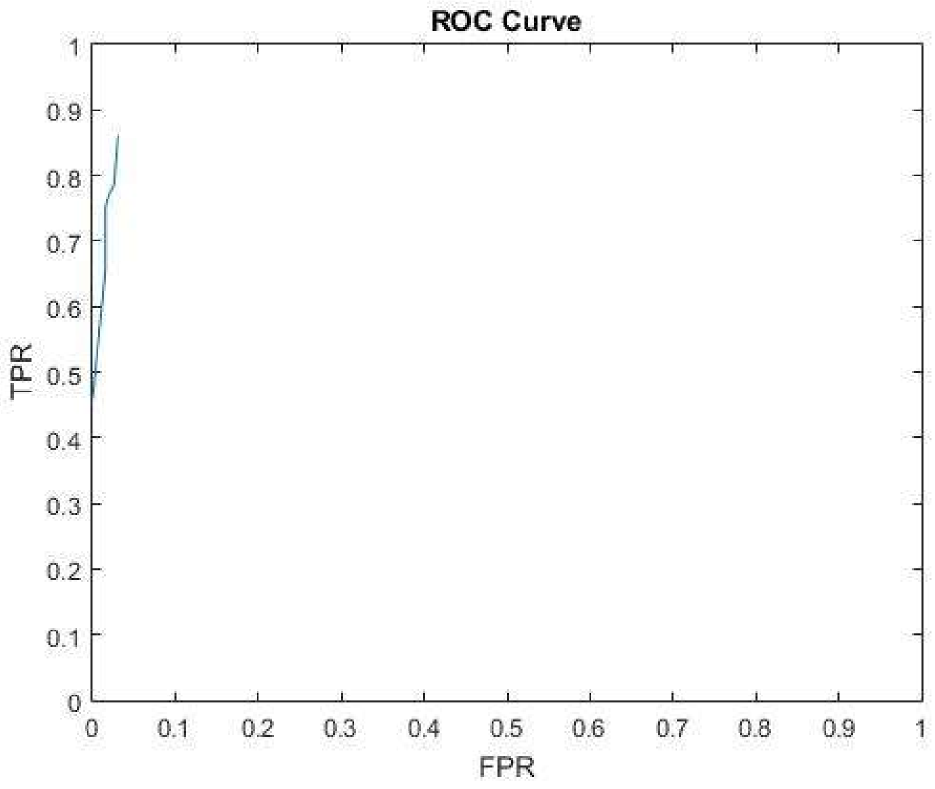
| | | |
|---|---|---|
| 113 | 0.317938463 | 0.350240242 |
| 114 | 0.292816976 | 0.301218806 |
| 115 | 0.182306049 | 0.229557043 |
| 116 | -0.068157468 | 0.027389205 |
| 117 | 0.162212971 | 0.171588564 |
| 118 | 0.107648726 | 0.112309823 |
| 119 | 0.113333898 | 0.124783609 |
| 120 | 0.135874222 | 0.171196813 |
| 121 | 0.104311494 | 0.107889757 |
| 122 | 0.105917518 | 0.110522682 |
| 123 | 0.187449019 | 0.246195696 |
| 124 | 0.104368334 | 0.107665896 |
| 125 | 0.106020467 | 0.109795215 |
| 126 | 0.10370182 | 0.106892606 |
| 127 | 0.103133055 | 0.106590846 |
| 128 | 0.103707606 | 0.106719067 |
| 129 | 0.158811266 | 0.201600121 |
| 130 | 0.534491859 | 0.567119925 |
| 131 | 0.632833241 | 0.65932389 |
| 132 | 0.613046119 | 0.640118763 |
| 133 | 0.54855408 | 0.573234234 |
| 134 | 0.582699816 | 0.600321666 |
| 135 | 0.596987731 | 0.618173833 |
| 136 | 0.507854456 | 0.550246986 |
| 137 | 0.611013174 | 0.648124816 |
| 138 | 0.538703576 | 0.563752818 |
| 139 | 0.545889215 | 0.569846065 |
| 140 | 0.135946901 | 0.165844446 |
| 141 | 0.571156447 | 0.584732957 |
| 142 | 0.727102716 | 0.769982316 |
| 143 | 0.408004279 | 0.414042333 |
| 144 | 0.400126204 | 0.402088922 |
| 145 | 0.398024959 | 0.399356561 |
| 146 | 0.398534517 | 0.400018407 |
| 147 | 0.394412007 | 0.396662323 |
| 148 | 0.400949959 | 0.403632165 |
| 149 | 0.397727181 | 0.398891167 |
| 150 | 0.394292244 | 0.397256276 |
| 151 | 0.401295867 | 0.402511679 |
| 152 | 0.40000133 | 0.40119648 |
| 153 | 0.39601449 | 0.397791979 |

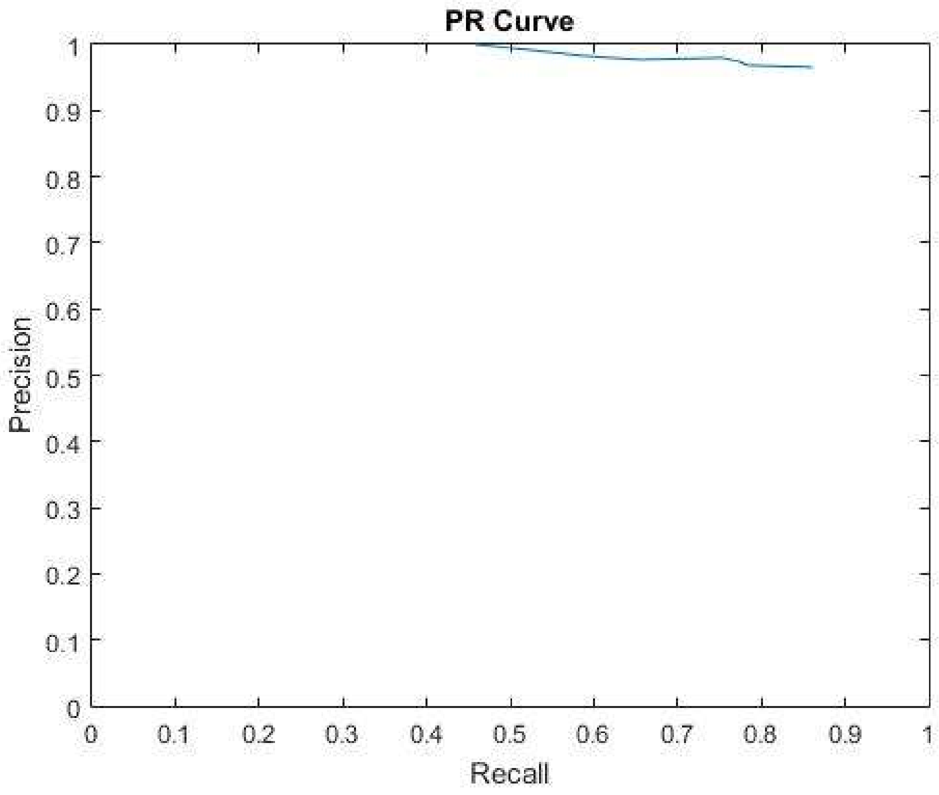
| | | |
|---|---|---|
| 154 | 0.400099999 | 0.40131648 |
| 155 | 0.398107438 | 0.400864515 |
| 156 | 0.392382409 | 0.396570238 |
| 157 | 0.408255173 | 0.414504982 |
| 158 | 0.098002824 | 0.135376061 |
| 159 | 0.646192567 | 0.670142118 |
| 160 | 0.584676244 | 0.599949722 |
| 161 | 0.588647453 | 0.602649949 |
| 162 | 0.29441013 | 0.32282675 |
| 163 | 0.314142594 | 0.343057924 |
| 164 | 0.338454977 | 0.377563963 |
| 165 | 0.28734361 | 0.317416691 |
| 166 | 0.172445885 | 0.242443613 |
| 167 | 0.288728871 | 0.312423502 |
| 168 | 0.225920284 | 0.30654822 |
| 169 | 0.335029002 | 0.371738735 |
| 170 | 0.324746942 | 0.344119793 |
| 171 | 0.31753668 | 0.340935633 |
| 172 | 0.557247555 | 0.578039983 |
| 173 | 0.337648096 | 0.369646298 |
| 174 | 0.294534286 | 0.310532221 |
| 175 | 0.561286448 | 0.5783625 |
| 176 | 0.537100752 | 0.568001016 |
| 177 | 0.516186094 | 0.552625651 |
| 178 | 0.514065573 | 0.52834214 |
| 179 | 0.459054825 | 0.477763971 |
| 180 | 0.197052103 | 0.198813439 |
| 181 | 0.200173278 | 0.201165134 |
| 182 | 0.200820248 | 0.202305287 |
| 183 | 0.197230249 | 0.203833356 |
| 184 | 0.200737671 | 0.207506921 |
| 185 | 0.197591813 | 0.200149403 |
| 186 | 0.199263676 | 0.200618192 |
| 187 | 0.167048593 | 0.180510259 |
| 188 | 0.201529499 | 0.203328722 |
| 189 | 0.196887699 | 0.205843311 |
| 190 | 0.19585881 | 0.206644513 |
| 191 | 0.192251777 | 0.196566788 |
| 192 | 0.202226646 | 0.204326338 |
| 193 | 0.199392958 | 0.200220997 |
| 194 | 0.198192502 | 0.199553888 |

3. For this problem I trained on the whole dataset to get the most accurate result. After generating the weights and taking the mean of every column of the input, the predicted growth rate outputted was .3936.
4. For this problem I used 4 muti-classification SVMs (one for each classification categories). To eliminate features I trained using ridge regression on the whole data set and picked the features that had a weight coefficient of greater than .001 of which there were 1561. I graphed the ROC curves in the preferred method using 10 points for confidence thresholds of .5, .55,…,.95 as recommended. However there was one change and one adjustment I made to this method. The change was that there was one point for the test run for each threshold, meaning that I took the generated labels for a particular threshold (after the whole cross validation was done) and calculated one value for FPR, TPR, recall, and precision. The adjustment I had to make came from the fact that the toolbox I was using did not give me the option of adjusting a threshold. However it did give me the confidence ratings of each class of prediction by outputting Mathew's coefficients (one set for each fold of the cross validation) which essentially measure correlation on a scale from -1 to 1. Using these coefficients I could mimic having thresholds by referring to the coefficients to see how much of a correlation there was for that class. I set the initial benchmark for a .5 threshold at a .5 (half) correlation. This process was further complicated because the toolbox only outputted the coefficients to the console and not in a vector, meaning that I was forced to manually copy the coefficients over to a text file and then read them in from there. This being said if you just run my code without manually copying the coefficients from the console to the text file (they change every time because of the random ordering) you will get a very high error because the old coefficients are no longer correct. To get similar results as are reported below you must run the SVMs while commenting out the graphing function, then copy over the coefficients manually, then uncomment the graphing function and run the code again commenting out the SVMs . Obviously this isn't the cleanest way to do things but it did give me the required results. Which are displayed below.

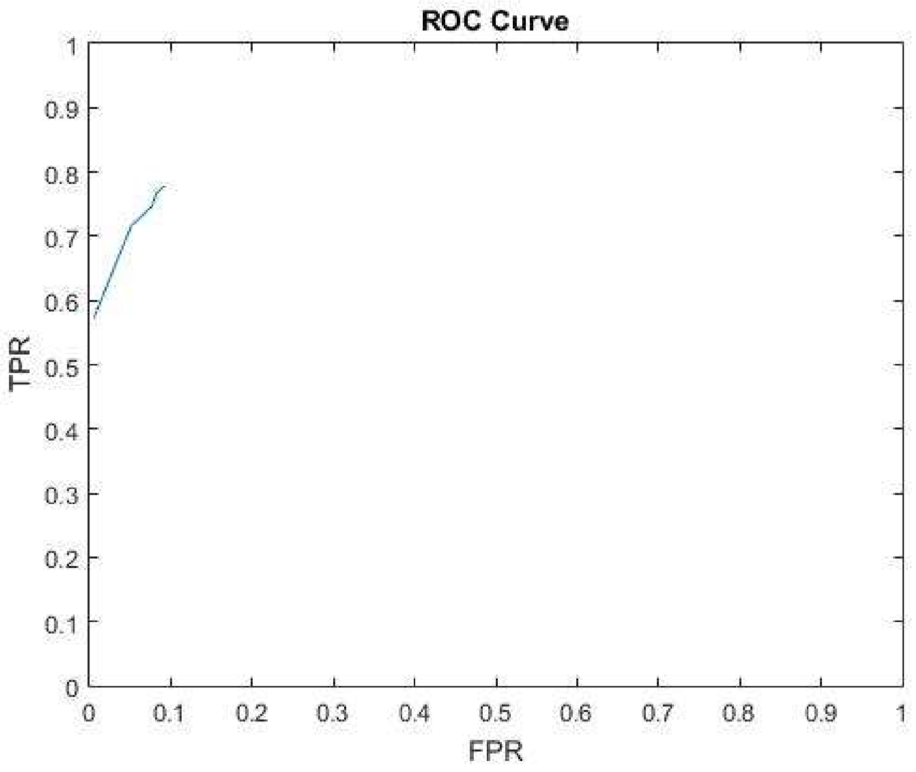
|  | Strain | Medium | Environment | Gene Perturbation |
|---|---|---|---|---|
| AUC | 0.0208 | 0.0607 | 0.0452 | 0.0245 |
| AUPRC | 0.3941 | 0.195 | 0.5428 | 0.5042 |

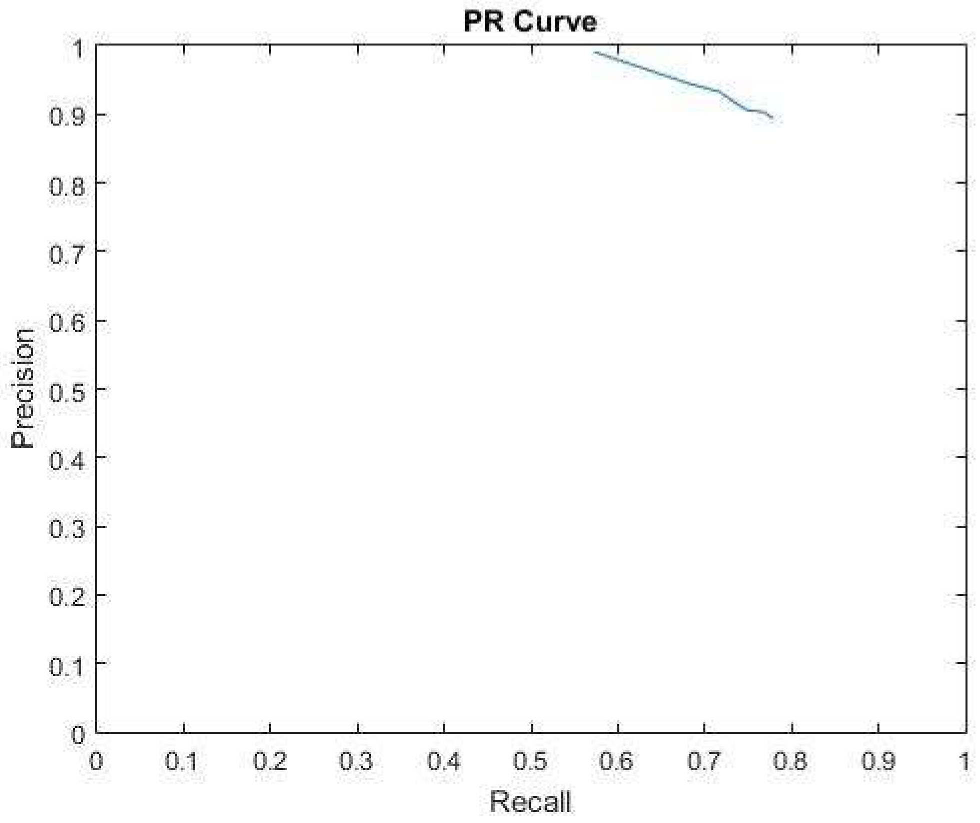
Strain Curves: (Error of .1186)



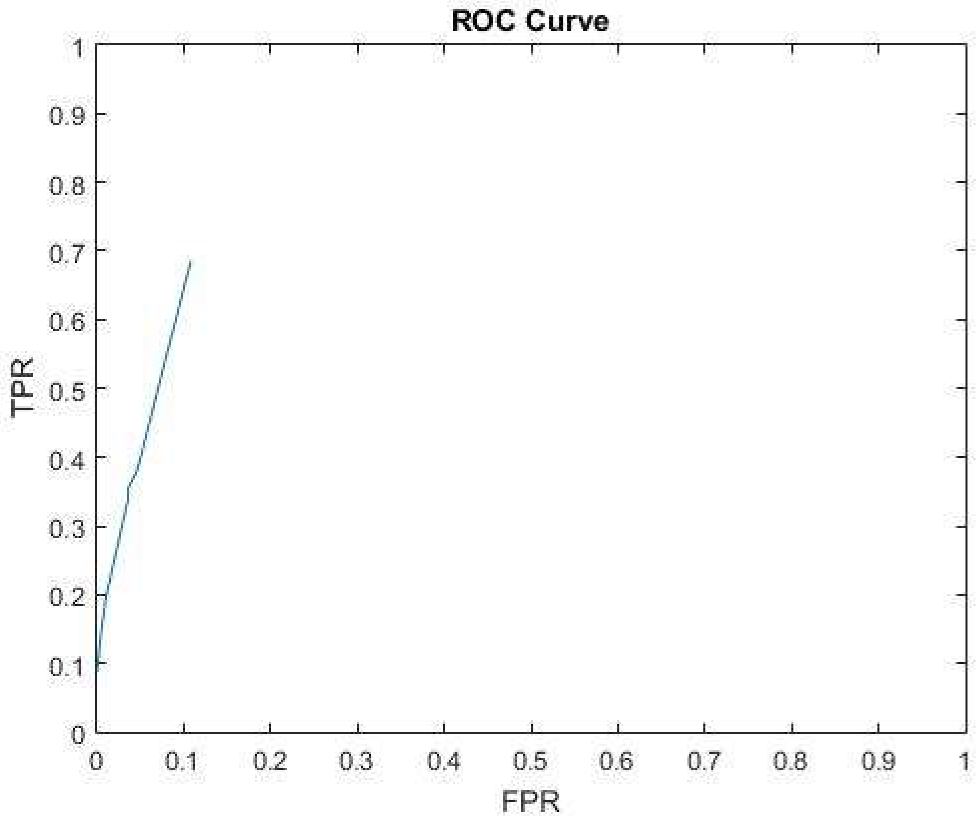


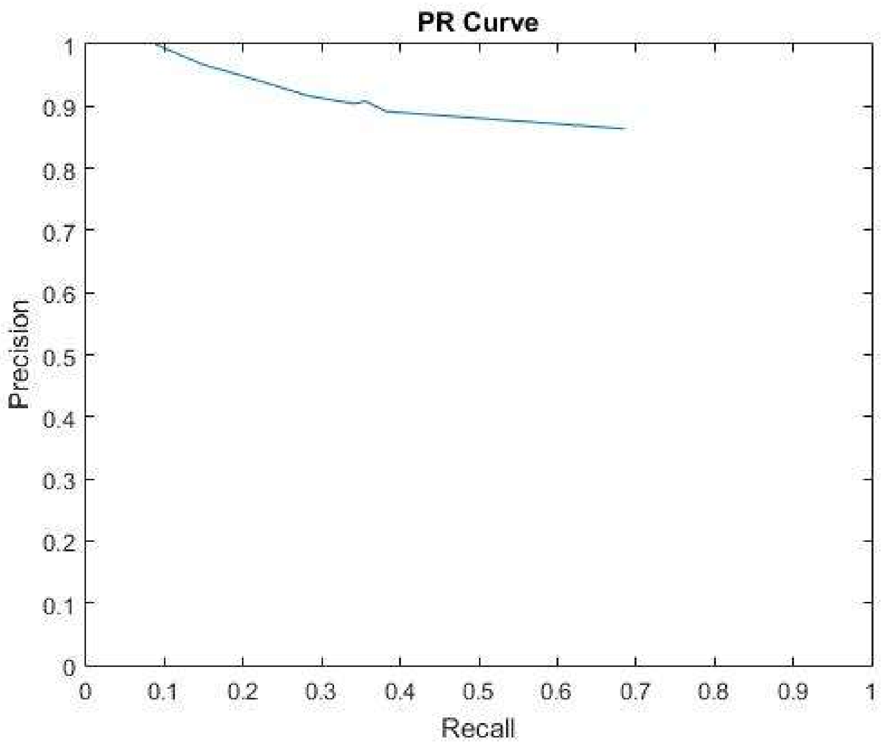
Medium Curves: (Error of .1959)



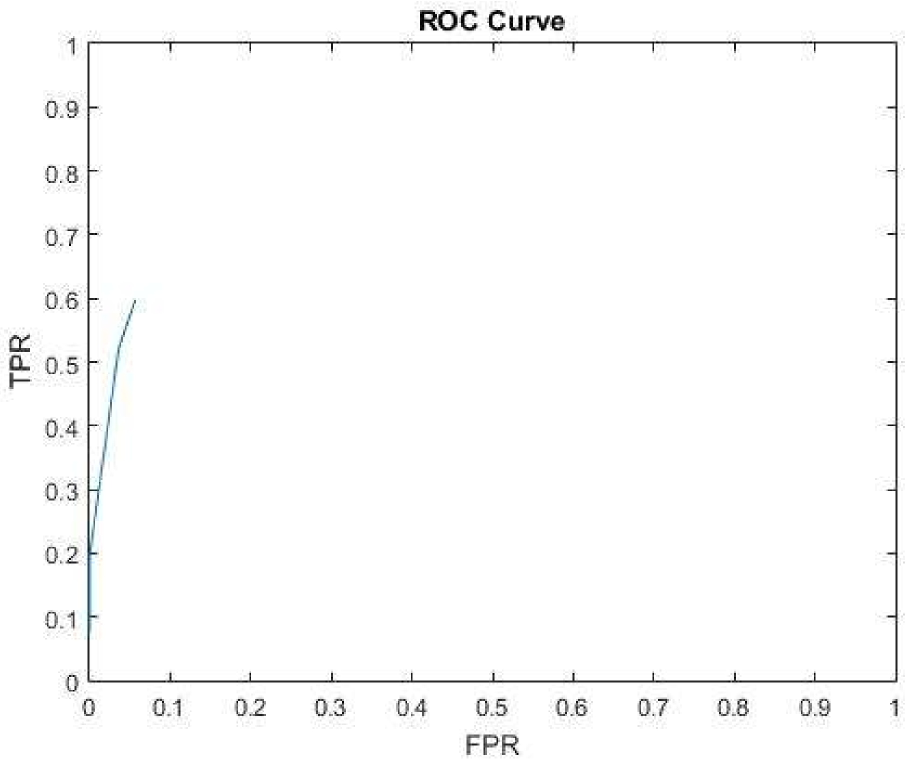


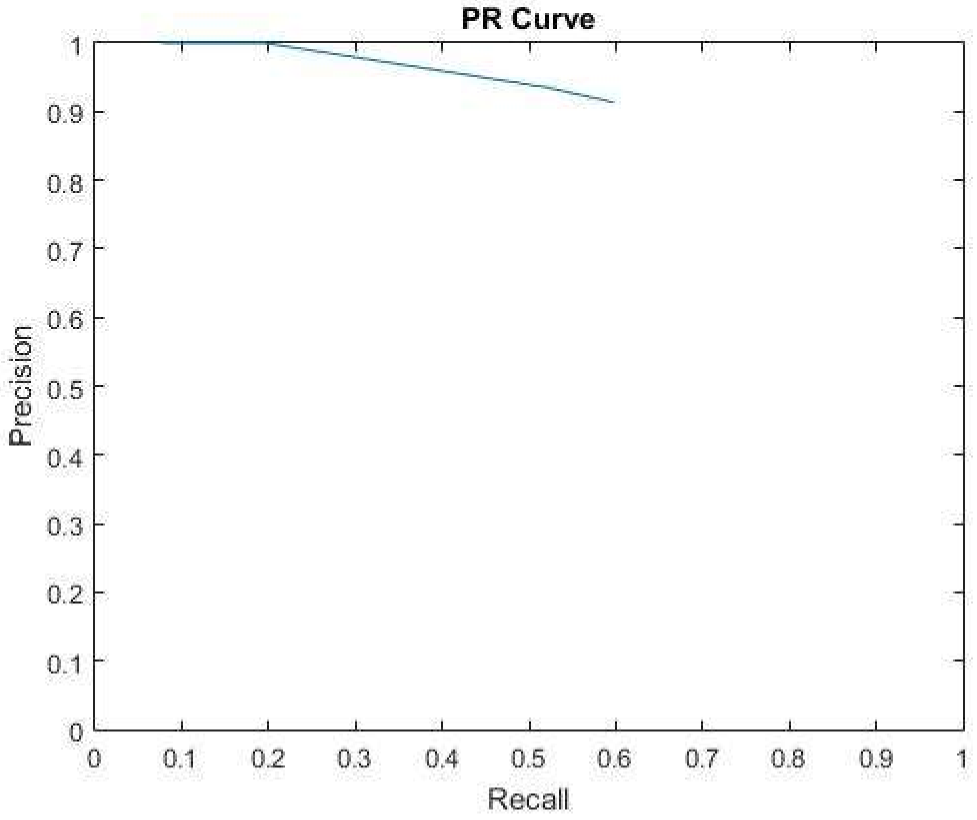
Environment Curves: (Error of .2062)





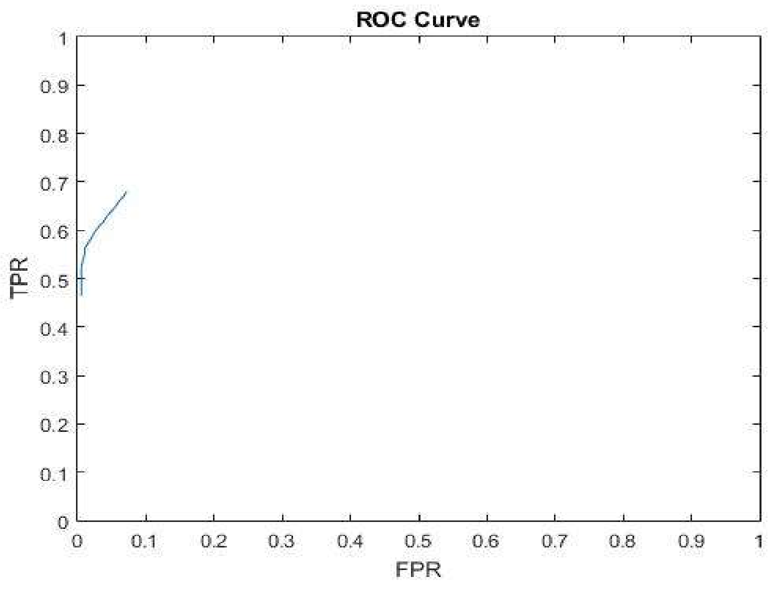
Gene Perturbation Curves: (Error of .1804)

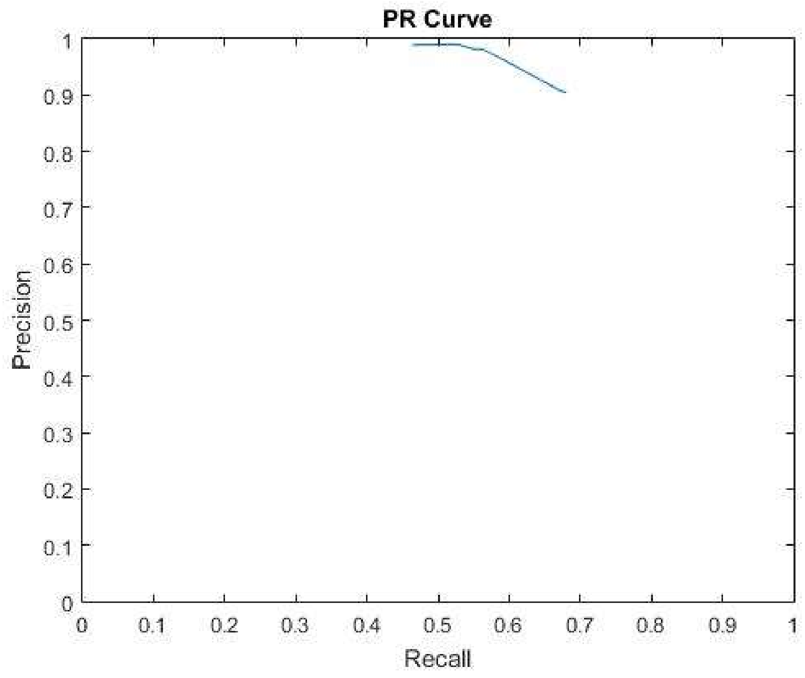




One thing I did notice was that my AUC and AUPRC values seemed very low considering that, for the most part, I had achieved an (approximate) 80% accuracy rating on each SVM.

5. For this problem I found that there was a total of 23 different pairing between the environmental and medium labeling. Naturally I simply encoded these labeling from 1 to 23 and graphed things in the same manner as I did for problem 4.
   Error: .2474
   AUC: .0414
   AUPRC: .2085

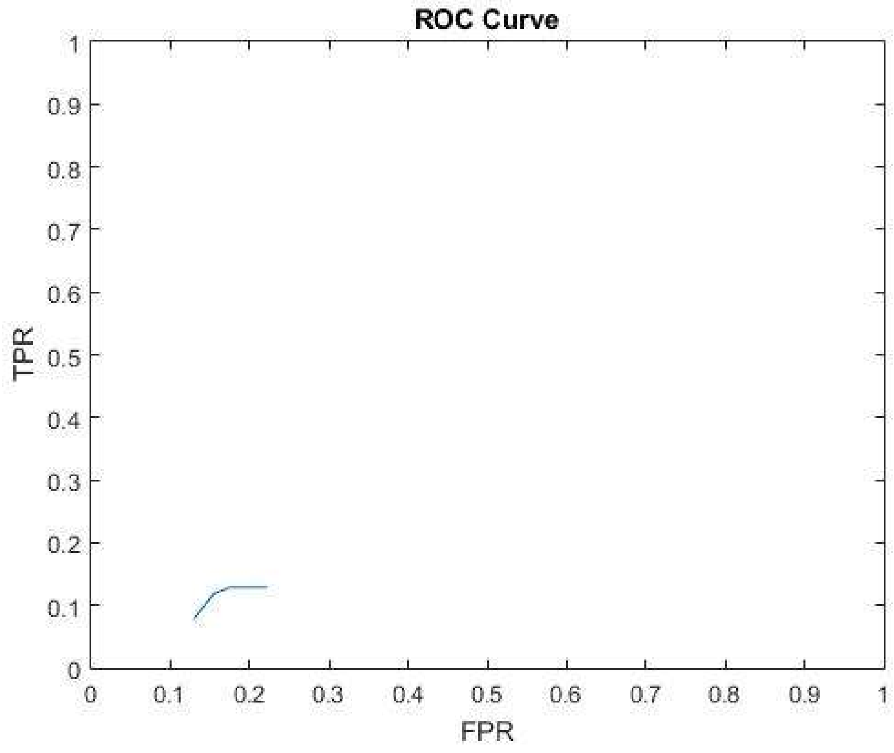


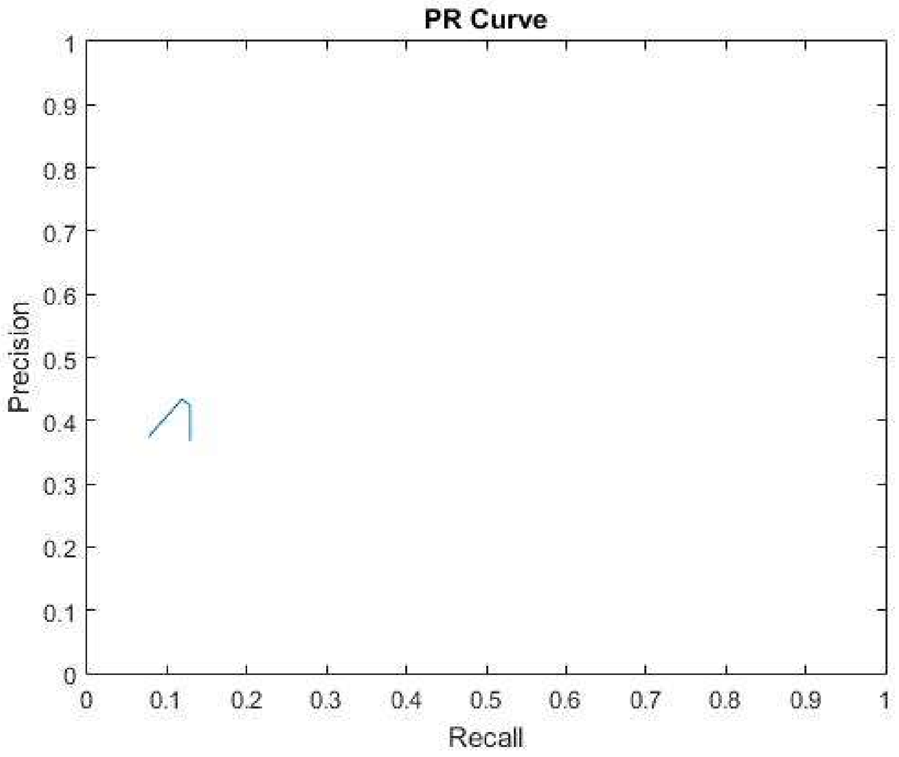


As you can see not only did the composite predictor have a higher error than the 2 separate predictors, it also had a lower AUC value. From this we can conclude that we are in fact better off with 2 predictors rather than one composite one. There were 23 classes in total for the composite predictor so the baseline accuracy would be .04347 (1/23). As before, I was surprised by the low AUC and AUPRC values considering the (approximate) 75% accuracy rating.

6. For this question I did the PCA analysis for the composite SVM as the instructions said to do this for "the SVM classifier". The TA later mentioned that if one had already done this problem in this

manner, that we could keep our results as long as we put a note for it in the report. The results are displayed below.
Error: .7938
AUC: .0111
AUPRC: .0211





After looking at the statistics and figures above it is painfully obvious that the three principle components did not retain nearly enough information about the original 4495 features to produce accurate results (the error margin alone increased be 50%).