Machine Learning
CSCI 5622 Fall 2017
Name: Chi Chen

Homework 4
Due Time Nov 10, 2017
CU identity key: chch4713

# 1 Support Vector Machines (50 pts)

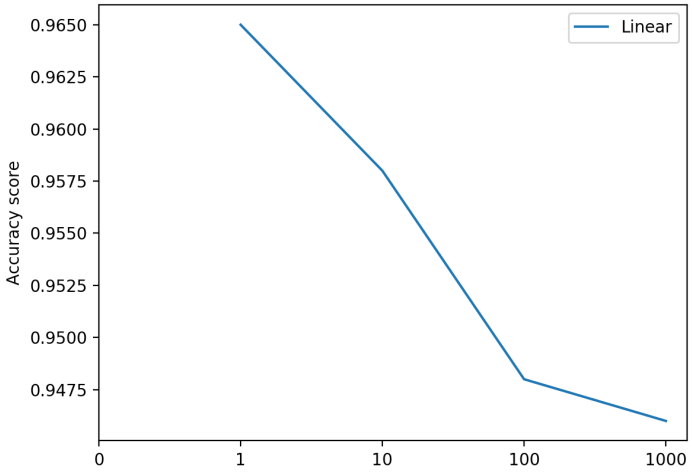
I. Experiment with linear, polynomial, and RBF kernels. In each case, perform a GridSearch to help determine optimal hyperparameters for the given model (e.g. C for linear kernel, C and p for polynomial kernel, and C and $\gamma$ for RBF). Comment on the experiments you ran and optimal hyperparameters you found.

The parameters I use were $\{1, 10, 100, 1000\}$ for C in all models, $\{2, 3, 4, 5, 6\}$ for p in polynomial kernel and $\{0.1, 0.01, 0.001, 0.0001\}$ for $\gamma$ in RBF.

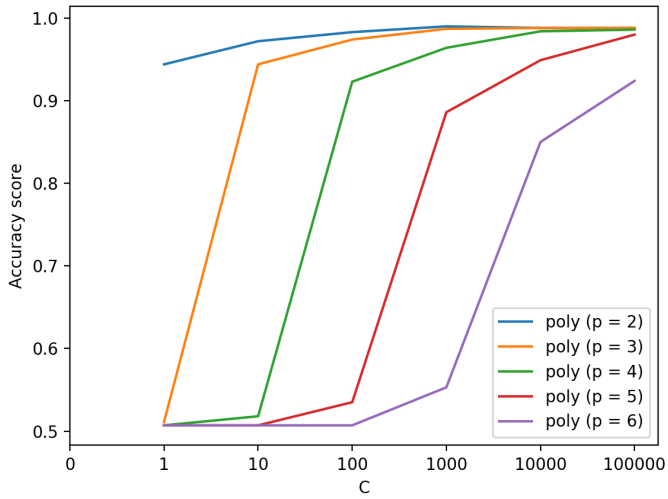
| | C = 1 | C = 10 | C = 100 | C = 1000 |
|---|---|---|---|---|
| Linear | 0.965 | 0.958 | 0.948 | 0.946 |
| Poly (p = 2) | 0.944 | 0.972 | 0.983 | **0.990** |
| Poly (p = 3) | 0.511 | 0.944 | 0.974 | 0.987 |
| Poly (p = 4) | 0.507 | 0.518 | 0.923 | 0.964 |
| Poly (p = 5) | 0.507 | 0.507 | 0.535 | 0.886 |
| Poly (p = 6) | 0.507 | 0.507 | 0.507 | 0.553 |
| RBF ($\gamma$ = 0.1) | 0.972 | 0.973 | 0.973 | 0.973 |
| RBF ($\gamma$ = 0.01) | 0.960 | 0.972 | 0.980 | 0.983 |
| RBF ($\gamma$ = 0.001) | 0.934 | 0.958 | 0.968 | 0.969 |
| RBF ($\gamma$ = 0.0001) | 0.517 | 0.934 | 0.957 | 0.966 |

As you can see, in all the parameter sets above, {'C': 1000, 'degree': 2, 'kernel': 'poly'} was the set that performed best.

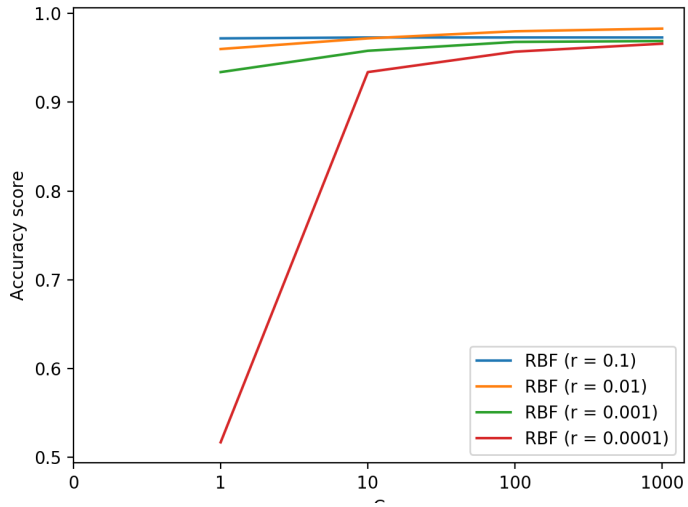
II. Comment on classification performance for each model for optimal parameters by either testing on a hold-out set or performing cross-validation.




In linear models, as you can see, the accuracy score will decrease if C increase.

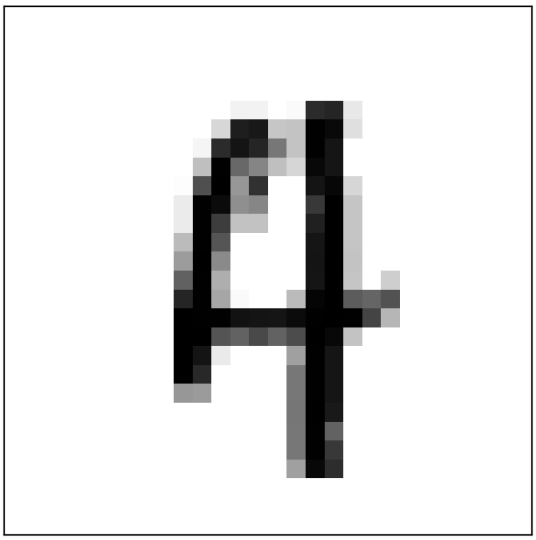


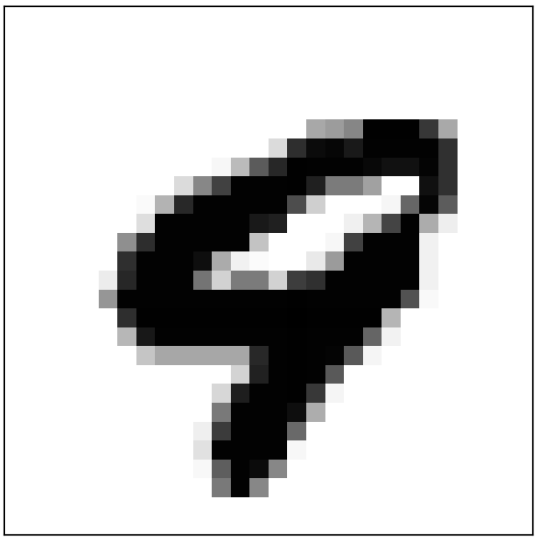

In Polynomial models, as you can see, when C increase, the accuracy score will increase too, but the increasing speed will decrease.




In RBF models, the accuracy score will be almost the same really quickly, when C equals to 100, the results of all will be similar.

III. Give examples (in picture form) of support vectors from each class when using a polynomial kernel.





The left one is one of the support vector for digit 4, and the right one is the one of the support vector for digit 9. The way to get this is using the ”support_” attribute in SVC to get the indices of the support vectors, then draw one of that for both digits.

## 2. Learnability (25 pts)

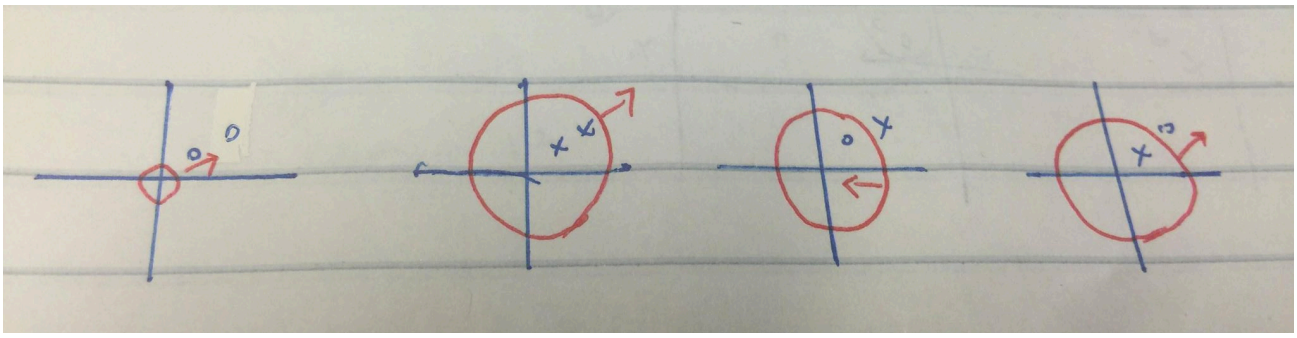
First, we define this question as a finite consistent hypothesis problem, so we will use the formula below.

$$m \geq \frac{1}{\epsilon}(ln|H| + ln\frac{1}{\delta})$$

In this question, we will get a triangle as a consistent training example in each train, and any configuration of three vertices in the space [0, 99] x [0, 99] will be a valid triangle, so we will have 10000C3 triangles. At the same time, since it has finite hypothesis, and getting consistent training example as input, it will be a finite consistent hypothesis problem.

Use 10000C3 as H, 0.15 as error, 0.05 as confidence in the formula above, then we will get m need to larger than 46 to have confidence 95% and accuracy 85%.

# 3. VC Dimension (25 pts)



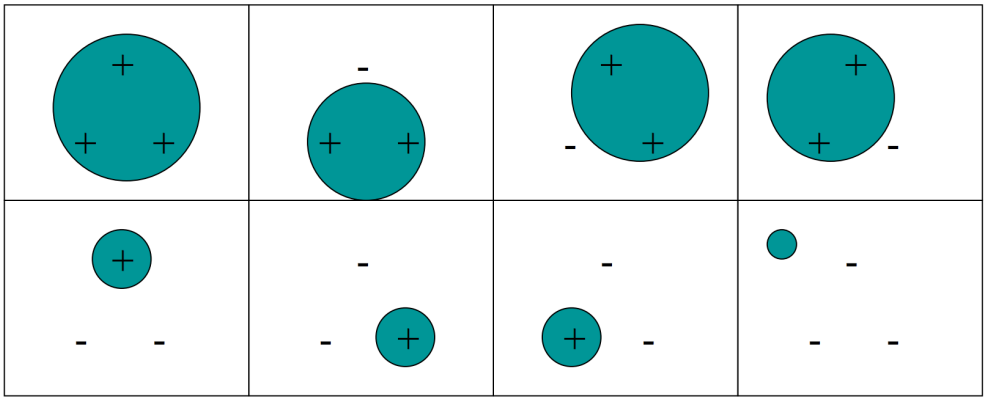
As the figure above, it showed that the Vcdim(H) will be at least 2. Now I need to proof the upper bound of VCdim(H).

For any three point, marked them by the distance form the origin as N(nearest from origin), M(middle distance from origin), F(farthest from origin), so dis(FO) >= dis(MO) >= dis(NO).

Then we assign {+1} to F and N, {-1} to M. For any circle want to split N & M, the radius R should be larger than dis(NO) and less than dis(MO) i.e. dis(MO) > R > dis(NO). And for any circle want to split F & M, the radius R should be larger than dis(MO) and less than dis(FO) i.e. dis(FO) > R > dis(MO).

If we combined all those three rules, we'll find there are no R can be exist, then VCdim(H) should be less than 3. So the VCdim(H) will at least 2 and less than 3, it can only be 2.

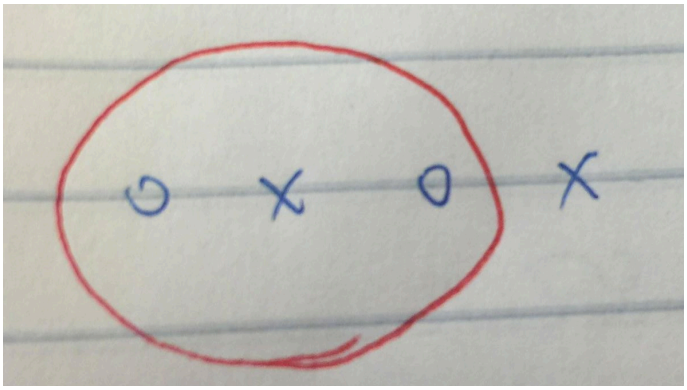
# Extra Credit (10 pts)



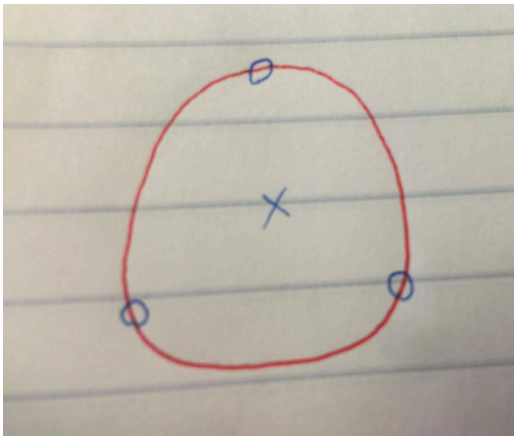
As the figure above, the VCdim(H) will be at least 3. Now I need to proof the upper bound of VCdim(H).

Randomly get four point, we can split all case into three cluster, four points co-line, three points make a triangle and the latest point in the middle and four points construct a convex quadrilateral.

In first cluster, we assign four point {+1} to first and third points, {-1} to second and forth points. Then these four points can't be shatter by a circle.



First cluster



Second cluster

For second cluster, we assign {+1} the points make triangle, {-1} to the point in the middle. In this case, the smallest circle for {+1} points will be circumcircle of the triangle, then the point in the triangle must be in this circle, this label must be {+1} too. So there is a contradiction there.

In third cluster, choose two point which can make a line and separate the rest two point as a set, and the other two point as another set. Then calculate the distance of point in each set and mark the longer distance set's points as F1, F2, the other two will label as S1, S2.

So now we have F1, F2, S1, S2 and dis(F1F2) > dis(S1S2). Then we label F1 and F2 {+1}, S1 and S2 {-1}. If we want circle to label S1 and S2, the radio of circle R should be less then dis(S1S2) i.e. R < dis(S1S2). And if we want circle to label F1 and F2 correctly, R should be larger than dis(F1F2) i.e R > dis(F1F2). Then there is a contradiction for R!

Combined all those three case, the upper bound of VCdim(H) should be less than 4. If VCdim(H) should less than 4 and at least 3, VCdim(H) can only be 3.