Comp 135 hw4

Fanying Ye

Due: December 3 2015

1 Introduction

In this assignment I conducted experiments to implement the primal and kernel version of k-nearest neighbors and the perceptron algorithm and evaluate the accuracy of them.

2 Plots and Conclusion

2.1 primal and kernel version of k-nearest neighbors and perceptron

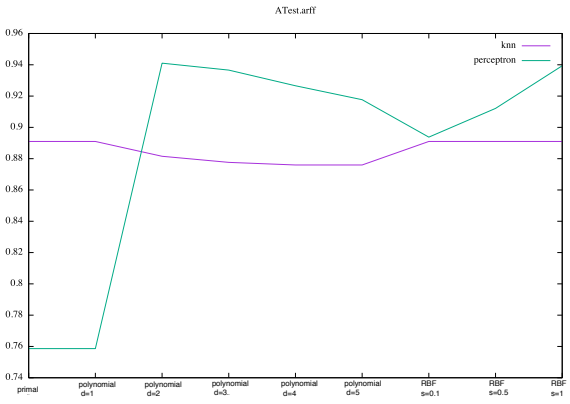
According to the plots, we can see that primal and dual version of algorithms with linear kernel indeed identical for these four datasets.
Using Knn, for the effect of kernel parameter, we can see that accuracy tends to increase with the increase of parameter s in RBF kernel. While for the d in polynomial kernel, accuracy decrease with the increase of d for datasets A and B; accuracy increase with the increase of d when d is smaller than 3 but decrease with the increase of d afterwards for datasest back and sonar.
Using perceptron, for the effect of kernel parameter, we can see that accuracy tends to increase with the increase of parameter s in RBF kernel also. While for the d in polynomial kernel,accuracy increase with the increase of d when d is smaller than 2 but decrease with the increase of d afterwards for datasets A and B; accuracy increase with the increase of d when d is smaller than 3 but decrease with the increase of d afterwards for datasest back and sonar.
Thus we can say that results are often very sensitive to kernel parameters.
Comparing the effect of the kernel parameter to the percptron with the effect to knn, we can conclude that, the effect of the kernel parameter s in RBF kernel is quite consistent across algorithm. While the effect of kernel parameter d in polynomial kernel is not that consistent across algorithm, especially for datasets A and B.
How do Perceptron and k-NN compare in the experiments across the kernels?: As we can see, perceptron's performance is often better than knn in A, B and sonar datasets, in which data are high dimensional or more noisy. We can also conclude that knn performs better in primal kernel than percptron in 4 datasets.

Dataset	Primal	Poly d=1	Poly d=2	Poly d=3	Poly d=4	Poly d=5	RBF s=0.1	RBF s=0.5	RBF s=1
A	0.8909	0.8909	0.8815	0.8776	0.8759	0.8759	0.8909	0.8909	0.8909
B	0.768	0.768	0.7452	0.7029	0.6850	0.6829	0.308	0.768	0.768
back	0.7741	0.7741	0.8387	0.8064	0.8064	0.7741	0.7419	0.7741	0.7741
sonar	0.85	0.85	0.8	0.85	0.7	0.6	0.45	0.6	0.85

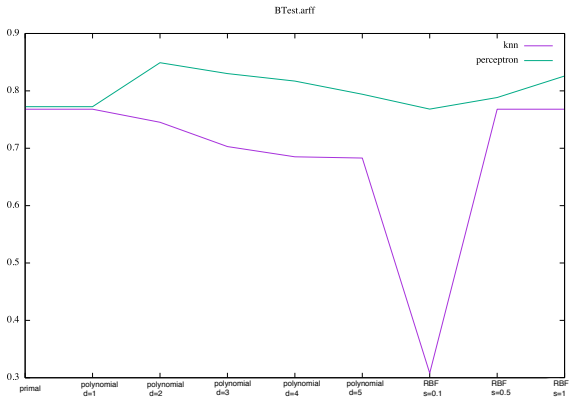
Table 1: Accuracies under primal and kernel version of k-nearest neighbors of 4 datasets

Dataset	Primal	Poly d=1	Poly d=2	Poly d=3	Poly d=4	Poly d=5	RBF s=0.1	RBF s=0.5	RBF s=1
A	0.7586	0.7586	0.9410	0.9365	0.9265	0.9176	0.8937	0.9121	0.9393
B	0.7724	0.7724	0.8490	0.8301	0.8170	0.7941	0.7682	0.7884	0.8258
back	0.6774	0.6774	0.8064	0.8387	0.8387	0.7419	0.7741	0.7741	0.7741
sonar	0.85	0.85	0.85	1.0	0.85	0.85	0.55	0.85	0.85

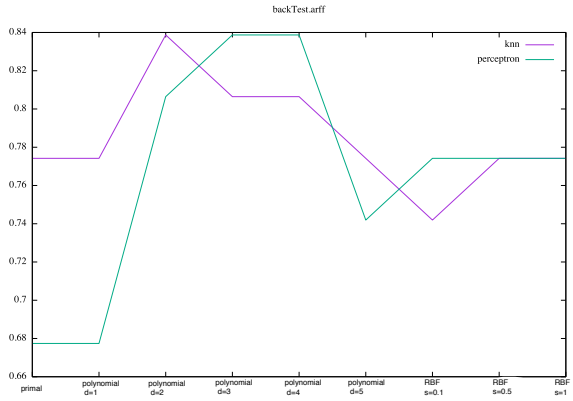
Table 2: Accuracies under percptron algorithm of 4 datasets



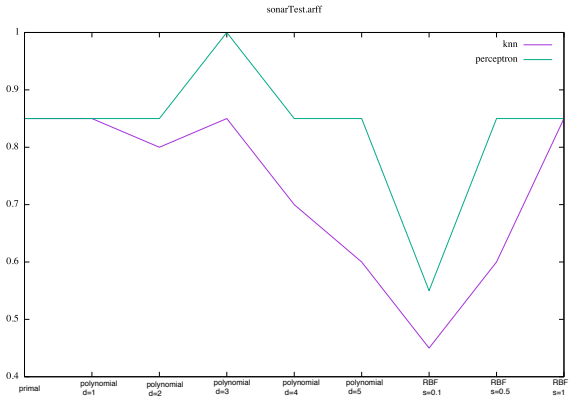
(a) accuracies on A



(b) accuracies on B



(c) accuracies on back



(d) accuracies on sonar

Figure 1: Accuracies under primal and kernel version of k-nearest neighbors and perceptron of 4 datasets