**Chapter 6: Additional Classic models**

Theoretical part:

1. SVM

Read about SVM from ISLR pages 337-359

Answer questions 3,6 in exercise section

Then proceed with the next question:

In the next exercises assume the data is 2 dimensional and distributes uniformly on [0,1] X [0,1]

unless otherwise stated.

**Part 1**

In this exercise we assume the hypothesis we want to learn is classification by y=1-x:

Class = 1 if y>1-x, otherwise 0

Create a train dataset by generating 1000 labeled points

Create a test dataset by generating 1000 unlabeled points

Classify each point according to a different kernel

Show them on the same plot (one for every kernel). Correctly classified points in blue, misclassified in red. Try at least 3 different kernels which you see fit. Also plot the separating line.

What is the difference in total train and test time between different kernels?

Choose the appropriate kernel and use it in the next parts:

**Part 2**

Extend the hypothesis for higher dimension while keeping the training and test sets balanced. Classify each point according to your chosen kernel. How are the training and test times depend upon the dimension? (what is the relation)

**Part 3**

Denote the number of train samples N, and the dimension P. Test samples number is fixed. For every region N<<P, N=P, N>>P show the test error. Plot all results one informative graph.

**Part 4**

Repeat the exercise 1 with hypothesis of circle locates at (0.5,0.5) with radius of size 0.25. No need to plot separating line.

Try this using linear and using polynomial kernel of deg 2.

1. Bayesian:

**Theoretical**

* Read chapter 5.6 in Deep Learning book.
* For illustrative purposes, read the 3.3-3.3.1 chapters in the book Bishop – Pattern Recognition

**Theoretical intermediate expansion (for those interested).**

* **Expanding** and more detailed on the differences in fundamental approaches can be read from MIT which explains in depth the pros and cons of the two different approaches of statistics for understanding distributions and prediction.
* You can also read about the entire method with an example from the original book accompanying the MIT course:

Read chapter 8 in introduction to probability

**Practical**

Bayesian methods are quite rare in most of the ML problems encountered, but it has a very big advantage when you do have a prior of your business knowledge, and only a little bit of data. This can be encountered in our day to day work.

* In the next blogpost the writer describes his use of the library that realizes BLR and makes a comparison of the success of various models: he tries to predict grades of students based on their raw data (relatively few students, with lots of data), and in addition it shows how you can explain the model.

He's a little digging, you can skim, but pay attention to the following ideas:

* How he builds the model
* MCMC as sampling method
* How it uses a model for paradigm-giving and a security measure evaluation.

[https://towardsdatascience.com/bayesian-linear-regression-in-python-u](https://towardsdatascience.com/bayesian-linear-regression-in-python-using-machine-learning-to-predict-student-grades-part-2-b72059a8ac7e)

[sing-machine-learning-to-predict-student-grades-part-2-b72059a8ac7e](https://towardsdatascience.com/bayesian-linear-regression-in-python-using-machine-learning-to-predict-student-grades-part-2-b72059a8ac7e)

For more and more intuitive explanation about MCMC You can see the explanation from the next blog section, which is better specs on this point:

<https://towardsdatascience.com/an-introduction-to-bayesian-inference-e6186cfc87bc>

Most of the examples you've seen here are about regression, of course you can turn any model into a Bayesian model, even the NN field – a research field that tries to complete the gap with the recent successes of non-Bayesian neurons networks (an ancient domain in itself) that were built out of an empirical trial in contrast to Bayesian networks.