Good Morning everyone, today we are going to discuss the impact of hard margin and soft margin in Support Vector machines.

**Learning objectives**

These are the learning objectives. Be familiar with SVM hard margin and soft margin concepts and how to select the suitable margin according to your data set, and in a way that it maximize the classification accuracy.

The role of C parameter, and finally choosing the Optimal C value.

**Plan for today**

This is what we are going to discuss today.

**What Is SVM**

What is SVM? As you all know, it is a supervised learning algorithm used in classification and regression tasks. Mostly in classification problems. . It works by finding the hyperplane that best separates data into different classes while maximizing the margin between them. As you can see, hyperplane is a decision boundary that separate different classes in the data. Margin is the distance between hyperplane and the nearest data points in each class.

**Hard Margin**

* In terms of the margin, there are two types, hard margin and soft margin. Hard margin perfectly separate the data by hyperplane. **No points lie inside the margin or no misclassifications are allowed.** This is the equation for hard margin SVM. But in reality a perfect separation may not be possible. Data often contains noise or outliers. So the strict margin can lead to the overfitting.

**Soft Margin**

Soft margin allows some misclassifications to handle non separable data.

Here, we have to think balancing of maximising the margin while minimizing the misclassifications. Here, what SVM does is finds the maximum margin and it adds a penalty each time a point crosses the margin.

minimize *(1/margin+∧(∑penalty))*

**Slack Variable**

Each data point has its own slack variable to find how much it violates the margin. It takes the distance of a misclassified point from its correct margin boundary.

To make a soft margin equation we add 2 more terms to this equation which is **zeta**and multiply that by a**hyperparameter ‘c’**

**C parameter**

C is a hyperparameter that controls the trade-off between maximizing the margin and minimizing margin violations. It acts as a penalty for slack variables.

* When you have a higher C value, that strongly penalizes margin violations leading to smaller margin with fewer misclassifications.
* A lower C value, less penalizes margin violations, focus on wider margin allowing more misclassifications.

**Visualization**

Here, you can see I have changed the C value, lets go for the code to see how this looks in practice. In the first plot C = 0.1 and there is a wider margin with data points inside the margin and one misclassified point in the hyperplane. The second plot with C value = 1, the margin gets slightly smaller and a few data points inside the margin with one misclassified point. In higher C values you can see, smaller margin with perfect separation of data points. No misclassified points.

**Choosing the better model**

From these three figures which one is correct. The first figure, second one or third one. There is no correct answer. That depends on your dataset. If there are outliers in your dataset, smaller C value helps you balancing between maximizing margin and minimizing classification errors. If you have a cleaned dataset, larger C will perfectly separate the data. But careful about overfitting.

So you can justify best model using Margin error + classification error

**Optimal C**

Optimal C value depends on your data set (the noise level and data distribution) + the kernel you have used. You can perform Grid search optimization or Cross validation to find the optimal C value.

**Case Study**

This is a case study of Medical Diagnostics. I have taken this from a research paper. The source is mentioned here. This research is based on finding the hyper parameters C and gamma for different kernels to identify the best combination which maximizes the classification accuracy.

This has chosen different kernels(RBF, Sigmoid kernel and linear kernel) with different C values and Gamma values and perform cross validation to find best combinations.