Neural Networks for Signal Processing-I

Prof. Shayan Srinivasa Garani

Department of Electronic System Engineering

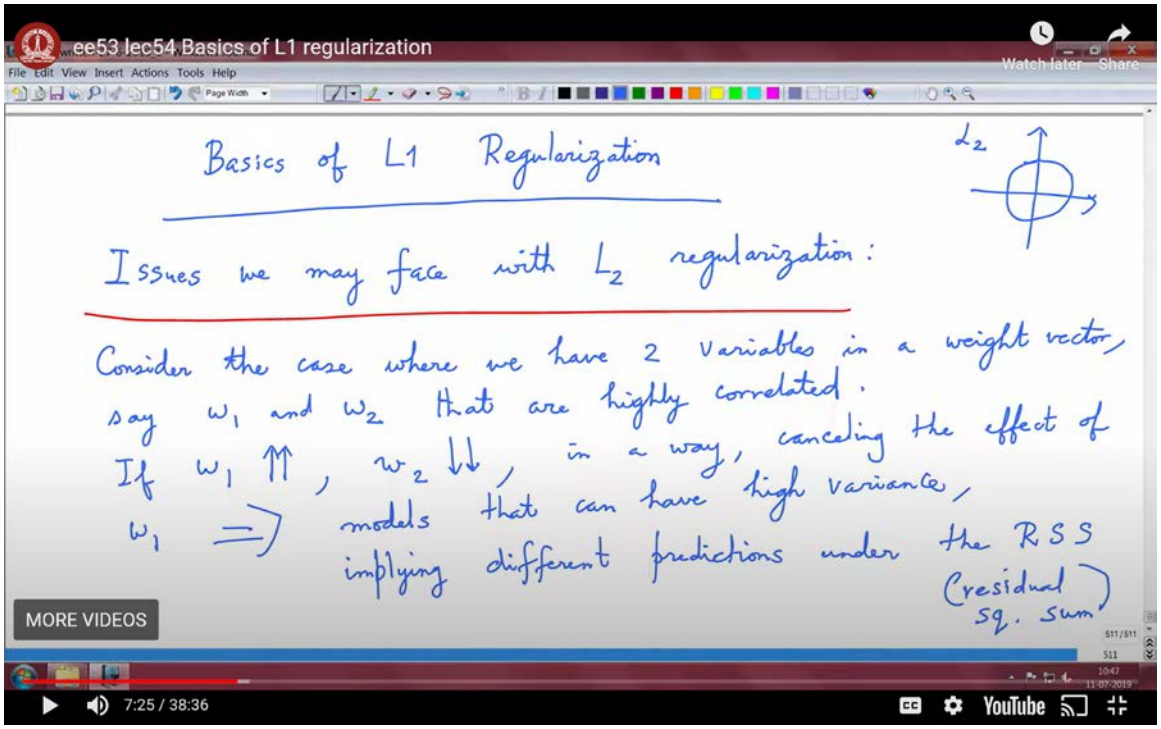
Indian Institute of Science, Bengaluru

Lecture – 54

Basics of L1 Regularization

Let's delve into the basics of L_1 regularization and some related algorithms. Previously, we discussed the necessity of setting up a cost functional with regularization constraints and explored Tikhonov regularization, which transforms an ill-posed problem into a well-posed one. But what type of constraints should we impose? The choice depends on the problem's geometry.

(Refer Slide Time: 07:25)

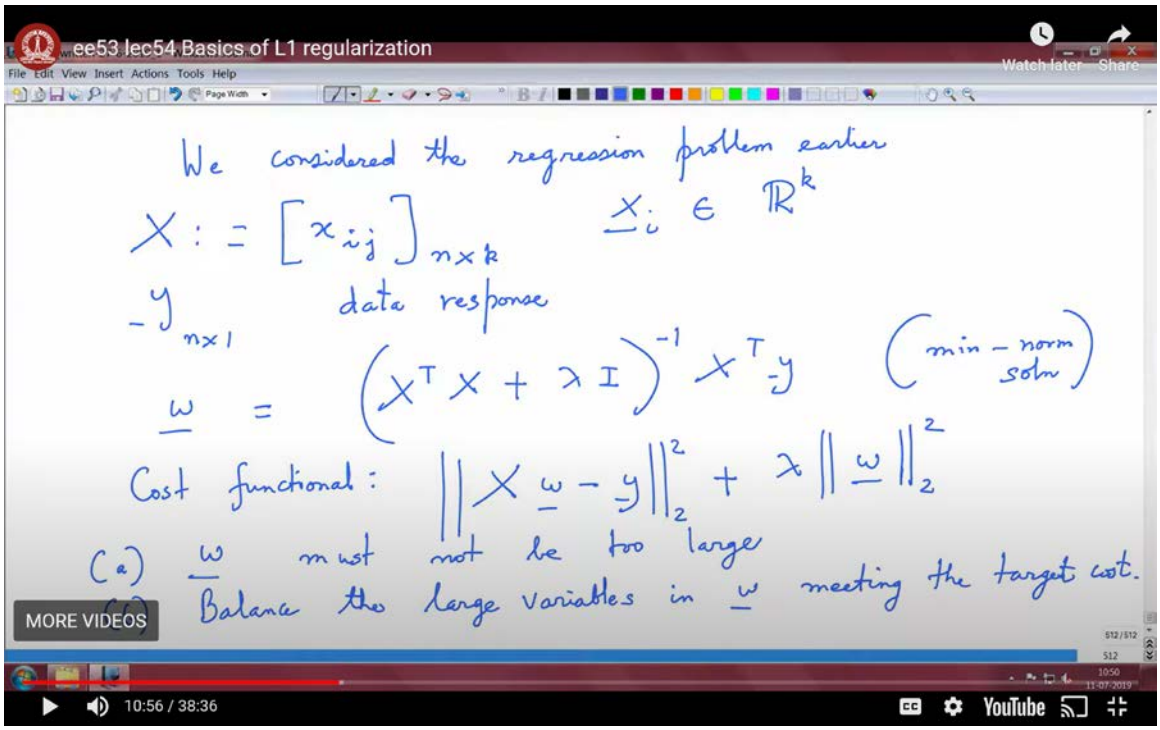


For instance, if our cost function involves minimizing the sum of the squares of the weights, we're essentially working with the induced L_2 norm. Minimizing this norm aligns with our

goal in the L_2 sense. However, this may not always be the constraint we're interested in. Sometimes, we might want to focus on minimizing the number of non-zero elements in our weight vector, which leads us to L_1 regularization. This approach is useful when we aim to achieve sparsity in our model.

The choice of regularity constraints is therefore influenced by the problem's geometry and practical considerations. With this in mind, we'll now formally introduce L_1 regularization, discuss its motivation, and explore how to set up both unconstrained and constrained optimization frameworks. We'll also examine some algorithms related to these concepts.

(Refer Slide Time: 10:56)

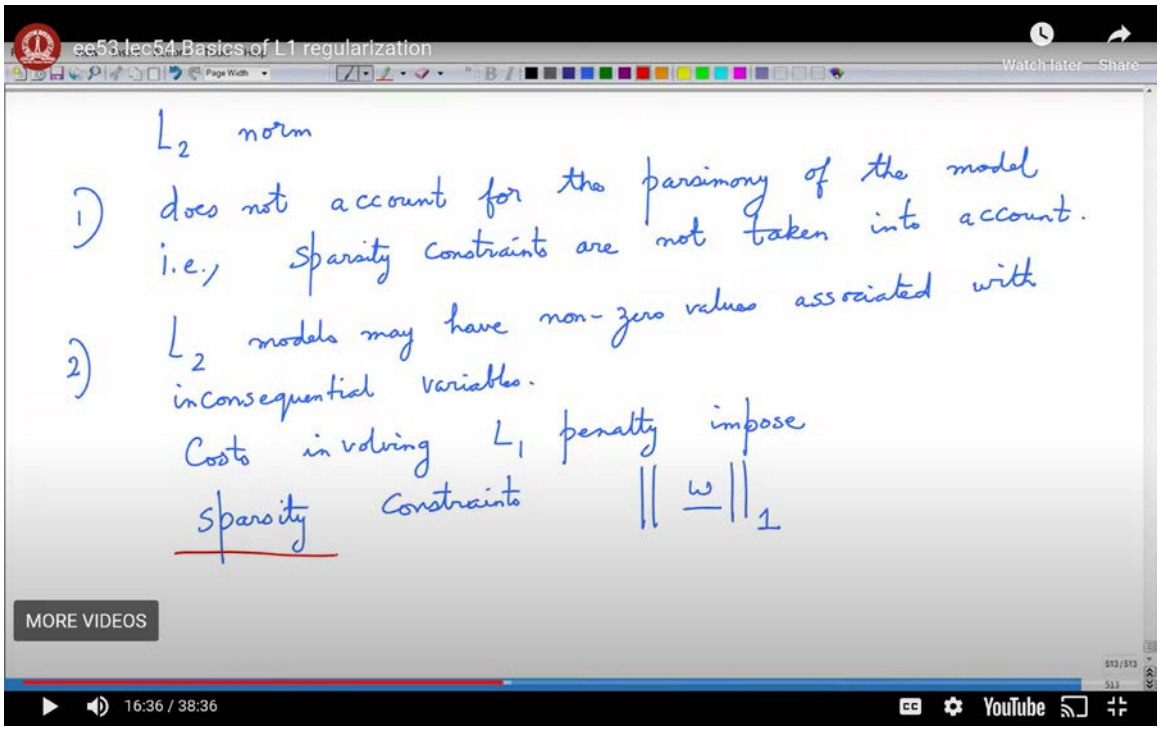


Firstly, let's address some issues associated with L_2 regularization. Consider a weight vector with two variables, w_1 and w_2. Suppose w_1 and w_2 are highly correlated. If w_1 increases, w_2 might decrease in such a way that it cancels out the effect of w_1. This correlation can lead to models with high variance, meaning that predictions can vary significantly under different conditions, particularly when evaluated using the Residual Sum of Squares (RSS) metric.

The L_2 norm has a specific geometric interpretation: it operates within a ball. When regularizing with L_2, you constrain the optimization to lie inside or outside a circular boundary. This geometric constraint influences how the problem is approached and solved within the L_2 framework.

The challenge we're facing is that L_2 regularization does not account for model sparsity. This means that variables which may be irrelevant to the problem or those that are highly correlated can lead to one variable offsetting the effect of another. To address this, we need a different geometric approach, which is where L_1 regularization comes into play.

(Refer Slide Time: 16:36)



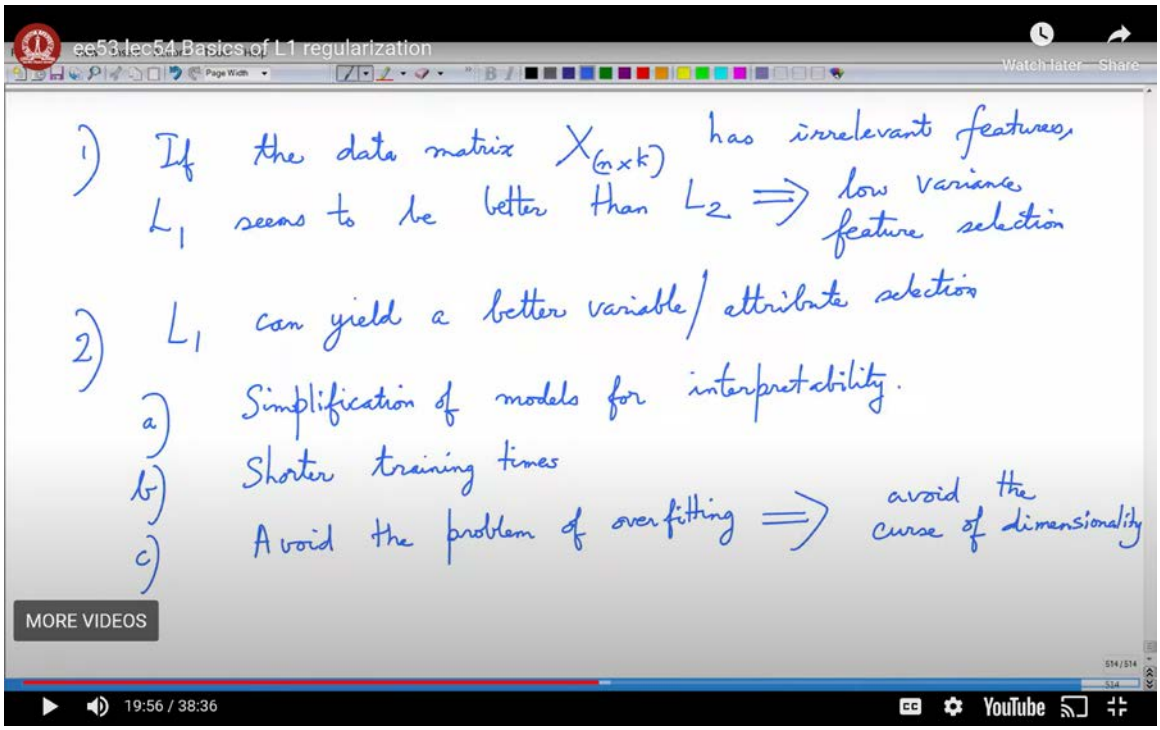
Let's revisit our regression problem. Given a data matrix X, which is n × k where each vector x_i belongs to R^k, and a data response vector y which is n × 1, our goal is to find the best fit line for this data while adhering to regularity constraints. The solution to this line corresponds to the minimum norm solution, which we have previously discussed.

In the L_2 norm setup, we aim to minimize $|Xw - y|_2^2$, subject to a constraint that the norm of the weight vector w should not be too large. This approach balances the weight variables,

ensuring they do not exceed certain limits while meeting the target cost. Essentially, we want to keep the weights controlled to avoid any variable from becoming excessively large.

However, the problem with L_2 regularization is that it can lead to issues when dealing with correlated variables. For instance, if one variable increases while another decreases in a way that cancels out the effect, it can result in models with high variance, leading to inconsistent predictions under different conditions. While L_2 regularization handles these issues smoothly because it results in a convex optimization problem, it does not necessarily lead to a sparse or parsimonious model.

(Refer Slide Time: 19:56)



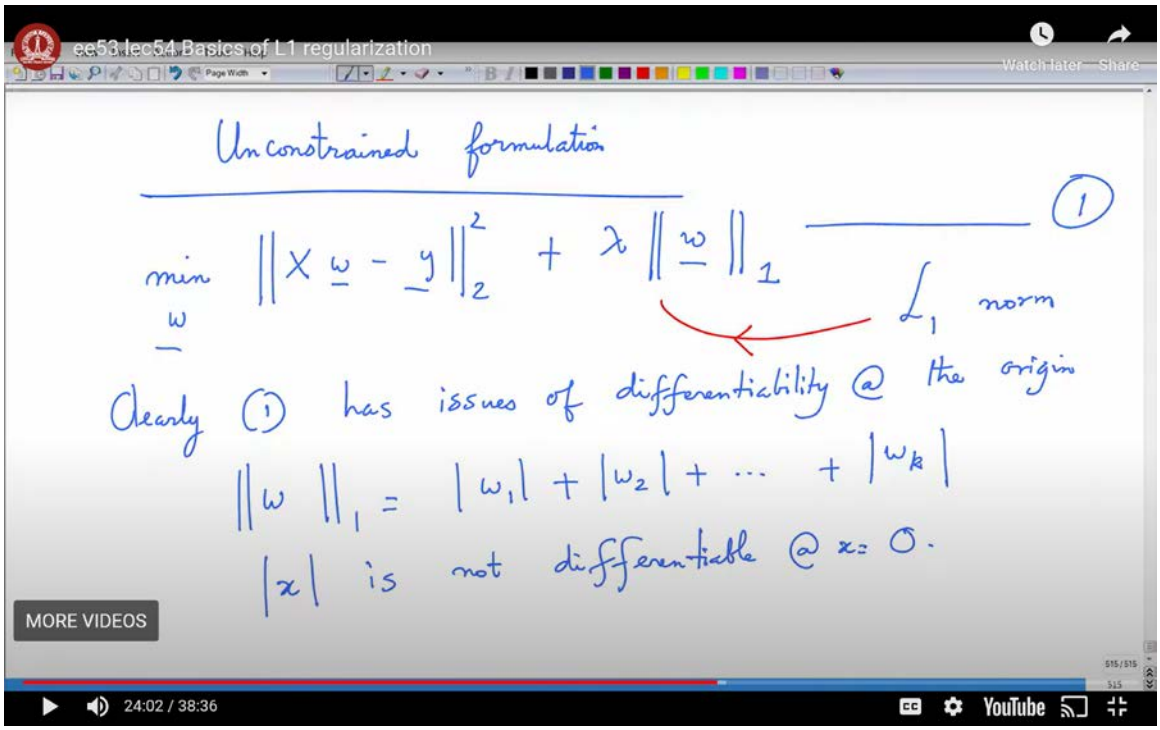
Let's consider practical constraints, such as in neural networks where we use backpropagation and other algorithms. If we encounter dead neurons or insignificant synaptic weights, keeping these weights is impractical due to computational complexity and memory constraints. Storing and computing with all these weights is both memory-intensive and burdensome. Hence, L_2 regularization, which does not encourage sparsity, may not be ideal for such practical scenarios.

To recap, L_2 norm does not account for model sparsity, meaning it does not address the issue of having many non-zero values in inconsequential variables. For instance, in the case of analyzing DNA strands, we might have hundreds of features, but only a subset of these features may be relevant for inferring gene expression. Identifying which variables to retain and which to discard is crucial and highly specific to the problem at hand.

Thus, L_1 regularization is motivated by the need to handle such issues, promoting sparsity in the model and thus addressing the limitations of L_2 regularization.

Let's dive into the scenario where we want to impose L_1 regularization constraints on the weight vector, focusing on the L_1 penalty which promotes sparsity. In L_1 regularization, you take the absolute value of each weight, sum these absolute values, and ensure that this sum is less than a pre-specified number.

(Refer Slide Time: 24:02)



In practical terms, if our data matrix X contains irrelevant features, L_1 regularization is often more effective than L_2. This is because L_1 regularization promotes feature selection

by driving some coefficients to exactly zero, which in turn simplifies the model. The benefits of better variable or attribute selection through L_1 regularization are significant:

1. Model Simplification for Interpretability: A simpler model is easier to interpret, allowing us to make clearer observations and understand the data better.

2. Reduced Training Times: Simplified models often lead to shorter training times, enhancing efficiency.

3. Mitigation of Overfitting and Curse of Dimensionality: By removing irrelevant features, L_1 regularization helps in avoiding overfitting and managing high-dimensional data more effectively.

These advantages highlight why L_1 regularization is preferable in certain scenarios over L_2.

Now, let's set up the unconstrained formulation. We aim to minimize the squared L_2 norm of Xw - y over all possible choices of w, subject to the constraint imposed by L_1 regularization. Specifically, we are interested in minimizing:

$$|Xw - y|_2^2$$

subject to the L_1 norm of w being less than or equal to a certain bound. Let's call this Equation 1.

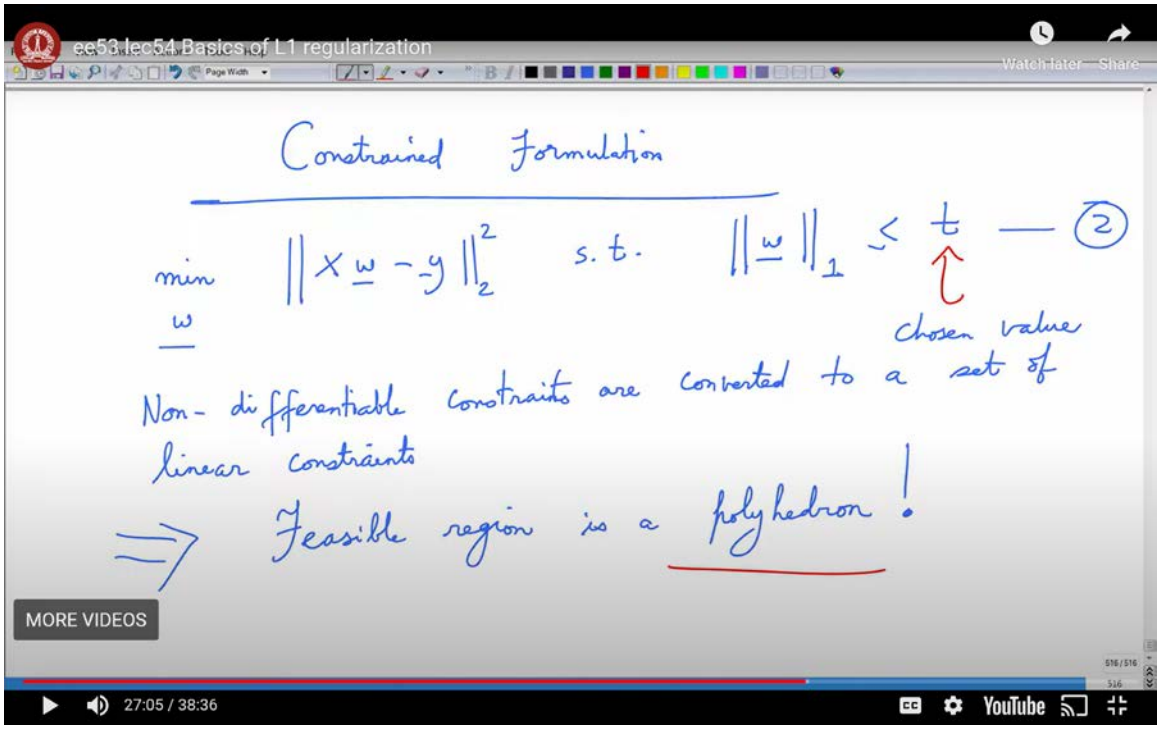
Previously, with L_2 regularization, we minimized $|Xw - y|_2^2$ subject to the norm of w being minimized, which led to a closed-form solution. For L_1 regularization, the situation differs. Here, the L_1 norm of w introduces sparsity constraints, making the geometry of the problem distinct.

In L_1 regularization, the critical aspect is the absolute value function, which impacts the differentiability of the objective function. Specifically, the L_1 norm is:

$$\text{norm}_{l_1} = |w_1| + |w_2| + \cdots + |w_k|$$

Differentiating this norm with respect to each coordinate presents challenges because the absolute value function is not differentiable at zero. This lack of differentiability complicates finding a closed-form solution, unlike the smooth quadratic cost function in L_2 regularization.

(Refer Slide Time: 27:05)



In the constrained formulation, the only difference from the unconstrained case is that we now want to bound the L_1 norm of w within a specific limit. For example, we might constrain the norm of w such that:

$$|w|_1 \leq t$$

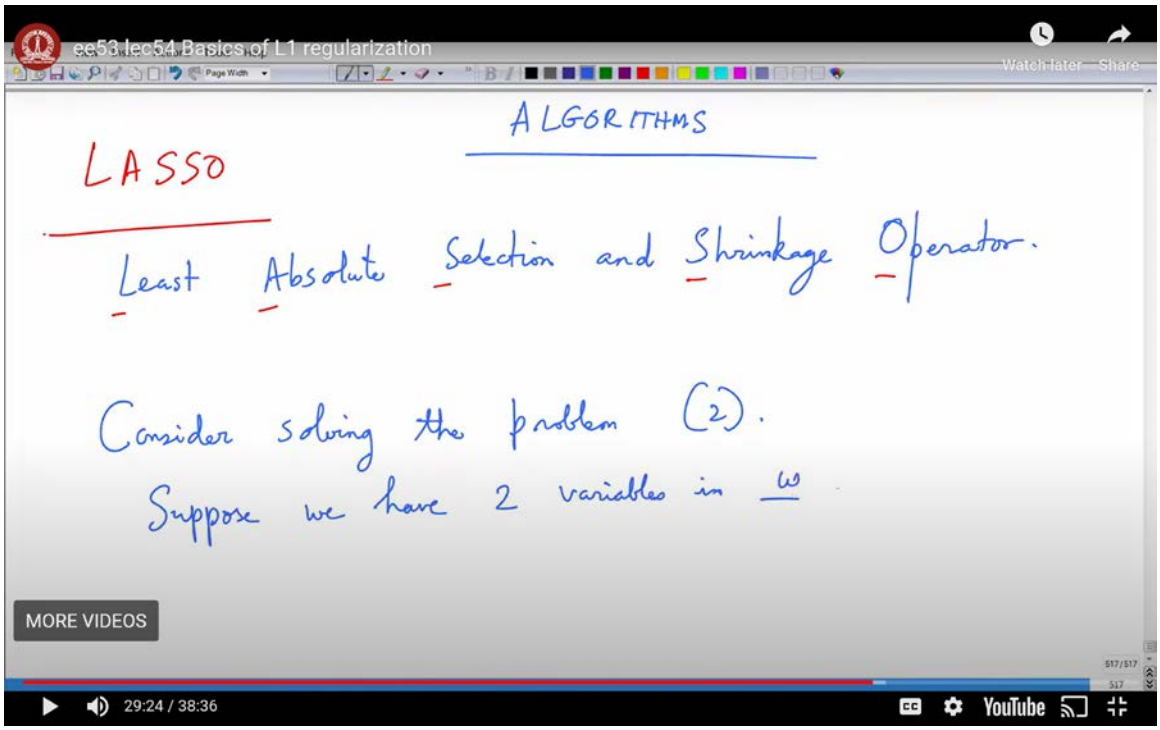
This subtle difference has significant implications for setting up and solving our optimization problem.

Let's consider the optimization problem where we want to minimize the squared Euclidean norm $|Xw - y|_2^2$ subject to the constraint that the L_1 norm of w is less than or equal to t, where t is a chosen constant. In this scenario, we are interested in the absolute values of

the weights in the weight vector w, and we require that their sum is constrained to be less than or equal to t. Essentially, this means:

$$|w|_1 = |w_1| + |w_2| + \cdots + |w_k| \leq t$$

(Refer Slide Time: 29:24)



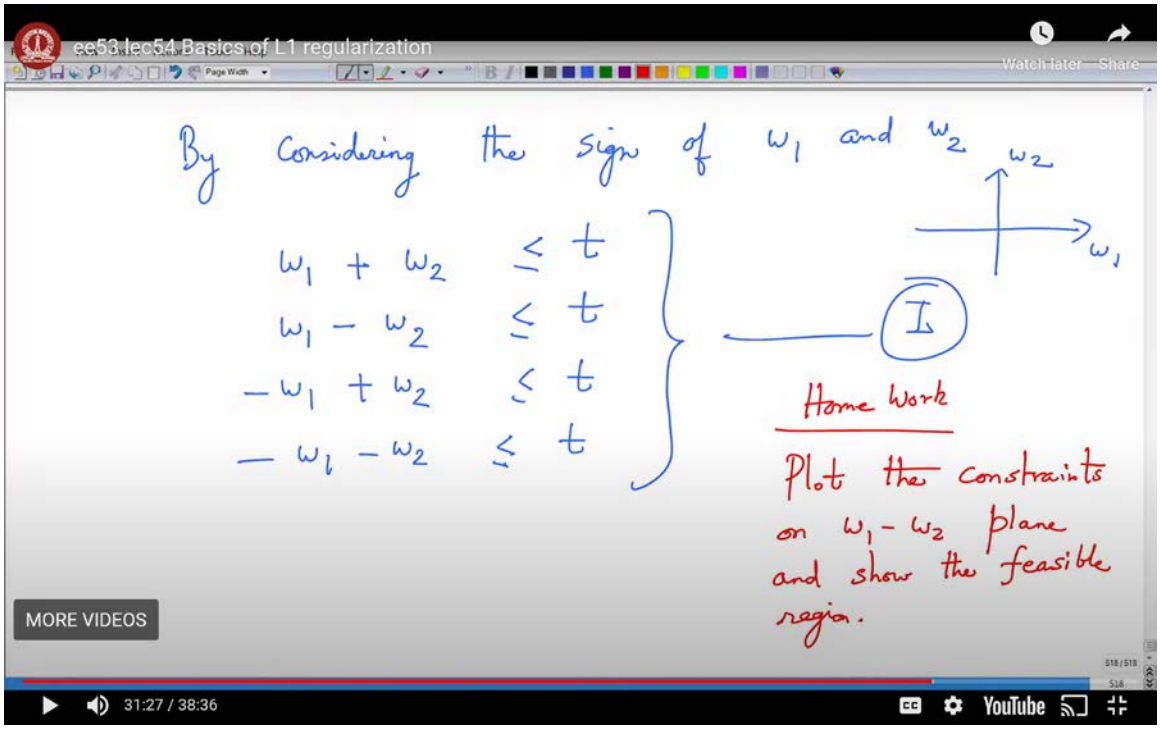
This forms what we call a set of non-differentiable constraints, which we will refer to as Equation 2. To handle these non-differentiable constraints, we convert them into a set of linear constraints, which means that the feasible region for this problem becomes a polyhedron.

For a deeper understanding of this geometry, I recommend reviewing the lecture on norms from the course *Mathematical Methods and Techniques for Signal Processing*. This lecture provides detailed insights into the geometry of normed spaces, which will help clarify these concepts.

Now, with the regularization problem set up, we can discuss algorithms designed to solve such optimization problems. One prominent algorithm used in this context is LASSO,

which stands for Least Absolute Shrinkage and Selection Operator. This technique, popular in statistics, was developed by Tibshirani and is specifically designed to address problems involving L_1 regularization.

(Refer Slide Time: 31:27)



To motivate the use of LASSO, consider the constraint optimization problem given by Equation 2. Suppose we have two variables, w_1 and w_2. We have four possible constraints based on the signs of w_1 and w_2:

$$1. w_1 + w_2 \leq t$$

$$2. w_1 - w_2 \leq t$$

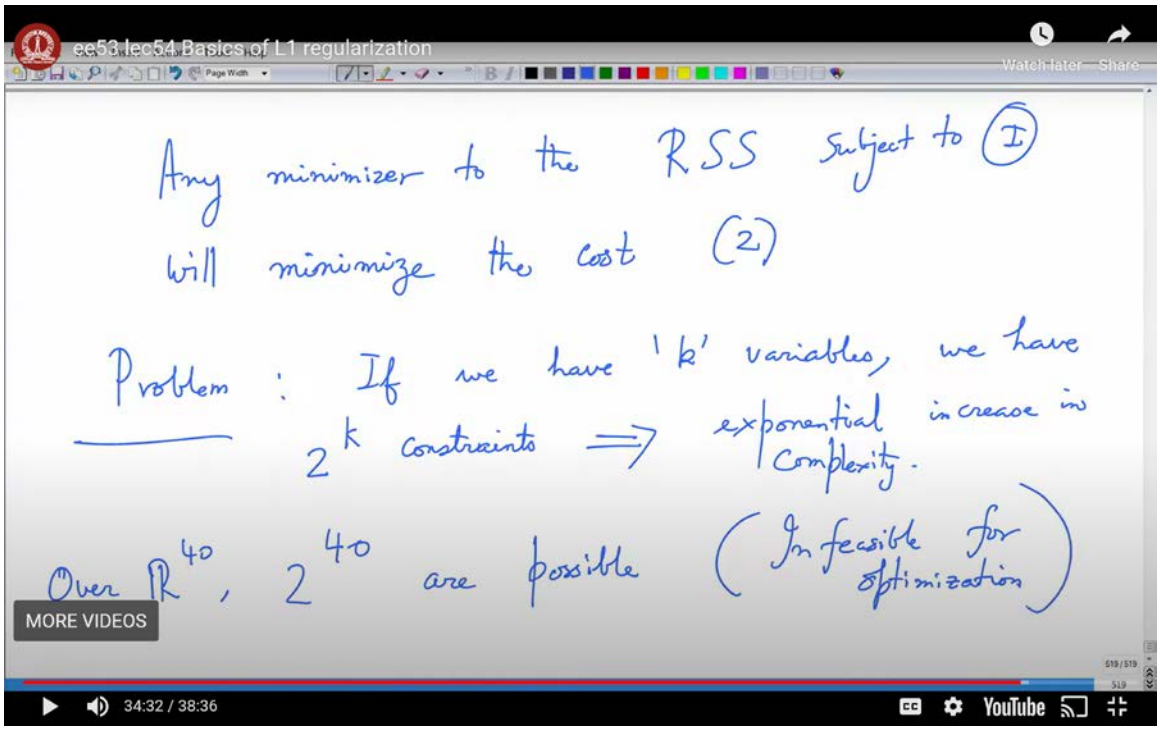
$$3. -w_1 + w_2 \leq t$$

$$4. -w_1 - w_2 \leq t$$

These constraints can be visualized in the w_1-w_2 plane, resulting in a feasible region that can be plotted based on these inequalities. You will see lines with slopes of 1 and -1, creating a geometric shape in the plane.

I encourage you to plot these constraints yourself as an exercise. As a graduate student, you should be able to draw the feasible region defined by these inequalities and understand how the solution to the original cost function $|Xw - y|_2^2$ fits within this feasible region.

(Refer Slide Time: 34:32)

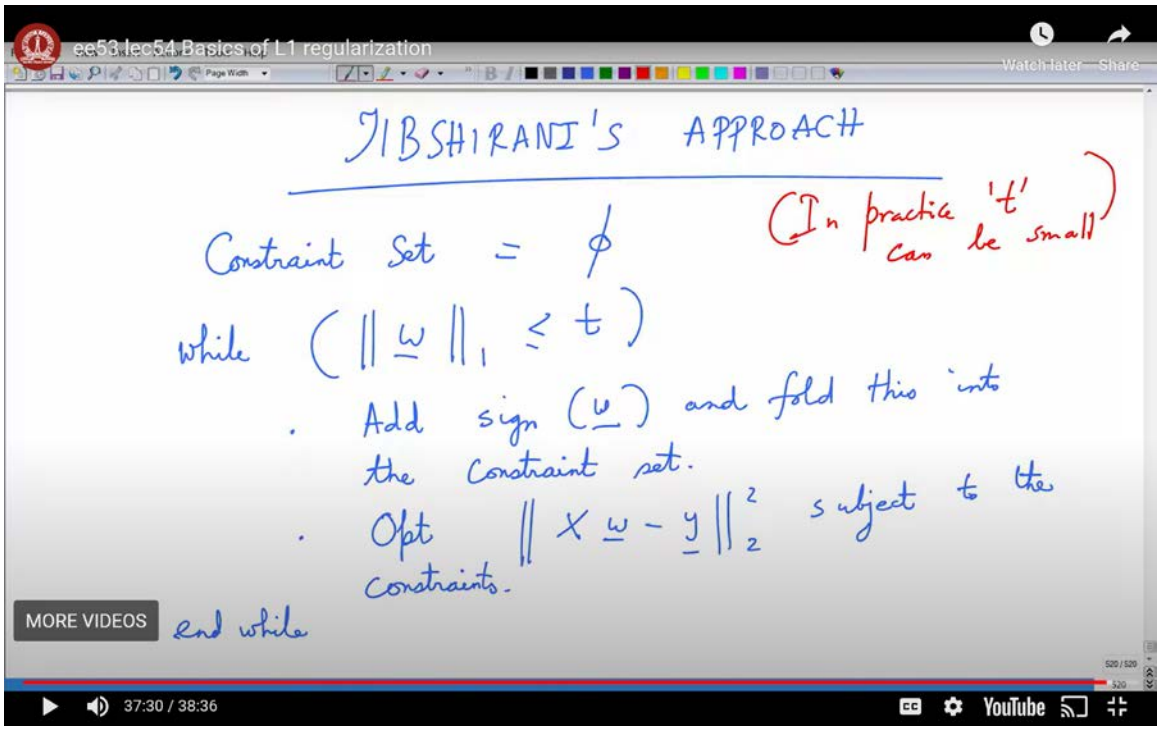


To address the problem of minimizing the squared residual sum under L_1 regularization constraints, we need to solve for the L_2 norm subject to the linear constraints that arise from the L_1 regularization. Specifically, we aim to minimize the L_2 norm squared of the residuals $|Xw - y|_2^2$ subject to the constraints imposed by the L_1 regularization. These constraints are captured by Equation 2, which defines the cost formulation we are optimizing.

Here's the challenge: With k variables, there can be 2^k possible linear constraints. This exponential growth in the number of constraints quickly becomes impractical. For example, in a 40-dimensional space, the number of constraints is 2^{40}, which is infeasible

for most optimization algorithms, even with advanced computational resources like cloud computing or multi-core processors.

(Refer Slide Time: 37:30)



To manage this complexity, consider the following approach inspired by Tibshirani's method. Start with an empty constraint set and iterate through a loop. While the L_1 norm of the weight vector w is less than or equal to t, add the signs of the weight vector w to the constraint set. Then, optimize the L_2 norm of the residuals $|Xw - y|_2^2$ subject to these constraints.

During each iteration of the loop:

1. Add the constraints based on the signs of w to the constraint set.

2. Optimize the l_2 norm under the updated constraints.

This process is repeated until all constraints are satisfied. Despite the theoretical feasibility, this method is computationally intensive due to the exponential number of constraints and the necessity to update constraints dynamically.

In practice, t can be small, but solving the problem efficiently requires careful handling of constraints and optimization iterations. As you add new constraints, the initial conditions of your previous solutions may no longer be applicable, leading to complex and potentially inefficient algorithms.

By carefully managing these iterations and constraints, we can ultimately find a weight vector w that satisfies all the constraints, providing a solution to the optimization problem. We'll discuss some practical considerations for this approach shortly.