Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 57

Hello everyone, this is the second lecture of week twelve. Recall that in the first lecture, we had a quick introduction to machine learning, specifically supervised learning. A supervised learning problem is one where you are given a labeled dataset. You have n data points; in each data point, you have a feature vector, say x_1, and the corresponding label y_1. For x_2, you have y_2, and so on, up to x_n and y_n.

You have n feature vectors in the data. There is an assumption that y and x are connected by a function f. So, if you are given an x, you can find y if you know f. We also have a loss function L.

Instead of finding f, if you find an estimate $\hat{f}$, the loss is given by the loss function L. Our task is to find f in such a way that the loss function for a given test vector is minimized. In general, if we do not know anything about f, it is very hard to estimate it. Different methods make simplifying assumptions. I mentioned some methods that assume f is linear, meaning y is linear in x. There was also the decision tree method, which assumes y is constant within axis-parallel rectangles. There are many such assumptions. We are going to consider the simplifying assumption where f is linear, and we will assume the loss is squared loss.

Note: $x_{i,j}=x_i^{(j)}$, $x^i=x^{(i)}$

In the previous lecture, we saw that if f is linear, then y can be written as

$f(x) = \beta_0 + \sum_{i=1}^{p} \beta_i x^i$.

Here, x is a p-dimensional vector $(x^1, x^2, ..., x^p)$.

I am writing the component index as a superscript because the subscript denotes the data point. So, x_1 itself is a vector, and its components are written with superscripts.

For the estimate $\hat{f}(x)$, which you can call $\hat{y}$, we have

$\hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i x^i$.

We want to minimize the squared loss $L(f(x), \hat{f}(x)) = ||y - \hat{f}(x)||^2 = ||y - \hat{\beta}_0 - \sum_{i=1}^{p} \hat{\beta}_i x^i||^2$.

Note that we are given y and x for n data points.

The task is to minimize $\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{i,j})^2$ over $\hat{\beta}_0, ..., \hat{\beta}_p$.

This is a well-defined problem called Ordinary Least Squares (OLS). OLS means we are minimizing the sum of squares.

If you give values for $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$, you get a loss for each data point. For the first data point, the loss is $L_1 = ||y_1 - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{1,j}||^2$.

For the second, it is $L_2 = ||y_2 - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{2,j}||^2$,

and so on, up to $L_n = ||y_n - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{n,j}||^2$.

We minimize the sum of all these losses.

The values of $\hat{\beta}_0, ..., \hat{\beta}_p$ that minimize this sum are the OLS solution.

This is called ordinary least squares because we are minimizing the sum of squares. It is the simplest least squares method, often the first machine learning problem taught, as it involves minimizing a quadratic function. Since it is a quadratic problem, we can use all the methods we learned in unconstrained optimization.

For a quadratic objective function of the form $\frac{1}{2}x^T Hx + b^T x + c$,

we know the solution is $x = -H^{-1}b$.

We can write the OLS problem in matrix form. Define a matrix H with a column of ones followed by the feature values:

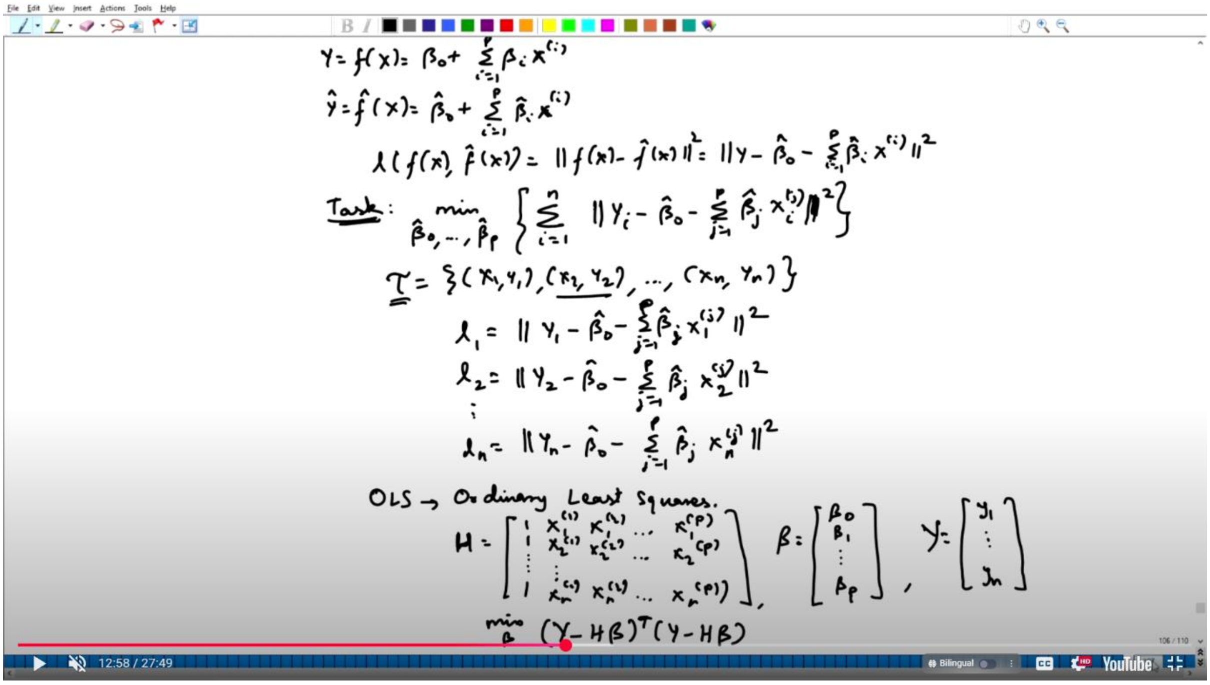
the first row is $[1, x_{1,1}, x_{1,2}, ..., x_{1,p}]$,

the second row is $[1, x_{2,1}, x_{2,2}, ..., x_{2,p}]$,

and so on, up to the n-th row $[1, x_{n,1}, x_{n,2}, ..., x_{n,p}]$.

Let β be the vector $[\beta_0, \beta_1, ..., \beta_p]^T$, and Y be the vector $[y_1, y_2, ..., y_n]^T$.

(Refer Slide Time 12:58)



$$Y = f(X) = \beta_0 + \sum_{i=1}^{p} \beta_i X^{(i)}$$

$$\hat{Y} = \hat{f}(X) = \hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i X^{(i)}$$

$$l(f(x), \hat{f}(x)) = \| f(x) - \hat{f}(x) \|^2 = \| Y - \hat{\beta}_0 - \sum_{i=1}^{p} \hat{\beta}_i X^{(i)} \|^2$$

$$\underline{\text{Task}}: \min_{\hat{\beta}_0, \dots, \hat{\beta}_p} \left\{ \sum_{i=1}^{n} \| Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_i^{(j)} \|^2 \right\}$$

$$\underline{\mathcal{T}} = \{ (x_1, y_1), \underline{(x_2, y_2)}, \dots, (x_n, y_n) \}$$

$$l_1 = \| Y_1 - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_1^{(j)} \|^2$$

$$l_2 = \| Y_2 - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_2^{(j)} \|^2$$

$$\vdots$$

$$l_n = \| Y_n - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_n^{(j)} \|^2$$

OLS → Ordinary Least Squares.

$$H = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(p)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(p)} \\ \vdots & \vdots & & & \\ 1 & x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(p)} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\min_{\beta} (Y - H\beta)^T (Y - H\beta)$$

Then the objective function can be written as

$(Y - H\beta)^T(Y - H\beta)$.

Expanding this, we get $Y^TY - 2Y^TH\beta + \beta^TH^TH\beta$.

This is of the form $\frac{1}{2}\beta^TA\beta + b^T\beta + c$,

where $A = 2H^TH$, $b = -2H^TY$, and $c = Y^TY$.

The solution is $\hat{\beta} = -A^{-1}b = (H^TH)^{-1}H^TY$.

Given a dataset, we can construct H and Y and apply this formula. We could also use optimization algorithms we learned previously to find the solution. However, I will take a slightly harder problem: ridge regression.

In ridge regression, the problem is similar to OLS but with an additional constraint. We minimize the same sum,

$\sum_{i=1}^{n} ||y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{i,j}||^2$, over $\hat{\beta}$,

but subject to the constraint that $\hat{\beta}_1^2 + \hat{\beta}_2^2 + \dots + \hat{\beta}_p^2 \leq s$.

This means the sum of squares of the coefficients (excluding the intercept $\hat{\beta}_0$) should not exceed a certain value s.

This is useful due to the bias-variance trade-off. In estimation, we want estimators with good properties like being unbiased and consistent. The OLS estimator is the Best Linear Unbiased Estimator (BLUE) according to the Gauss-Markov theorem, meaning it has the least variance among all unbiased linear estimators. However, for prediction problems, we often want to minimize the mean squared error, which is variance plus bias squared. An estimator with some bias might have lower variance, leading to a lower overall error. Ridge regression accomplishes this by reducing variance at the cost of introducing some bias, for an appropriate value of s.

There is also an unconstrained formulation of ridge regression. Consider the Lagrangian of the constrained problem:

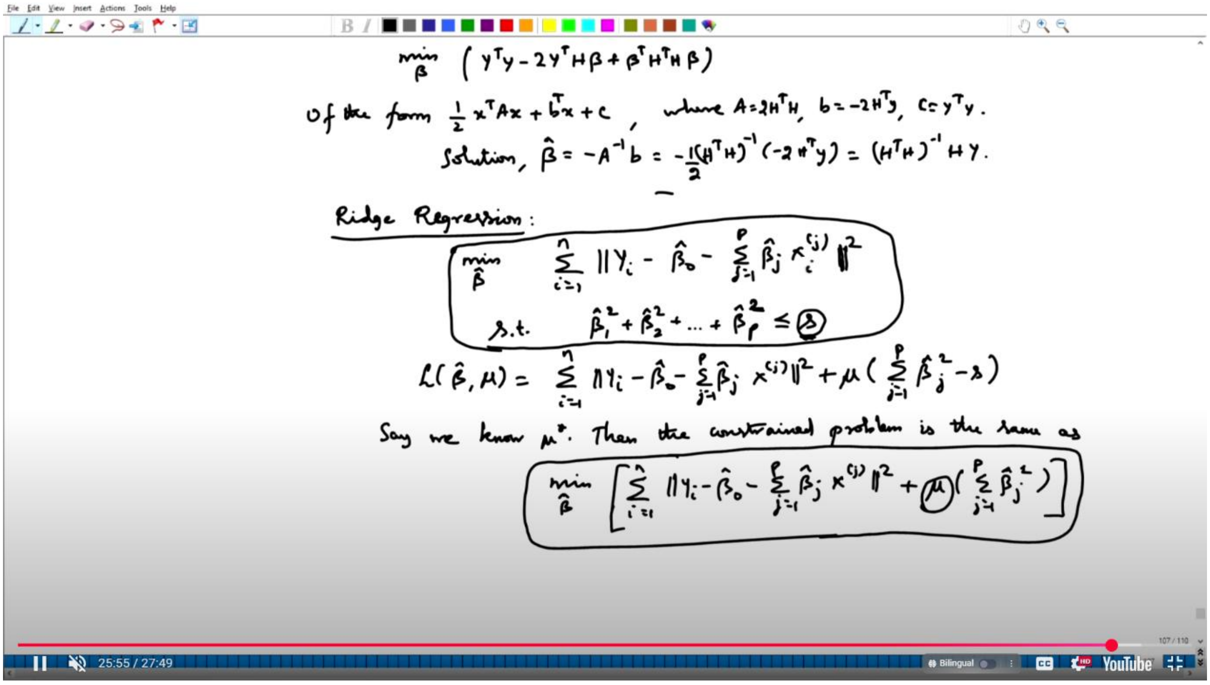
$L(\hat{\beta}, \mu) = \sum_{i=1}^{n} ||y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{i,j}||^2 + \mu(\sum_{j=1}^{p} \hat{\beta}_j^2 - s)$.

If we know the optimal Lagrange multiplier μ*, the constrained problem is equivalent to

minimizing $\sum_{i=1}^{n} ||y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{i,j}||^2 + \mu\sum_{j=1}^{p} \hat{\beta}_j^2$ (since the constant -μs does not affect the minimization).

For each value of s in the constrained problem, there is a corresponding value of μ in the unconstrained problem that yields the same solution.

(Refer Slide Time 26:00)



$$\min_{\beta} \left(y^T y - 2y^T H\beta + \beta^T H^T H \beta \right)$$

Of the form $\frac{1}{2} x^T A x + b^T x + c$, where $A = 2H^T H$, $b = -2H^T y$, $c = y^T y$.

Solution, $\hat{\beta} = -A^{-1} b = -\frac{1}{2}(H^T H)^{-1}(-2H^T y) = (H^T H)^{-1} H Y$.

Ridge Regression:

$$\boxed{\begin{aligned} &\min_{\hat{\beta}} \quad \sum_{i=1}^{n} ||Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_i^{(j)}||^2 \\ &\text{s.t.} \quad \hat{\beta}_1^2 + \hat{\beta}_2^2 + \dots + \hat{\beta}_p^2 \le \textcircled{s} \end{aligned}}$$

$$\mathcal{L}(\hat{\beta}, \mu) = \sum_{i=1}^{n} ||Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x^{(j)}||^2 + \mu \left(\sum_{j=1}^{p} \hat{\beta}_j^2 - s \right)$$

Say we know μ^*. Then the constrained problem is the same as

$$\boxed{\min_{\hat{\beta}} \left[\sum_{i=1}^{n} ||Y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x^{(j)}||^2 + \textcircled{\mu} \left(\sum_{j=1}^{p} \hat{\beta}_j^2 \right) \right]}$$

This is why ridge regression can be solved either as a constrained optimization problem or an unconstrained one. I chose ridge regression as an application because it allows us to cover both constrained and unconstrained optimization algorithms. In the next three lectures, we will take a publicly available dataset and solve the ridge regression problem, finding the vector $\beta = (\beta_0, \beta_1, ..., \beta_p)$ using both types of algorithms.

We will complete the course with that. Thank you.