https://www.cs.ubc.ca/~fwood/CS340/

Lecture XIII

February 3rd, 2020
https://www.cs.ubc.ca/~fwood/CS340/lectures/L13.pdf

Normal Equations

To find the normal equations, we set the gradient equal to 0, to find the critical points:

$$X^T Xw - X^T y = 0$$

We now move the terms not involving w to the other side:

$$X^T Xw = X^T y$$

This is a set of d linear equations, called the normal equations. In Python we sole linear equations in one line, using `numpy.linalg.solve`.

Least squares cost

We investigate the cost of solving normal equations $X^T Xw = X^T y$:

- Forming $X^T y$ vectors costs $O(nd)$
  - It has d elements, and each is an inner product between n numbers.
- Forming the matrix $X^T X$ costs $O(nd^2)$
  - It has d^2 elements, and each is an inner product between n numbers.
- Solving a $d \times d$ system of equations costs $O(d^3)$
  - Cost of Gaussian elimination on a d-variable linear system.
  - Other standard methods have the same cost
- Therefore the overall cost is $O(nd^2 + d^3)$.
  - The dominating term depends on n and d

Least squares issues

- Solution might not be unique.
- It is sensitive to outliers.
- It always uses all features.
- Data can might so big we can't store $X^T X$.
  - Or you can't afford the $O(nd^2 + d^3)$ cost.
- It might predict outside range of yi values.
- It assumes a linear relationship between xi and yi.

Non-Uniqueness of Least Squares Solution

Why isn't the solution unique?

- We investigate the case where we have two features that are identical for *all* examples.
- We can increase weight on one feature, and decrease it on the other, without changing predictions.

$$\hat{y}_i = w_1 x_{i1} + w_2 \underbrace{x_{i1}}_{\text{copy}} = (w_1 + w_2) x_{i1} + 0 x_{i1}$$

- Thus, if (w_1, w_2) is a solution then $(w_1 + w_2, 0)$ is another solution.
- This is special case of features being collinear:

But any w where $\nabla f(w) = 0$ is a global minimizer of f.