

Machine Learning (CSE 446): Learning as Minimizing Loss: Regularization and Gradient Descent

Sham M Kakade


© 2018

University of Washington
cse446-staff@cs.washington.edu

Announcements

- ▶ Assignment 2 due tomo.
- ▶ Midterm: Weds, Feb 7th.
- ▶ Qz section: review
- ▶ Today:
Regularization and Optimization!

one page of
handwritten notes



Review

Relax!

- ▶ The mis-classification optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y_n(\mathbf{w} \cdot \mathbf{x}_n) \leq 0]$$

- ▶ Instead, use loss function $\ell(y_n, \mathbf{w} \cdot \mathbf{x})$ and solve **relaxation**:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \ell(y_n, \mathbf{w} \cdot \mathbf{x}_n)$$

Relax!

- ▶ The mis-classification optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y_n(\mathbf{w} \cdot \mathbf{x}_n) \leq 0]$$

- ▶ Instead, use loss function $\ell(y_n, \mathbf{w} \cdot \mathbf{x})$ and solve **relaxation**:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \ell(y_n, \mathbf{w} \cdot \mathbf{x}_n)$$

- ▶ What do we want?
- ▶ How do we get it?
speed? accuracy?

Some loss functions:

- ▶ The square loss:

$$\ell(y, \mathbf{w} \cdot \mathbf{x}) = (y - \mathbf{w} \cdot \mathbf{x})^2$$

- ▶ The logistic loss:

~~$\ell^{\text{logistic}}(y, \mathbf{w} \cdot \mathbf{x}) = \log(1 + \exp(-y\mathbf{w} \cdot \mathbf{x})).$~~

holder
on

- ▶ They both ~~“upper bound”~~ the mistake rate.

$y \approx \mathbf{w} \cdot \mathbf{x}$

- ▶ Instead:

- ▶ Instead, we let's care about “regression” where y is real valued.
- ▶ What if we have multiple classes? (not just binary classification?)

Least squares: let's minimize it!

$$[Xw]_n = w \cdot x_n$$

\leftarrow N -vector

- ▶ The optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 =$$
$$\min_{\mathbf{w}} \|Y - X\mathbf{w}\|^2 \quad \cdot \frac{1}{N}$$

where Y is an N -vector and X is our $N \times d$ data matrix.

- ▶ The solution is the **least squares estimator**:

$$\mathbf{w}^{\text{least squares}} = (X^T X)^{-1} X^T Y$$

Matrix calculus proof: scratch space

$$\|Xw - Y\|^2 = (Xw - Y)^T (Xw - Y) = w^T X^T X w - 2Y^T X w + Y^T Y$$

$$\frac{\partial}{\partial w} () = 2X^T X w - 2X^T Y$$

want
w. s.t.

$$(X^T X) w = X^T Y = 0$$


$$Y \approx X w$$

Matrix calculus proof: scratch space

Let's remember our linear system solving!

$$2w_1 + 5w_2 = 8$$

$$3w_1 + 10w_2 = 11$$


$$\begin{bmatrix} 2 & 5 \\ 3 & 10 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 11 \end{bmatrix}$$

Today

Least squares: What could go wrong?!

- ▶ The optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 =$$
$$\min_{\mathbf{w}} \|Y - X\mathbf{w}\|^2$$

where Y is an n -vector and X is our $n \times d$ data matrix.

- ▶ The solution is the **least squares estimator**:

$$\mathbf{w}^{\text{least squares}} = (X^T X)^{-1} X^T Y$$

Least squares: What could go wrong?!

- ▶ The optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 =$$
$$\min_{\mathbf{w}} \|Y - X\mathbf{w}\|^2$$

where Y is an n -vector and X is our $n \times d$ data matrix.

- ▶ The solution is the **least squares estimator**:

$$\mathbf{w}^{\text{least squares}} = (X^T X)^{-1} X^T Y$$

What if d is bigger than n ? Even if not?

What could go wrong?

Suppose $d > n$:

$(X^T X)$ — not invertible

e.g. $\frac{\text{solve}}{5x_1 + 3x_2 = 11}$

What about $n > d$?

What could go wrong?

Suppose $d > n$:

What about $n > d$?

- ▶ What happens if features are very correlated?
(e.g. 'rows/columns' in our matrix are **co-linear**.)

rows/columns
exactly

linear system solving: scratch space

$$2w_1 + 3w_2 = 11$$

$$2x_1 + 3.0003x_2 = 1081$$

A fix: Regularization

- ▶ **Regularize** the optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|^2 =$$
$$\min_{\mathbf{w}} \|Y - X^T \mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

L₂-regularization

- ▶ This particular case: “Ridge” Regression, Tikhonov regularization
- ▶ The solution is the **least squares estimator**:

$$\mathbf{w}^{\text{least squares}} = \left(\frac{1}{N} X^T X + \lambda \mathbb{I} \right)^{-1} \left(\frac{1}{N} X^T Y \right)$$

dxd identity matrix

The “general” approach

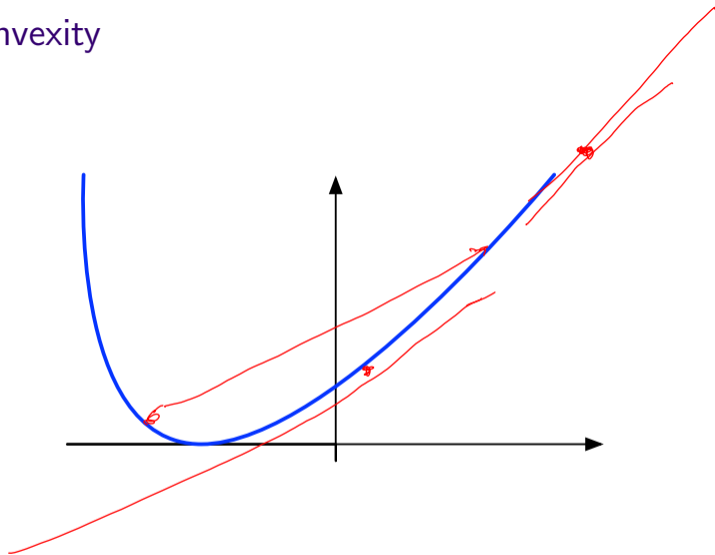
- ▶ The **regularized** optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \ell(y_n, \mathbf{w} \cdot \mathbf{x}_n) + R(\mathbf{w})$$

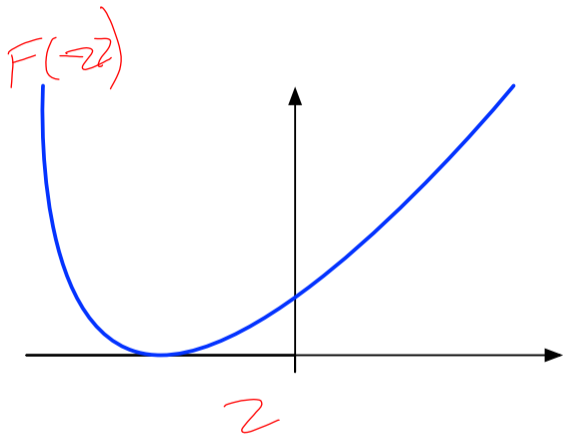
- ▶ Penalty some w more than others.
Example: $R(w) = \|w\|^2$

How do we find a solution quickly?

Remember: convexity



Gradient Descent



- ▶ Want to solve:

$$\min_z F(z)$$

- ▶ How should we update z ?

$$z \leftarrow z - \eta \nabla F(z)$$

↘
steps: z_0

Gradient Descent

Data: function $F : \mathbb{R}^d \rightarrow \mathbb{R}$, number of iterations K , step sizes $\langle \eta^{(1)}, \dots, \eta^{(K)} \rangle$

Result: $\mathbf{z} \in \mathbb{R}^d$

initialize: $\mathbf{z}^{(0)} = \mathbf{0}$;

for $k \in \{1, \dots, K\}$ **do**

$\mathbf{z}^{(k)} = \mathbf{z}^{(k-1)} - \eta^{(k)} \cdot \nabla_{\mathbf{z}} F(\mathbf{z}^{(k-1)})$;

end

return $\mathbf{z}^{(K)}$;

Algorithm 1: GRADIENTDESCENT

Gradient Descent: Convergence

- ▶ Letting $\mathbf{z}^* = \operatorname{argmin}_{\mathbf{z}} F(\mathbf{z})$ denote the global minimum
- ▶ Let $\mathbf{z}^{(k)}$ be our parameter after k updates.
- ▶ Thm: Suppose F is convex and “ L -smooth”. Using a **fixed step size** $\eta \leq \frac{1}{L}$, we have:

$$F(\mathbf{z}^{(k)}) - F(\mathbf{z}^*) \leq \frac{\|\mathbf{z}^{(0)} - \mathbf{z}^*\|^2}{\eta \cdot k}$$

That is the **convergence rate** is $O(\frac{1}{k})$.

Smoothness and Gradient Descent Convergence

- ▶ Smooth functions: for all z, z'

$$\|\nabla F(z) - \nabla F(z')\| \leq L\|z - z'\|$$

- ▶ Proof idea:

1. If our gradient is large, we will make good progress decreasing our function value:
2. If our gradient is small, we must have value near the optimal value: