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

Sham M Kakade

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

Announcements

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

Review

Relax!

- The mis-classification optimization problem:

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

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

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

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

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

hold on

$y \approx w \cdot x$

- ~~They both "upper bound" the mistake rate.~~
- 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!

$$[X w]_n = w \cdot X_n$$

N-vector

- The optimization problem:

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

$$\min_{\mathbf{w}} \|Y - X\mathbf{w}\|^2 \cdot \frac{1}{N}$$

where Y is an $\cancel{n}$ N-vector and X is our $\cancel{n}$ N $\times\, d$ data matrix.

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

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

Matrix calculus proof: scratch space

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

$$\frac{\partial}{\partial \vec{w}}\left(\quad\right) = 2X^T X w - 2X^T Y = 0$$

want w s.t. $(X^T X) w = X^T Y$

$$Y \approx Xw$$

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 $\cancel{n}N$-vector and X is our $\cancel{n}N \times d$ data matrix.

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

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

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 $\cancel{n}N$-vector and X is our $\cancel{n}N \times d$ data matrix.

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

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

What if d is bigger than $\cancel{n}N$? Even if not?

What could go wrong?

Suppose $d > \cancel{n}\,N$:

$(X^T X)$ – not invertible

e.g. solve

$5w_1 + 3w_2 = 11$

(under constrained system)

What about $\cancel{n}\,N > d$?

What could go wrong?

Suppose $d > \cancel{n}\,N$:

What about $\cancel{n}\,N > d$?

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

exactly

linear system solving: scratch space

$$2w_1 + 3w_2 = 11$$

$$3w_1 + 3.0003\,w_2 = 10.81$$

A fix: Regularization

- **Regularize** the optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_n - \mathbf{w} \cdot \mathbf{x}_n)^2 + \lambda \|\mathbf{w}\|_2^2 =$$

$$\min_{\mathbf{w}} \frac{1}{N} \|Y - X^\top \mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

L_2-regularization

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

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

dxd identity matrix

The "general" approach

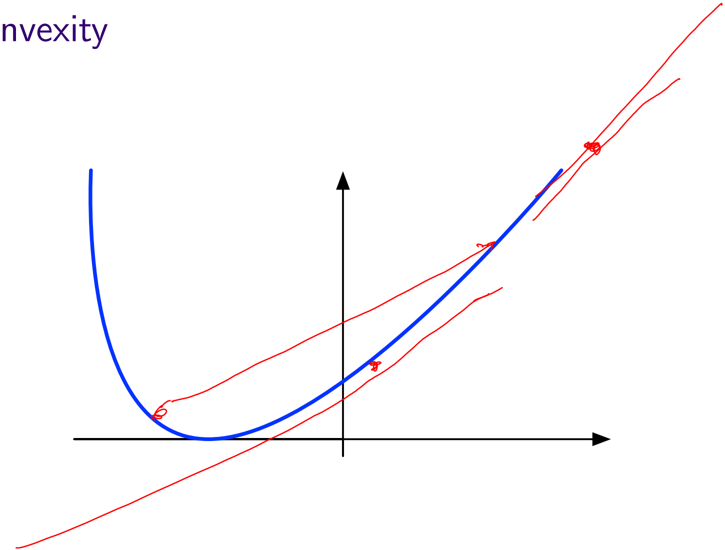
- The **regularized** optimization problem:

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

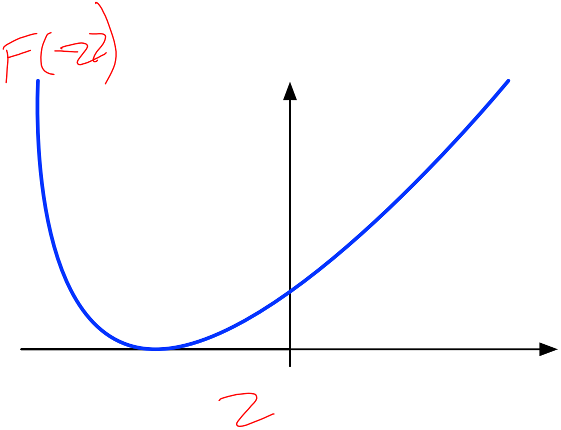
- Penalty some w more than others.
  Example: $R(w) = \|w\|^2$

How do we find a solution quickly?

Remember: convexity



Gradient Descent



- Want to solve:

$$\min_z F(z)$$

- How should we update z?

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

stepsize

Gradient Descent

Data: function $F : \mathbb{R}^d \to \mathbb{R}$, number of iterations K, step sizes $\langle \eta^{(1)}, \ldots, \eta^{(K)} \rangle$
Result: $\mathbf{z} \in \mathbb{R}^d$
initialize: $\mathbf{z}^{(0)} = \mathbf{0}$;
for $k \in \{1, \ldots, K\}$ **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: