Machine Learning (CSE 446): Regression and Regularization

Sham M Kakade

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

Announcements

- HW2 empirical problem for extra credit added
- milestone due tonight
- Fri will be 'tricks'/feature construction

Relax!

- The mis-classification optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y_i(\mathbf{w} \cdot \mathbf{x}_i) \leq 0\}$$

- Instead, let's try to choose a "reasonable" loss function $\ell(y_i, \mathbf{w} \cdot \mathbf{x})$ and then try to solve the **relaxation:**

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \mathbf{w} \cdot \mathbf{x}_i)$$

The square loss! (and linear regression)

- The square loss: $\ell(y, \mathbf{w} \cdot \mathbf{x}) = (y - \mathbf{w} \cdot \mathbf{x})^2$.
- The relaxed optimization problem:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2$$

- nice properties:
  - for binary classification, it is a an upper bound on the zero-one loss.
  - It makes sense more generally, e.g. if we want to predict real valued y.
  - We have a convex optimization problem.
- For classification, what is your decision rule using a $\mathbf{w}$?

Least squares: let's minimize it!

- The optimization problem:

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

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

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

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

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

- Let's give some hits on how to find this solution!

Vector calculus hints I

- suppose we have a function $f(w) = w \cdot c = w^\top c = c^\top w = \sum_j w[j]c[j]$, where w and c are d-dimensional vectors.
- Elementary calculus tells gives us scalar derivatives:

$$\frac{\partial f(w)}{\partial w[i]} = c[i]$$

- The *gradient* is the vector of all the partial derivatives:

$$\nabla f(w) := \left(\frac{\partial f(w)}{\partial w[1]}, \frac{\partial f(w)}{\partial w[2]}, \dots \frac{\partial f(w)}{\partial w[d]} \right)^\top$$

- So we have that:

$$\nabla f(w) = c$$

Vector calculus hints II

- suspose we have a function

  $$f(w) = w^\top M w = \sum_{j,k} w[j]w[k]M[j,k]\,,$$

  where M is a *symmetric* $d \times d$ matrix.
- Elementary calculus tells gives us scalar derivatives:

  $$\frac{\partial f(w)}{\partial w[i]} = 2\sum_j M[i,j]w[j]$$

- The *gradient* is just the matrix of all the partial derivatives:

  $$\nabla f(w) := \left(\frac{\partial f(w)}{\partial w[1]}, \frac{\partial f(w)}{\partial w[2]}, \ldots \frac{\partial f(w)}{\partial w[d]}\right)^\top$$

- It is straightforward to see that a far more compact way to write the gradient is:

  $$\nabla f(w) = 2Mw$$

  (just equate each coordinate with the scalar derivative).

Least squares derivation

- Using that $\|a\|^2 = a^\top a$,

$$\begin{aligned}\frac{1}{N}\|Y - X\mathbf{w}\|^2 &= \frac{1}{N}\left(Y^\top Y - Y^\top X\mathbf{w} - (X\mathbf{w})^\top Y + \mathbf{w}^\top X^\top X\mathbf{w}\right)\\ &= \frac{1}{N}\left(Y^\top Y - 2Y^\top X\mathbf{w} + \mathbf{w}^\top X^\top X\mathbf{w}\right)\end{aligned}$$

- Our optimization problem is then:

$$\min_{\mathbf{w}} \frac{1}{N}\left(Y^\top Y - 2Y^\top X\mathbf{w} + \mathbf{w}^\top X^\top X\mathbf{w}\right)$$

- Taking the derivative of the above (using our "hints") and setting it to 0 leads to: we want a $\mathbf{w}$ such that:

$$X^\top X\mathbf{w} = X^\top Y$$

- 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:

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

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

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

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

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

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**.)

A fix: Regularization

- **Regularize** the optimization problem:

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

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

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

The “general” approach

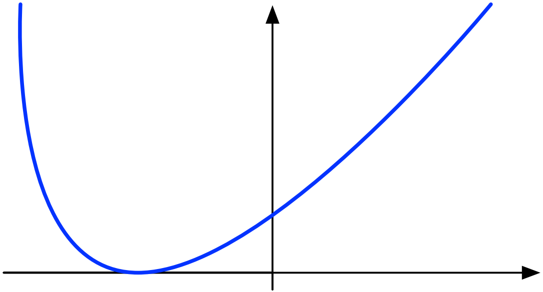
- The **regularized** optimization problem:

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

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

How do we find a solution quickly?

Gradient Descent (for a convex function)



- Want to solve:

$$\min_z F(z)$$

- How should we update z?

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