CSE 446 – Machine Learning

Taylor Blau

Maximum Likelihood Estimates

Given some model class parameterized by θ and some data $\mathcal{D}$, it is often desirable to find the parameter(s) θ with *maximum likelihood* given $\mathcal{D}$.

$$\begin{aligned}\hat{\theta}_{\text{MLE}} &= \arg\max_\theta \Pr(\mathcal{D} \mid \theta) \\ &= \arg\max_\theta \log \Pr(\mathcal{D} \mid \theta)\end{aligned}$$

For binomial random variables, recall that $f = \theta^k(1-\theta)^{n-k}$. Hoeffding's inequality states that:

$$\Pr(|\hat{\theta}_{\text{MLE}} - \theta^*| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

Note that not all maximum likelihood estimates are unbiased; in particular the MLE for the variance of a Gaussian random variable has non-zero bias.

Linear Regression

A generic *linear regression model* is as follows. For $X \in \mathbb{R}^{n\times d}$, with true labels $y \in \mathbb{R}$ (where $y_i = w^T x_i + \epsilon^i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$). We find a w such that the error is minimal.

$$\begin{aligned}\text{RSS}(w) &= \sum_{i=1}^n (y_i - x_i^T w)^2 \\ &= (y - Xw)^T(y - Xw) = ||y - Xw||_2^2\end{aligned}$$

$$\nabla_w ||y - Xw||_2^2 = -2X(y - Xw) = 0$$

Leading to the normal solution $X^T X w = X^T y$. If we instead suppose an intercept b (s.t., $y_i = x_i^T w + b + \epsilon_i$), then predict:

$$(X^T - \mu^T)\hat{w} + b, \quad \mu = \overline{x}, \widehat{b} = \overline{y}$$

Bias/variance trade-off

Define the error on the test set $\mathcal{T}$ to be:

$$\epsilon_{\text{test}} = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} L(y_i, f_{\hat{w}}(x_i))$$

Error comes from three sources: (1) noise, (2) bias, and (3) variance. Low-complexity models have high bias (and low variance), and vice-versa.

$$\begin{aligned}&\mathbb{E}_{\mathcal{T},y|x_t}[(y - f_{\hat{w}}(x_t))^2] \\ &= \mathbb{E}_{\mathcal{T},y|x_t}[(\underbrace{(y - f_{w^*}(x_t))}_{C} + \underbrace{(f_{w^*}(x_t) - f_{\hat{w}}(x_t))}_{D})^2]\end{aligned}$$

$$\mathbb{E}_{\mathcal{T},y|x_t}[C^2] = \sigma^2$$

$$\mathbb{E}_{\mathcal{T},y|x_t}[CD] = \mathbb{E}_{\mathcal{T}}[(f_{w^*}(x_t) - f_{\hat{w}}(x_t))\cancel{\mathbb{E}_{y|x_t}[y - f_{w^*}(x_t)]}]$$

$$\mathbb{E}_{\mathcal{T},y|x_t}[D^2] = \mathbb{E}_{\mathcal{T}}[(f_{w^*}(x_t) - f_{\hat{w}}(x_t))^2] = \text{MSE}[f_{\hat{w}}(x_t)]$$

Let $f_{\bar{w}} = \mathbb{E}_{\mathcal{T}}[f_{\hat{w}}(x_t)]$, and observe:

$$\begin{aligned}&\text{MSE}[f_{\hat{w}}(x_t)] = \mathbb{E}_{\mathcal{T}}[(f_{w^*}(x_t) - f_{\hat{w}}(x_t))^2] \\ &= \mathbb{E}_{\mathcal{T}}[(\underbrace{(f_{w^*}(x_t) - f_{\bar{w}}(x_t))}_{A} + \underbrace{(f_{\bar{w}}(x_t) - f_{\hat{w}}(x_t))}_{B})^2]\end{aligned}$$

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}[A^2] &= \mathbb{E}_{\mathcal{T}}[(f_{w^*}(x_t) - f_{\bar{w}}(x_t))^2] \\ &= (f_{w^*}(x_t) - f_{\bar{w}}(x_t))^2 = (\text{bias}(x_t))^2\end{aligned}$$

$$\mathbb{E}_{\mathcal{T}}[AB] = (f_{w^*}(x_t) - f_{\bar{w}}(x_t))\cancel{\mathbb{E}_{\mathcal{T}}[f_{\bar{w}}(x_t) - f_{\hat{w}}(x_t)]}$$

$$\mathbb{E}_{\mathcal{T}}[B^2] = \mathbb{E}_{\mathcal{T}}[(f_{\hat{w}}(x_t) - f_{\bar{w}}(x_t))^2] = \text{var}(f_{\hat{w}}(x_t))$$

In particular, we have:

- High-complexity functions have low bias and high variance.
- Low-complexity functions have high bias and low variance.

We also have that for fixed model complexity:

- Training sets containing one sample has no training error (i.e., pick a function through your single point) but high true error.
- As we add more points, training error goes up (i.e., bias is increasing while variance decreases) since your function (at some point) can no longer "pass through" all the points.
- This asymptotically approaches the value $\sigma +$ bias^2, since var $\to 0$ as the training set grows to encompass all values.

In summary we have that:

$$\text{bias}(x) = f_{w^*}(x) - \mathbb{E}_{\text{train}}[f_{\hat{w}}(x)]$$

or how much we deviate from the "true" f. Likewise,

$$\text{var}(x) = \mathbb{E}_{\text{train}}[(f_{\hat{w}}(x) - \mathbb{E}_{\text{train}}[f_{\hat{w}}(x)])^2]$$

or how much the function varies over different training sets.

ℓ_2-regularization

Recall the non-regularized least-squares objective:

$$\begin{aligned}\hat{w}_{\text{LS}} &= \arg\min_w \sum_{i=1}^n (y_i - x_i^T w)^2 \\ &= \arg\min_w (y - Xw)^T(y - Xw)\end{aligned}$$

When $(y - Xw)^{-1}$ exists, $\hat{w} = (X^T X)^{-1} X^T y$. But, if $d > n$, we have some flat directions and thus $X^T X$ is not full-rank and is non-invertible. Then, we "mix" with the ℓ_2-norm, and obtain the ridge regression objective:

$$\hat{w}_{\text{ridge}} = \arg\min_w \left(\sum_{i=1}^n (y_i - x_i^T w)^2\right) + \lambda||w||_2^2$$

$$\begin{aligned}0 &= \nabla_w \left(\left(\sum_{i=1}^n (y_i - x_i^T w)^2\right) + \lambda||w||_2^2\right) \\ &= -\sum_{i=1}^n 2x_i(y_i - x_i^T w) + 2\lambda w \\ &= -2\left(\sum_{i=1}^n x_i y_i\right) + 2\left(\sum_{i=1}^n x_i x_i^T + \lambda I\right) w\end{aligned}$$

Leading to:

$$\hat{w}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$$

Note that:

$$\lim_{\lambda\to 0} \hat{w}_{\text{ridge}} = \hat{w}_{\text{LS}}, \quad \lim_{\lambda\to\infty} \hat{w}_{\text{ridge}} = 0$$

One can see that:

- Setting λ close to 0 attains better training error, but does not generalize well (i.e., it has high training error due to over-fitting the training data).
- Setting λ large, we have both high training and testing error (due to under-fitting the training data, i.e., we didn't learn anything meaningful at all).

Note that λ is a hyper-parameter. We can choose an optimal λ based on k-fold cross-validation. Partition $\mathcal{T} = \mathcal{T}_1 \cup \cdots \cup \mathcal{T}_k$, and let:

$$\text{error}_{\mathcal{T}_i} = \frac{1}{|\mathcal{T}_i|} \sum_{(x_j,y_j)\in\mathcal{T}_I} (y_j - f_{\mathcal{T}\setminus\mathcal{T}_i}(x_j))^2$$

And the k-fold error as:

$$\text{error}_{k\text{-fold}} = \frac{1}{k}\sum_{i=1}^k \text{error}_{\mathcal{T}_i}$$

Typically k-fold CV is preferred to LOOCV, since the former is tractable and the later is often not.

ℓ_1-regularization, LASSO

Note that for d large, we would prefer *sparse* solutions. These are efficient to compute, and often lead to more interpretable solutions. ℓ_2-thresholding can increase weight on correlated features when one of the correlates is removed. Instead, we prefer solutions with low ℓ_1-

norm.

$$\hat{w}_{\text{LASSO}} = \arg\min_w \sum_{i=1}^n (y_i - x_i^T w)^2 + \lambda ||w||_1$$

We can optimize this naïvely using coordinate descent. Until convergence, cycle through coordinates j, doing:

$$\hat{w}_j = \begin{cases} (c_j + \lambda)/a_j & \text{if } c_j < -\lambda \\ 0 & \text{if } |c_j| \leq \lambda \\ (c_j - \lambda)/a_j & \text{if } c_j > \lambda \end{cases}$$

Where:

$$a_j = \sum_{i=1}^n x_{i,j}^2, \quad c_j = 2\sum_{i=1}^n \left(y_i - \sum_{k \neq j} x_{i,k} w_k \right) x_{i,j}$$

LASSO leads to solutions that have:

- Coordinates that are *exactly* zero-valued.
- Solutions that have greater sparsity than low ℓ_p-norm solutions ($p > 1$).
- Solutions with higher *interpretability*.

Logistic Regression

Suppose now instead we want to learn a function $f : \mathbb{R}^d \to \mathcal{K}$ where each $k \in \mathcal{K}$ is some *feature class*. Define the logistic loss of f to be:

$$\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$$

Compute the expected loss as:

$$\begin{aligned} \mathbb{E}_{XY}[\mathbf{1}\{f(X) \neq Y\}] &= \mathbb{E}_X[\mathbb{E}_{Y|X}[\mathbf{1}\{f(x) \neq Y\} \mid X = x]] \\ &= \textstyle\sum_i \Pr(Y = i \mid X = x)\mathbf{1}\{f(x) \neq i\} \\ &= \textstyle\sum_{f(x) \neq i} \Pr(Y = i \mid X = x) \\ &= 1 - \Pr(Y = f(x) \mid X = x) \end{aligned}$$

Observe the Bayes-optimal classifier is:

$$f(x) = \arg\max_y \Pr(Y = y \mid X = x)$$

Define the Sigmoid function as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}$$

Then, for binary $\mathcal{K}$, we have:

$$\begin{aligned} \Pr(Y = 1 \mid X = x, w) = \sigma(w^T x) &= \frac{1}{1 + \exp(-w^T x)} \\ \Pr(Y = 0 \mid X = x, w) = 1 - \sigma(w^T x) &= \frac{\exp(-w^T x)}{1 + \exp(-w^T x)} \\ &= \frac{1}{1 + \exp(w^T x)} \end{aligned}$$

So our decision rule becomes:

$$\frac{\Pr(Y = 1 \mid x, w)}{\Pr(Y = 0 \mid x, w)} = \exp(w^T x) \underset{Y=0}{\overset{Y=1}{\gtrless}} 1$$

Switching our classes to $\{-1, 1\}$, we have:

$$\begin{aligned} \hat{w}_{\text{MLE}} &= \arg\max_w \prod_{i=1}^n \Pr(y_i \mid x_i, w) \\ &= \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i x_i^T w) \end{aligned}$$

Here we're computing the maximum likelihood estimate of our prediction being correct against the true label y_i. This is *log-loss* objective function, which penalizes incorrect guesses smoothly.

Adding an offset term and applying ℓ_2 regularization, we have:

$$\hat{w}_{\text{MLE}} = \arg\min_{w,b} \sum_{i=1}^n \log(1 + \exp(-y_i(x_i^T w + b)) + \lambda ||w||_2^2$$

Note that for x such that $w^T x + b = 0$, we guess that x has equal probability of either class (i.e., that $\Pr(Y = \pm 1 \mid x, w) = 1/2$).

Be sure that the normal $\hat{w}$ "points towards" the positive class points.

Gradient Descent, Stochastic Methods

Note that we can locate the minimum of a convex function using gradient descent with an appropriate step size η as follows:

$$w^{(t+1)} = w^{(t)} - \eta \nabla_w \left(\frac{1}{n} \sum_{i=1}^n \ell_i(w) \right) \Bigg|_{w=w^{(t)}}$$

This process can take time proportional to the size of $\mathcal{T}$, N. We can instead approximate the empirical risk by some batch I_t, and obtain *stochastic gradient descent*:

$$w^{(t+1)} = w^{(t)} - \eta \nabla_w \, \ell_{I_t}(w)|_{w=w^{(t)}}$$

In practice, stochastic gradient descent with a reasonable batch size can attain a convergence rate higher than gradient descent. With lower batch sizes, the path is less smooth.

Convexity

Recall that a function f is convex on $\mathcal{A}$ if for $x_1, x_2 \in \mathcal{A}$ and $\lambda \in [0, 1]$, that:

$$f(\lambda x_1 + (1 - \lambda) x_2) \leq \lambda f(x_1) + (1 - \lambda) f(x_2)$$

The ℓ_p-norm is convex for all p and is defined as:

$$\ell_p(x) = \left(\sum_{i=1}^d (|x_i|)^p \right)^{1/p}$$

Where ℓ_∞ is given as:

$$\ell_\infty(x) = \lim_{p \to \infty} \ell_p(x) = \max_{i \in [d]} |x_i|$$

Linear Algebra Cookbook

Some common forms of partial derivatives on vectors/matrices include:

$$\begin{aligned} \frac{\partial a^T x}{\partial x} = \frac{\partial x^T a}{\partial x} &= a \\ \frac{\partial x^T A x}{\partial x} &= (A + A^T) x \\ \frac{\partial a^T X b}{\partial X} = a b^T, \quad \frac{\partial a^T X^T b}{\partial X} &= b a^T \\ \frac{\partial a^T X a}{\partial X} = \frac{\partial a^T X^T a}{\partial X} &= a a^T \\ \frac{\partial x^T A}{\partial x} = A \frac{\partial x^T}{\partial x} &= I \end{aligned}$$