# Some Calculations from Bias Variance

Christopher Ré

May 7, 2019

This note contains a reprise of the eigenvalue arguments to understand how variance is reduced by regularization. We also ¿ describe different ways regularization can occur including from the ¿ algorithm or initialization. This note contains some additional calculations from the lecture and Piazza, just so that we have typeset versions of them. They contain **no** new information over the lecture, but they do supplement the notes.

Recall we have a design matrix $X \in \mathbb{R}^{n \times d}$ and labels $y \in \mathbb{R}^n$. We are interested in the underdetermined case $n < d$ so that $\text{rank}(X) \leq n < d$. We consider the following optimization problem for least squares with a regularization parameter $\lambda \geq 0$:

$$\ell(\theta; \lambda) = \min_{\theta \in \mathbb{R}^d} \frac{1}{2}\|X\theta - y\|^2 + \frac{\lambda}{2}\|\theta\|^2$$

**Normal Equations** Computing derivatives as we did for the normal equations, we see that:

$$\nabla_\theta \ell(\theta; \lambda) = X^T(X\theta - y) + \lambda\theta = (X^TX + \lambda I)\theta - X^Ty$$

By setting $\nabla_\theta \ell(\theta, \lambda) = 0$ we can solve for the $\hat{\theta}$ that minimizes the above problem. Explicitly, we have:

$$\hat{\theta} = \left(X^TX + \lambda I\right)^{-1} X^Ty \tag{1}$$

To see that the inverse in Eq. 1 exists, we observe that $X^TX$ is a symmetric, real $d \times d$ matrix so it has $d$ eigenvalues (some may be 0). Moreover, it is positive semidefinite, and we capture this by writing $\text{eig}(X^TX) = \{\sigma_1^2, \ldots, \sigma_d^2\}$. Now, inspired by the regularized problem, we examine:

$$\text{eig}(X^TX + \lambda I) = \left\{\sigma_1^2 + \lambda, \ldots, \sigma_d^2 + \lambda\right\}$$

Since $\sigma_i^2 \geq 0$ for all $i \in [d]$, if we set $\lambda > 0$ then $X^TX + \lambda I$ is full rank, and the inverse of $(X^TX + \lambda I)$ exists. In turn, this means there is a unique such $\hat{\theta}$.

**Variance** Recall that in bias-variance, we are concerned with the variance of $\hat{\theta}$ as we sample the training set. We want to argue that as the regularization parameter $\lambda$ increases, the variance in the fitted $\hat{\theta}$ decreases. We won't carry

out the full formal argument, but it suffices to make one observation that is immediate from Eq. 1: *the variance of $\hat{\theta}$ is proportional to the eigenvalues of* $(X^TX + \lambda I)^{-1}$. To see this, observe that the eigenvalues of an inverse are just the inverse of the eigenvalues:

$$\text{eig}\left(\left(X^TX + \lambda I\right)^{-1}\right) = \left\{\frac{1}{\sigma_1^2 + \lambda}, \ \ldots, \ \frac{1}{\sigma_d^2 + \lambda}\right\}$$

Now, condition on the points we draw, namely $X$. Then, recall that randomness is in the label noise (recall the linear regression model $y \sim X\theta^* + \mathcal{N}(0, \tau^2 I) = \mathcal{N}(X\theta^*, \tau^2 I)$).

Recall a fact about the multivariate normal distribution:

$$\text{if } y \sim \mathcal{N}(\mu, \Sigma) \text{ then } Ay \sim \mathcal{N}(A\mu, A\Sigma A^T)$$

Using linearity, we can verify that the expectation of $\hat{\theta}$ is

$$\begin{aligned}
\mathbb{E}[\hat{\theta}] &= \mathbb{E}[(X^TX + \lambda I)^{-1}X^Ty] \\
&= \mathbb{E}[(X^TX + \lambda I)^{-1}X^T(X\theta^* + \mathcal{N}(0, \tau^2, I))] \\
&= \mathbb{E}[(X^TX + \lambda I)^{-1}X^T(X\theta^*)] \\
&= (X^TX + \lambda I)^{-1}(X^TX)\theta^* \quad \text{(essentially a ``shrunk'' } \theta^*\text{)}
\end{aligned}$$

The last line above suggests that the more regularization we add (larger the $\lambda$), the more the estimated $\hat{\theta}$ will be shrunk towards 0. In other words, regularization adds bias (towards zero in this case). Though we paid the cost of higher bias, we gain by reducing the variance of $\hat{\theta}$. To see this bias-variance tradeoff concretely, observe the covariance matrix of $\hat{\theta}$:

$$\begin{aligned}
C &:= \text{Cov}[\hat{\theta}] \\
&= \left((X^TX + \lambda I)^{-1}X^T\right)(\tau^2 I)\left(X(X^TX + \lambda I)^{-1}\right)
\end{aligned}$$

and

$$\text{eig}(C) = \left\{\frac{\tau^2\sigma_1^2}{(\sigma_1^2 + \lambda)^2}, \ \ldots, \ \frac{\tau^2\sigma_d^2}{(\sigma_d^2 + \lambda)^2}\right\}$$

Notice that the entire spectrum of the covariance is a *decreasing* function of $\lambda$. By decomposing in the eigenvalue basis, we can see that actually $\mathbb{E}[\|\hat{\theta} - \theta^*\|^2]$ is a decreasing function of $\lambda$, as desired.

**Gradient Descent** We show that you can initialize gradient descent in a way that effectively regularizes undetermined least squares–even with no regularization penalty ($\lambda = 0$). Our first observation is that any point $x \in \mathbb{R}^d$ can be decomposed into two orthogonal components $x_0, x_1$ such that

$$x = x_0 + x_1 \text{ and } x_0 \in \mathsf{Null}(X^T) \text{ and } \mathsf{Range}(X).$$

Recall that $\mathsf{Null}(X^T)$ and $\mathsf{Range}(X)$ are orthogonal subspaces by the fundamental theory of linear algebra. We write $P_0$ for the projection on the null and $P_1$ for the projection on the range, then $x_0 = P_0(x)$ and $x_1 = P_1(x)$.

If one initializes at a point $\theta$ then, we observe that the gradient is orthogonal to the null space. That is, if $g(\theta) = X^T(X\theta - y)$ then $g^T P_0(v) = 0$ for any $v \in \mathbb{R}^d$. But, then:

$$P_0(\theta^{(t+1)}) = P_0(\theta^t - \alpha g(\theta^{(t)})) = P_0(\theta^t) - \alpha P_0 g(\theta^{(t)}) = P_0(\theta^{(t)})$$

That is, no learning happens in the null. Whatever portion is in the null that we initialize stays there throughout execution.

A key property of the Moore-Penrose pseudoinverse, is that if $\hat{\theta} = (X^T X)^+ X^T y$ then $P_0(\hat{\theta}) = 0$. Hence, the gradient descent solution initialized at $\theta_0$ can be written $\hat{\theta} + P_0(\theta_0)$. Two immediate observations:

- Using the Moore-Penrose inverse acts as regularization, because it selects the solution $\hat{\theta}$.
- So does gradient descent–provided that we initialize at $\theta_0 = 0$. This is particularly interesting, as many modern machine learning techniques operate in these underdetermined regimes.

We've argued that there are many ways to find equivalent solutions, and that this allows us to understand the effect on the model fitting procedure as regularization. Thus, there are many ways to find that equivalent solution. Many modern methods of machine learning including dropout and data augmentation are not penalty, but their effect is understood as regularization. One contrast with the above methods is that they often depend on some property of the data or for how much they effectively regularization. In some sense, they adapt to the data. A final comment is that in the same sense above, adding more data regularizes the model as well!