A Comprehensive Guide to Machine Learning

Soroush Nasiriany, Garrett Thomas, William Wang, Alex Yang, Jennifer Listgarten, Anant Sahai
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

November 18, 2019

About

CS 189 is the Machine Learning course at UC Berkeley. In this guide we have created a comprehensive course guide in order to share our knowledge with students and the general public, and hopefully draw the interest of students from other universities to Berkeley's Machine Learning curriculum.

This guide was started by CS 189 TAs Soroush Nasiriany and Garrett Thomas in Fall 2017, with the assistance of William Wang and Alex Yang.

We owe gratitude to Professors Anant Sahai, Stella Yu, and Jennifer Listgarten, as this book is heavily inspired from their lectures. In addition, we are indebted to Professor Jonathan Shewchuk for his machine learning notes, from which we drew inspiration.

The latest version of this document can be found either at `http://www.eecs189.org/` or `http://snasiriany.me/cs189/`. Please report any mistakes to the staff, and contact the authors if you wish to redistribute this document.

Notation

Notation	Meaning
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set (vector space) of n-tuples of real numbers, endowed with the usual inner product
$\mathbb{R}^{m\times n}$	set (vector space) of m-by-n matrices
δ_{ij}	Kronecker delta, i.e. $\delta_{ij} = 1$ if $i = j$, 0 otherwise
$\nabla f(\mathbf{x})$	gradient of the function f at $\mathbf{x}$
$\nabla^2 f(\mathbf{x})$	Hessian of the function f at $\mathbf{x}$
$p(X)$	distribution of random variable X
$p(x)$	probability density/mass function evaluated at x
$\mathbb{E}[X]$	expected value of random variable X
$\text{Var}(X)$	variance of random variable X
$\text{Cov}(X, Y)$	covariance of random variables X and Y

Other notes:

- Vectors and matrices are in bold (e.g. $\mathbf{x}, \mathbf{A}$). This is true for vectors in $\mathbb{R}^n$ as well as for vectors in general vector spaces. We generally use Greek letters for scalars and capital Roman letters for matrices and random variables.

- We assume that vectors are column vectors, i.e. that a vector in $\mathbb{R}^n$ can be interpreted as an n-by-1 matrix. As such, taking the transpose of a vector is well-defined (and produces a row vector, which is a 1-by-n matrix).

Contents

Chapter 1

Regression I

Our goal in machine learning is to extract a *relationship* from data. In **regression** tasks, this relationship takes the form of a function $y = f(\mathbf{x})$, where $y \in \mathbb{R}$ is some quantity that can be predicted from an input $\mathbf{x} \in \mathbb{R}^d$, which should for the time being be thought of as some collection of numerical measurements. The true relationship f is unknown to us, and our aim is to recover it as well as we can from data. Our end product is a function $\hat{y} = h(\mathbf{x})$, called the **hypothesis**, that should approximate f. We assume that we have access to a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where each pair $(\mathbf{x}_i, y_i)$ is an example (possibly noisy or otherwise approximate) of the input-output mapping to be learned. Since learning arbitrary functions is intractable, we restrict ourselves to some **hypothesis class** $\mathcal{H}$ of allowable functions. More specifically, we typically employ a **parametric model**, meaning that there is some finite-dimensional vector $\mathbf{w} \in \mathbb{R}^d$, the elements of which are known as **parameters** or **weights**, that controls the behavior of the function. That is,

$$h_{\mathbf{w}}(\mathbf{x}) = g(\mathbf{x}, \mathbf{w})$$

for some other function g. The hypothesis class is then the set of all functions induced by the possible choices of the parameters $\mathbf{w}$:

$$\mathcal{H} = \{h_{\mathbf{w}} \mid \mathbf{w} \in \mathbb{R}^d\}$$

After designating a **cost function** L, which measures how poorly the predictions $\hat{y}$ of the hypothesis match the true output y, we can proceed to search for the parameters that best fit the data by minimizing this function:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w})$$

1.1 Ordinary Least Squares

Ordinary least squares (OLS) is one of the simplest regression problems, but it is well-understood and practically useful. It is a **linear regression** problem, which means that we take $h_{\mathbf{w}}$ to be of the form $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}$. We want

$$y_i \approx \hat{y}_i = h_{\mathbf{w}}(\mathbf{x}_i) = {\mathbf{x}_i}^\top \mathbf{w}$$

for each $i = 1, \dots, n$. This set of equations can be written in matrix form as

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} \approx \underbrace{\begin{bmatrix} \mathbf{x}_1{}^\top \\ \vdots \\ \mathbf{x}_n{}^\top \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}}_{\mathbf{w}}$$

In words, the matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has the input datapoint $\mathbf{x}_i$ as its ith row. This matrix is sometimes called the **design matrix**. Usually $n \geq d$, meaning that there are more datapoints than measurements.

There will in general be no exact solution to the equation $\mathbf{y} = \mathbf{X}\mathbf{w}$ (even if the data were perfect, consider how many equations and variables there are), but we can find an approximate solution by minimizing the sum (or equivalently, the mean) of the squared errors:

$$L(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{x}_i{}^\top \mathbf{w} - y_i)^2 = \min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

Now that we have formulated an optimization problem, we want to go about solving it. We will see that the particular structure of OLS allows us to compute a closed-form expression for a globally optimal solution, which we denote $\mathbf{w}^*_{\text{OLS}}$.

Approach 1: Vector calculus

Calculus is the primary mathematical workhorse for studying the optimization of differentiable functions. Recall the following important result: if $L : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable, then any local optimum $\mathbf{w}^*$ satisfies $\nabla L(\mathbf{w}^*) = \mathbf{0}$. In the OLS case,

$$\begin{aligned} L(\mathbf{w}) &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \\ &= (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= (\mathbf{X}\mathbf{w})^\top \mathbf{X}\mathbf{w} - (\mathbf{X}\mathbf{w})^\top \mathbf{y} - \mathbf{y}^\top \mathbf{X}\mathbf{w} + \mathbf{y}^\top \mathbf{y} \\ &= \mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y} \end{aligned}$$

Using the following results from matrix calculus

$$\begin{aligned} \nabla_{\mathbf{x}}(\mathbf{a}^\top \mathbf{x}) &= \mathbf{a} \\ \nabla_{\mathbf{x}}(\mathbf{x}^\top \mathbf{A}\mathbf{x}) &= (\mathbf{A} + \mathbf{A}^\top)\mathbf{x} \end{aligned}$$

the gradient of L is easily seen to be

$$\begin{aligned} \nabla L(\mathbf{w}) &= \nabla_{\mathbf{w}}(\mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}) \\ &= \nabla_{\mathbf{w}}(\mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w}) - 2\nabla_{\mathbf{w}}(\mathbf{w}^\top \mathbf{X}^\top \mathbf{y}) + \underbrace{\nabla_{\mathbf{w}}(\mathbf{y}^\top \mathbf{y})}_{\mathbf{0}} \\ &= 2\mathbf{X}^\top \mathbf{X}\mathbf{w} - 2\mathbf{X}^\top \mathbf{y} \end{aligned}$$

where in the last line we have used the symmetry of $\mathbf{X}^\top \mathbf{X}$ to simplify $\mathbf{X}^\top \mathbf{X} + (\mathbf{X}^\top \mathbf{X})^\top = 2\mathbf{X}^\top \mathbf{X}$. Setting the gradient to $\mathbf{0}$, we conclude that any optimum $\mathbf{w}^*_{\text{OLS}}$ satisfies

$$\mathbf{X}^\top \mathbf{X}\mathbf{w}^*_{\text{OLS}} = \mathbf{X}^\top \mathbf{y}$$

If $\mathbf{X}$ is full rank, then $\mathbf{X}^\top\mathbf{X}$ is as well (assuming $n \geq d$), so we can solve for a unique solution

$$\mathbf{w}^*_{\text{OLS}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Note: Although we write $(\mathbf{X}^\top\mathbf{X})^{-1}$, in practice one would not actually compute the inverse; it is more numerically stable to solve the linear system of equations above (e.g. with Gaussian elimination).

In this derivation we have used the condition $\nabla L(\mathbf{w}^*) = \mathbf{0}$, which is a *necessary* but not *sufficient* condition for optimality. We found a critical point, but in general such a point could be a local minimum, a local maximum, or a saddle point. Fortunately, in this case the objective function is **convex**, which implies that any critical point is indeed a global minimum. To show that L is convex, it suffices to compute the **Hessian** of L, which in this case is

$$\nabla^2 L(\mathbf{w}) = 2\mathbf{X}^\top\mathbf{X}$$

and show that this is positive semi-definite:

$$\forall \mathbf{w},\ \mathbf{w}^\top(2\mathbf{X}^\top\mathbf{X})\mathbf{w} = 2(\mathbf{X}\mathbf{w})^\top\mathbf{X}\mathbf{w} = 2\|\mathbf{X}\mathbf{w}\|_2^2 \geq 0$$

Approach 2: Orthogonal projection

There is also a linear algebraic way to arrive at the same solution: orthogonal projections.

Recall that if V is an inner product space and S a subspace of V, then any $\mathbf{v} \in V$ can be decomposed uniquely in the form

$$\mathbf{v} = \mathbf{v}_S + \mathbf{v}_\perp$$

where $\mathbf{v}_S \in S$ and $\mathbf{v}_\perp \in S^\perp$. Here $S^\perp$ is the orthogonal complement of S, i.e. the set of vectors that are perpendicular to every vector in S.

The **orthogonal projection** onto S, denoted P_S, is the linear operator that maps $\mathbf{v}$ to $\mathbf{v}_S$ in the decomposition above. An important property of the orthogonal projection is that

$$\|\mathbf{v} - P_S\mathbf{v}\| \leq \|\mathbf{v} - \mathbf{s}\|$$

for all $\mathbf{s} \in S$, with equality if and only if $\mathbf{s} = P_s\mathbf{v}$. That is,

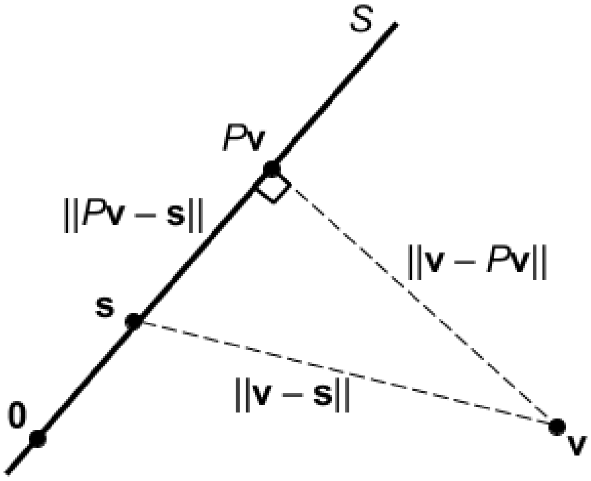
$$P_S\mathbf{v} = \operatorname*{arg\,min}_{\mathbf{s} \in S} \|\mathbf{v} - \mathbf{s}\|$$

Proof. By the Pythagorean theorem,

$$\|\mathbf{v} - \mathbf{s}\|^2 = \|\underbrace{\mathbf{v} - P_S\mathbf{v}}_{\in S^\perp} + \underbrace{P_S\mathbf{v} - \mathbf{s}}_{\in S}\|^2 = \|\mathbf{v} - P_S\mathbf{v}\|^2 + \|P_S\mathbf{v} - \mathbf{s}\|^2 \geq \|\mathbf{v} - P_S\mathbf{v}\|^2$$

with equality holding if and only if $\|P_S\mathbf{v} - \mathbf{s}\|^2 = 0$, i.e. $\mathbf{s} = P_S\mathbf{v}$. Taking square roots on both sides gives $\|\mathbf{v} - \mathbf{s}\| \geq \|\mathbf{v} - P_S\mathbf{v}\|$ as claimed (since norms are nonnegative). □

Here is a visual representation of the argument above:



In the OLS case,

$$\mathbf{w}^*_{\text{OLS}} = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

But observe that the set of vectors that can be written $\mathbf{X}\mathbf{w}$ for some $\mathbf{w} \in \mathbb{R}^d$ is precisely the range of $\mathbf{X}$, which we know to be a subspace of $\mathbb{R}^n$, so

$$\min_{\mathbf{z}\in\text{range}(\mathbf{X})} \|\mathbf{z} - \mathbf{y}\|_2^2 = \min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

By pattern matching with the earlier optimality statement about P_S, we observe that $P_{\text{range}(\mathbf{X})}\mathbf{y} = \mathbf{X}\mathbf{w}^*_{\text{OLS}}$, where $\mathbf{w}^*_{\text{OLS}}$ is any optimum for the right-hand side. The projected point $\mathbf{X}\mathbf{w}^*_{\text{OLS}}$ is always unique, but if $\mathbf{X}$ is full rank (again assuming $n \geq d$), then the optimum $\mathbf{w}^*_{\text{OLS}}$ is also unique (as expected). This is because $\mathbf{X}$ being full rank means that the columns of $\mathbf{X}$ are linearly independent, in which case there is a one-to-one correspondence between $\mathbf{w}$ and $\mathbf{X}\mathbf{w}$.

To solve for $\mathbf{w}^*_{\text{OLS}}$, we need the following fact[1]:

$$\text{null}(\mathbf{X}^\top) = \text{range}(\mathbf{X})^\perp$$

Since we are projecting onto range($\mathbf{X}$), the orthogonality condition for optimality is that $\mathbf{y} - P\mathbf{y} \perp \text{range}(\mathbf{X})$, i.e. $\mathbf{y} - \mathbf{X}\mathbf{w}^*_{\text{OLS}} \in \text{null}(\mathbf{X}^\top)$. This leads to the equation

$$\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\mathbf{w}^*_{\text{OLS}}) = \mathbf{0}$$

which is equivalent to

$$\mathbf{X}^\top\mathbf{X}\mathbf{w}^*_{\text{OLS}} = \mathbf{X}^\top\mathbf{y}$$

as before.

1.2 Ridge Regression

While Ordinary Least Squares can be used for solving linear least squares problems, it falls short due to numerical instability and generalization issues. Numerical instability arises when the features of the data are close to collinear (leading to linearly dependent feature columns), causing the input

[1] This result is often stated as part of the Fundamental Theorem of Linear Algebra.

matrix $\mathbf{X}$ to lose its rank or have singular values that very close to 0. Why are small singular values bad? Let us illustrate this via the singular value decomposition (SVD) of $\mathbf{X}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}, \mathbf{\Sigma} \in \mathbb{R}^{n\times d}, \mathbf{V} \in \mathbb{R}^{d\times d}$. In the context of OLS, we must have that $\mathbf{X}^\top\mathbf{X}$ is invertible, or equivalently, $\text{rank}(\mathbf{X}^\top\mathbf{X}) = \text{rank}(\mathbf{X}^\top) = \text{rank}(\mathbf{X}) = d$. Assuming that $\mathbf{X}$ and $\mathbf{X}^\top$ are full column rank d, we can express the SVD of $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_d \\ \mathbf{0}\end{bmatrix}\mathbf{V}^\top$$

where $\mathbf{\Sigma}_d \in \mathbb{R}^{d\times d}$ is a diagonal matrix with strictly positive entries. Now let's try to expand the $(\mathbf{X}^\top\mathbf{X})^{-1}$ term in OLS using the SVD of $\mathbf{X}$:

$$\begin{aligned}
(\mathbf{X}^\top\mathbf{X})^{-1} &= (\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_d & \mathbf{0}\end{bmatrix}\mathbf{U}^\top\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_d \\ \mathbf{0}\end{bmatrix}\mathbf{V}^\top)^{-1} \\
&= (\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_d & \mathbf{0}\end{bmatrix}\mathbf{I}\begin{bmatrix}\mathbf{\Sigma}_d \\ \mathbf{0}\end{bmatrix}\mathbf{V}^\top)^{-1} \\
&= (\mathbf{V}\mathbf{\Sigma}_d^2\mathbf{V}^\top)^{-1} = (\mathbf{V}^\top)^{-1}(\mathbf{\Sigma}_d^2)^{-1}\mathbf{V}^{-1} = \mathbf{V}\mathbf{\Sigma}_d^{-2}\mathbf{V}^\top
\end{aligned}$$

This means that $(\mathbf{X}^\top\mathbf{X})^{-1}$ will have singular values that are the squared inverse of the singular values of $\mathbf{X}$, potentially leading to extremely large singular values when the singular value of $\mathbf{X}$ are close to 0. Such excessively large singular values can be very problematic for numerical stability purposes. In addition, abnormally high values to the optimal $\mathbf{w}$ solution would prevent OLS from generalizing to unseen data.

There is a very simple solution to these issues: penalize the entries of $\mathbf{w}$ from becoming too large. We can do this by adding a penalty term constraining the norm of $\mathbf{w}$. For a fixed, small scalar $\lambda > 0$, we now have:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

Note that the λ in our objective function is a **hyperparameter** that measures the sensitivity to the values in $\mathbf{w}$. Just like the degree in polynomial features, λ is a value that we must choose arbitrarily through validation. Let's expand the terms of the objective function:

$$\begin{aligned}
L(\mathbf{w}) &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \\
&= \mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} - 2\mathbf{w}^\top\mathbf{X}^\top\mathbf{y} + \mathbf{y}^\top\mathbf{y} + \lambda\mathbf{w}^\top\mathbf{w}
\end{aligned}$$

Finally take the gradient of the objective and find the value of $\mathbf{w}$ that achieves $\mathbf{0}$ for the gradient:

$$\begin{aligned}
\nabla_{\mathbf{w}} L(\mathbf{w}) &= \mathbf{0} \\
2\mathbf{X}^\top\mathbf{X}\mathbf{w} - 2\mathbf{X}^\top\mathbf{y} + 2\lambda\mathbf{w} &= \mathbf{0} \\
(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w} &= \mathbf{X}^\top\mathbf{y} \\
\mathbf{w}^*_{\text{RIDGE}} &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}$$

This value is guaranteed to achieve the (unique) global minimum, because the objective function is **strongly convex**. To show that f is strongly convex, it suffices to compute the Hessian of f, which in this case is

$$\nabla^2 L(\mathbf{w}) = 2\mathbf{X}^\top\mathbf{X} + 2\lambda\mathbf{I}$$

and show that this is **positive definite (PD)**:

$$\forall \mathbf{w} \neq \mathbf{0},\ \mathbf{w}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = (\mathbf{X}\mathbf{w})^\top\mathbf{X}\mathbf{w} + \lambda\mathbf{w}^\top\mathbf{w} = \|\mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 > 0$$

Since the Hessian is positive definite, we can equivalently say that the eigenvalues of the Hessian are strictly positive and that the objective function is strongly convex. A useful property of strongly convex functions is that they have a unique optimum point, so the solution to ridge regression is unique. We cannot make such guarantees about ordinary least squares, because the corresponding Hessian could have eigenvalues that are 0. Let us explore the case in OLS when the Hessian has a 0 eigenvalue. In this context, the term $\mathbf{X}^\top\mathbf{X}$ is not invertible, but this does *not* imply that no solution exists! In OLS, there always exists a solution, and when the Hessian is PD that solution is unique; when the Hessian is PSD, there are infinitely many solutions. (There always exists a solution to the expression $\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{y}$, because the range of $\mathbf{X}^\top\mathbf{X}$ and the range space of $\mathbf{X}^\top$ are equivalent; since $\mathbf{X}^\top\mathbf{y}$ lies in the range of $\mathbf{X}^\top$, it must equivalently lie in the range of $\mathbf{X}^\top\mathbf{X}$ and therefore there always exists a $\mathbf{w}$ that satisfies the equation $\mathbf{X}^\top\mathbf{X}\mathbf{w} = \mathbf{X}^\top\mathbf{y}$.)

The technique we just described is known as **ridge regression**. Note that now the expression $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ is invertible, regardless of rank of $\mathbf{X}$. Let's find $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$ through SVD:

$$\begin{aligned}
(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1} &= \left(\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_r & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{U}^\top\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_r & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^\top + \lambda\mathbf{I}\right)^{-1}\\
&= \left(\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_r^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^\top + \lambda\mathbf{I}\right)^{-1}\\
&= \left(\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_r^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^\top + \mathbf{V}(\lambda\mathbf{I})\mathbf{V}^\top\right)^{-1}\\
&= \left(\mathbf{V}\left(\begin{bmatrix}\mathbf{\Sigma}_r^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} + \lambda\mathbf{I}\right)\mathbf{V}^\top\right)^{-1}\\
&= \left(\mathbf{V}\begin{bmatrix}\mathbf{\Sigma}_r^2 + \lambda\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \lambda\mathbf{I}\end{bmatrix}\mathbf{V}^\top\right)^{-1}\\
&= (\mathbf{V}^\top)^{-1}\begin{bmatrix}\mathbf{\Sigma}_r^2 + \lambda\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \lambda\mathbf{I}\end{bmatrix}^{-1}\mathbf{V}^{-1}\\
&= \mathbf{V}\begin{bmatrix}(\mathbf{\Sigma}_r^2 + \lambda\mathbf{I})^{-1} & \mathbf{0}\\ \mathbf{0} & \frac{1}{\lambda}\mathbf{I}\end{bmatrix}\mathbf{V}^\top
\end{aligned}$$

Now with our slight tweak, the matrix $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ has become full rank and thus invertible. The singular values have become $\frac{1}{\sigma^2+\lambda}$ and $\frac{1}{\lambda}$, meaning that the singular values are guaranteed to be at most $\frac{1}{\lambda}$, solving our numerical instability issues. Furthermore, we have partially solved the overfitting issue. By penalizing the norm of $\mathbf{x}$, we encourage the weights corresponding to relevant features that capture the main structure of the true model, and penalize the weights corresponding to complex features that only serve to fine tune the model and fit noise in the data.