The Method of Least Squares

From Geometry to Regularization

Problem Setup

We are given a dataset consisting of:

- A data matrix $X \in \mathbb{R}^{n \times p}$, where n is the number of samples and p is the number of features.
- A vector of labels $y \in \mathbb{R}^n$.

The Goal

We want to find a weight vector $w \in \mathbb{R}^p$ such that the model's prediction, Xw, is the "best" approximation of the true labels y.

Assumptions

- The system is **overdetermined**: $n > p$. We have more equations (samples) than unknowns (features).
- The matrix X has **full column rank**: $\text{rank}(X) = p$. This means its columns are linearly independent.

The Residual Vector

In an overdetermined system, there is generally no exact solution for $Xw = y$. The vector y does not lie in the column space of X.
We define the **residual vector** r as the error of our approximation:

$$r = y - Xw$$

The Least Squares Formulation

The "best" approximation is the one that minimizes the magnitude of this residual vector. We choose to minimize its squared Euclidean norm (L_2 norm):

$$\min_{w \in \mathbb{R}^p} \|r\|^2 = \min_{w \in \mathbb{R}^p} \|y - Xw\|^2$$

The vector $\hat{w}$ that achieves this minimum is called the **least squares solution**.

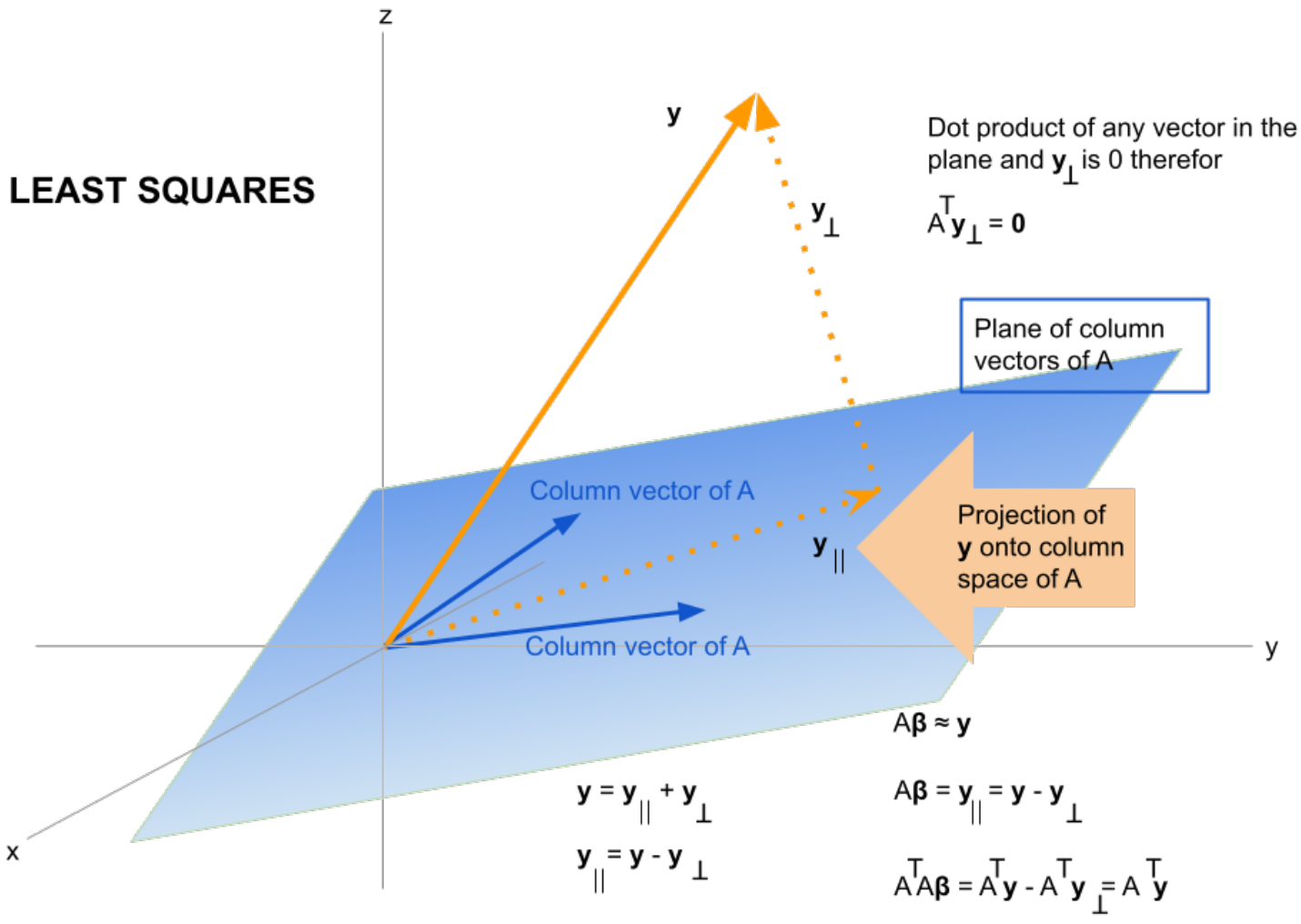
The Geometry of the Problem

The set of all possible predictions, $\{Xw \mid w \in \mathbb{R}^p\}$, forms a p-dimensional subspace in $\mathbb{R}^n$. This is the **column space** of X, denoted Col(X).
The least squares problem asks: What is the vector $\hat{p} = X\hat{w}$ in Col(X) that is closest to the vector y?

The Answer: Orthogonal Projection

The closest point in a subspace is the **orthogonal projection** of the vector onto that subspace.
This means the residual vector for the optimal solution, $r = y - \hat{p}$, must be **orthogonal** to the column space of X.

Geometric view



Deriving the Normal Equations Geometrically

The residual $r = y - X\hat{w}$ must be orthogonal to *every* vector in Col(X). This is equivalent to saying that r must be orthogonal to every column of X. Let the columns be $x_1, \ldots, x_p$.

$$x_i^T (y - X\hat{w}) = 0 \quad \text{for } i = 1, \ldots, p$$

We can write this compactly in matrix form:

$$\begin{pmatrix} — & x_1^T & — \\ & \vdots & \\ — & x_p^T & — \end{pmatrix} (y - X\hat{w}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

This is precisely:

The Normal Equations

$$X^T(y - X\hat{w}) = 0 \implies X^T X\hat{w} = X^T y$$

The Solution

The least squares solution $\hat{w}$ is found by solving the normal equations:

$$X^T X \hat{w} = X^T y$$

Existence and Uniqueness

If the matrix $X^T X$ is invertible, then a unique solution exists and is given by:

$$\hat{w} = (X^T X)^{-1} X^T y$$

When is $X^T X$ invertible?

- $X^T X$ is a square matrix of size $p \times p$.
- We need to show it has full rank, i.e., rank p.

Why is $X^T X$ Invertible? I

Theorem

If $X \in \mathbb{R}^{n \times p}$ has linearly independent columns (i.e., rank$(X) = p$), then $X^T X$ is invertible.

Why is X^TX Invertible? II

Proof.

To prove X^TX is invertible, we show that its null space contains only the zero vector. Let $v \in \mathbb{R}^p$ be a vector in the null space:

$$X^TXv = 0$$

Multiply by v^T from the left:

$$v^TX^TXv = v^T0 = 0$$

$$(Xv)^T(Xv) = 0$$

$$\|Xv\|^2 = 0$$

This implies $Xv = 0$. This means v is in the null space of X. But we assumed X has linearly independent columns, so its null space contains only the zero vector. Therefore, $v = 0$. Since the null space of X^TX is trivial, the matrix is invertible. □

Example: $n = 3, p = 2$ I

Let $\mathrm{X} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}$ and $\mathrm{y} = \begin{pmatrix} 1 \\ 3 \\ 8 \end{pmatrix}$. Rank is 2.

1. **Compute $\mathrm{X}^T\mathrm{X}$:**

$$\mathrm{X}^T\mathrm{X} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix}$$

2. **Compute $\mathrm{X}^T\mathrm{y}$:**

$$\mathrm{X}^T\mathrm{y} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 8 \end{pmatrix} = \begin{pmatrix} 12 \\ 31 \end{pmatrix}$$

3. **Solve the normal equations:** $\begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 12 \\ 31 \end{pmatrix}$

Example: $n = 3, p = 2$ II

The inverse of $\mathsf{X}^T\mathsf{X}$ is $\frac{1}{6}\begin{pmatrix} 14 & -6 \\ -6 & 3 \end{pmatrix}$.

$$\hat{\mathsf{w}} = \frac{1}{6}\begin{pmatrix} 14 & -6 \\ -6 & 3 \end{pmatrix}\begin{pmatrix} 12 \\ 31 \end{pmatrix} = \frac{1}{6}\begin{pmatrix} -18 \\ 21 \end{pmatrix} = \begin{pmatrix} -3 \\ 3.5 \end{pmatrix}$$

The Projection Matrix

The projection of y onto the column space of X is $\hat{p} = X\hat{w}$.
Substituting the solution for $\hat{w}$:

$$\hat{p} = X\left((X^T X)^{-1} X^T y\right)$$

We can group the terms that act on y:

$$\hat{p} = \left(X(X^T X)^{-1} X^T\right) y$$

Definition (Projection Matrix)

The matrix $P = X(X^T X)^{-1} X^T$ is the **projection matrix** that projects any vector in $\mathbb{R}^n$ onto the column space of X.

Properties

- P is symmetric ($P^T = P$).
- P is idempotent ($P^2 = P$). Projecting a second time doesn't change anything.

The Minimization Problem

The least squares solution is the vector $\hat{w}$ that minimizes the cost function $J(w)$:

$$\hat{w} = \arg\min_{w \in \mathbb{R}^p} J(w) = \arg\min_{w \in \mathbb{R}^p} \|y - Xw\|^2$$

Let's expand the squared norm:

$$J(w) = (y - Xw)^T (y - Xw)$$

$$= (y^T - w^T X^T)(y - Xw)$$

$$= y^T y - y^T Xw - w^T X^T y + w^T X^T Xw$$

Since $w^T X^T y$ is a scalar, it's equal to its transpose: $(y^T Xw)^T$.

$$J(w) = y^T y - 2y^T Xw + w^T X^T Xw$$

Finding the Minimum

To find the minimum of $J(\mathbf{w})$, we take its gradient with respect to $\mathbf{w}$ and set it to zero.

$$J(\mathbf{w}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

Using matrix calculus identities:

- $\nabla_{\mathbf{w}}(\mathbf{a}^T\mathbf{w}) = \mathbf{a}$
- $\nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{A}\mathbf{w}) = 2\mathbf{A}\mathbf{w}$ (for symmetric $\mathbf{A}$)

The gradient of our cost function is:

$$\begin{aligned}\nabla_{\mathbf{w}}J(\mathbf{w}) &= \nabla_{\mathbf{w}}(\mathbf{y}^T\mathbf{y}) - \nabla_{\mathbf{w}}(2\mathbf{y}^T\mathbf{X}\mathbf{w}) + \nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}) \\ &= 0 - 2(\mathbf{y}^T\mathbf{X})^T + 2(\mathbf{X}^T\mathbf{X})\mathbf{w} \\ &= -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w}\end{aligned}$$

Set the gradient to zero:

$$-2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = 0 \implies \mathbf{X}^T\mathbf{X}\hat{\mathbf{w}} = \mathbf{X}^T\mathbf{y}$$

This is exactly the same set of **normal equations** derived from the geometric approach.

Solving Least Squares with SVD

Let the reduced SVD of X be $X = U_r \Sigma_r V_r^T$.
Substitute this into the normal equations solution:

$$\hat{w} = (X^T X)^{-1} X^T y$$

Step 1: $X^T X$

$$X^T X = (U_r \Sigma_r V_r^T)^T (U_r \Sigma_r V_r^T) = V_r \Sigma_r^T U_r^T U_r \Sigma_r V_r^T$$

Since $U_r^T U_r = I$, this simplifies to $V_r \Sigma_r^2 V_r^T$.
Step 2: $(X^T X)^{-1}$

$$(V_r \Sigma_r^2 V_r^T)^{-1} = (V_r^T)^{-1} (\Sigma_r^2)^{-1} (V_r)^{-1} = V_r \Sigma_r^{-2} V_r^T$$

Step 3: Combine terms

$$\hat{w} = (V_r \Sigma_r^{-2} V_r^T)(V_r \Sigma_r U_r^T) y$$

Since $V_r^T V_r = I$, this simplifies:

$$\hat{w} = V_r \Sigma_r^{-1} U_r^T y$$

This term, $V_r \Sigma_r^{-1} U_r^T$, is the **pseudoinverse** of X, denoted $X^\dagger$.

The Problem of Small Singular Values

The solution is $\hat{w} = V_r \Sigma_r^{-1} U_r^T y$.
The matrix Σ_r^{-1} is a diagonal matrix with entries $1/\sigma_i$.

What if some σ_i are very small?

If a singular value σ_i is close to zero, its reciprocal $1/\sigma_i$ will be very large.

- This means that small changes or noise in the input data y can lead to huge changes in the solution vector $\hat{w}$.
- The problem is said to be **ill-conditioned**.
- The solution may have an extremely large norm, which often corresponds to overfitting.

Solving Ill-Conditioning: Regularization

To combat ill-conditioning and overfitting, we add a **penalty term** to the cost function. This penalizes large weight vectors.

$$J_{reg}(\mathrm{w}) = \|\mathrm{y} - \mathrm{X}\mathrm{w}\|^2 + \lambda\Omega(\mathrm{w})$$

- $\lambda \geq 0$ is the **regularization parameter**. It controls the strength of the penalty.
- $\Omega(\mathrm{w})$ is the penalty function, typically a norm of w.

Ridge Regression (L2 Regularization)

Ridge regression uses the squared L2 norm as the penalty: $\Omega(\mathbf{w}) = \|\mathbf{w}\|_2^2$. The cost function is:

$$J_{ridge}(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2$$

Taking the gradient and setting to zero yields the modified normal equations:

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

The Solution

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

Adding $\lambda\mathbf{I}$ to $\mathbf{X}^T\mathbf{X}$ adds λ to its eigenvalues, ensuring the matrix is invertible and well-conditioned, even if $\mathbf{X}^T\mathbf{X}$ had very small eigenvalues.

Proof of the Ridge Regression Solution I

The cost function is $J(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2$.

1. Expand the cost function:

$$\begin{aligned} J(\mathbf{w}) &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^T\mathbf{w} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} + \lambda\mathbf{w}^T\mathbf{I}\mathbf{w} \\ &= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{w}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} \end{aligned}$$

2. Take the gradient with respect to $\mathbf{w}$:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \nabla_{\mathbf{w}}(\mathbf{y}^T\mathbf{y}) - \nabla_{\mathbf{w}}(2\mathbf{y}^T\mathbf{X}\mathbf{w}) + \nabla_{\mathbf{w}}(\mathbf{w}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w})$$

Using the identities $\nabla_{\mathbf{w}}(\mathbf{a}^T\mathbf{w}) = \mathbf{a}$ and $\nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{A}\mathbf{w}) = 2\mathbf{A}\mathbf{w}$:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = -2\mathbf{X}^T\mathbf{y} + 2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w}$$

Proof of the Ridge Regression Solution II

3. Set the gradient to zero to find the minimum:

$$-2X^T y + 2(X^T X + \lambda I)\hat{w} = 0$$

$$(X^T X + \lambda I)\hat{w} = X^T y$$

The Solution

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

Step 1: Simplify the Inverse I

Now, we simplify the core inverse term from the ridge solution: $(X^T X + \lambda I)^{-1}$.

$$
\begin{aligned}
(X^T X + \lambda I)^{-1} &= (V\Sigma^2 V^T + \lambda I)^{-1} \\
\text{Since } I = VV^T \quad &= (V\Sigma^2 V^T + \lambda VV^T)^{-1} \\
\text{Factor out V and } V^T \quad &= (V(\Sigma^2 + \lambda I)V^T)^{-1} \\
\text{Using } (ABC)^{-1} = C^{-1}B^{-1}A^{-1} \quad &= (V^T)^{-1}(\Sigma^2 + \lambda I)^{-1}V^{-1} \\
\text{Since V is orthogonal} \quad &= V(\Sigma^2 + \lambda I)^{-1}V^T
\end{aligned}
$$

The inverse of the diagonal matrix $(\Sigma^2 + \lambda I)$ is simple:

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

Step 2: Finding the Prediction $\hat{y}$ I

The predicted values $\hat{y}$ are obtained by $\hat{y} = X\hat{w}_{ridge}$. Let's find the operator that maps y to $\hat{y}$.

$$\begin{aligned}
\hat{y} &= X\hat{w}_{ridge} \\
&= X(X^TX + \lambda I)^{-1}X^Ty \\
&= (U\Sigma V^T)\left[V(\Sigma^2 + \lambda I)^{-1}V^T\right](U\Sigma V^T)^Ty \\
&= (U\Sigma V^T)\left[V(\Sigma^2 + \lambda I)^{-1}V^T\right](V\Sigma^T U^T)y \\
\text{Since } V^TV = I \qquad &= U\Sigma(\Sigma^2 + \lambda I)^{-1}\Sigma U^Ty
\end{aligned}$$

This gives us the mapping: $\hat{y} = (U\Sigma_{ridge}U^T)y$, where $\Sigma_{ridge} = \Sigma(\Sigma^2 + \lambda I)^{-1}\Sigma$.

Step 3: The Matrix Σ_{ridge} and its Singular Values I

Deriving Σ_{ridge}

Let's find the explicit form of $\Sigma_{ridge} = \Sigma(\Sigma^2 + \lambda I)^{-1}\Sigma$.
Since all matrices involved are diagonal, we can multiply their diagonal elements:

$$\begin{aligned}\Sigma_{ridge} &= \text{diag}(\sigma_i) \cdot \text{diag}\left(\frac{1}{\sigma_i^2 + \lambda}\right) \cdot \text{diag}(\sigma_i) \\ &= \text{diag}\left(\sigma_i \cdot \frac{1}{\sigma_i^2 + \lambda} \cdot \sigma_i\right) \\ &= \text{diag}\left(\frac{\sigma_i^2}{\sigma_i^2 + \lambda}\right)\end{aligned}$$

Step 3: The Matrix Σ_{ridge} and its Singular Values II

The Singular Values of the Ridge Operator

The matrix for the reduced SVD of the ridge regression operator is:

$$\Sigma_{ridge} = \begin{pmatrix} \frac{\sigma_1^2}{\sigma_1^2+\lambda} & 0 & \dots & 0 \\ 0 & \frac{\sigma_2^2}{\sigma_2^2+\lambda} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\sigma_d^2}{\sigma_d^2+\lambda} \end{pmatrix}$$

The singular values are $\sigma_i' = \frac{\sigma_i^2}{\sigma_i^2+\lambda}$.

Interpretation: Shrinkage I

The expression $\sigma_i' = \frac{\sigma_i^2}{\sigma_i^2+\lambda}$ reveals that ridge regression acts as a **shrinkage operator**.

- It takes the singular values σ_i of the original data matrix X and maps them to new, smaller values σ_i'.
- **Case 1**: No Regularization ($\lambda \to 0$)

$$\lim_{\lambda \to 0} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} = \frac{\sigma_i^2}{\sigma_i^2} = 1$$

  This recovers the Ordinary Least Squares solution (for the components of y in the column space of X).
- **Case 2**: Strong Regularization ($\lambda \to \infty$)

$$\lim_{\lambda \to \infty} \frac{\sigma_i^2}{\sigma_i^2 + \lambda} = 0$$

  This shrinks all singular values towards zero, resulting in $\hat{y} \to 0$.

Interpretation: Shrinkage II

Ridge regression effectively filters the data by reducing the influence of directions associated with smaller singular values, which helps to control variance and prevent overfitting.

Ridge regression: conclusions

- We started with the closed-form solution for ridge regression:

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

- By substituting the SVD of the data matrix, $X = U\Sigma V^T$, we analyzed the prediction operator that maps y to $\hat{y}$.
- We found this operator has the form $M_{ridge} = U\Sigma_{ridge} U^T$, which is an SVD-like decomposition.
- The diagonal matrix of new singular values, Σ_{ridge}, is given by:

$$\Sigma_{ridge} = \text{diag}\left(\frac{\sigma_1^2}{\sigma_1^2 + \lambda}, \ldots, \frac{\sigma_d^2}{\sigma_d^2 + \lambda}\right)$$

- This shows precisely how the regularization parameter λ shrinks the singular values of the original problem to produce a more stable solution.

Lasso Regression (L1 Regularization)

Lasso uses the L1 norm as the penalty: $\Omega(\mathrm{w}) = \|\mathrm{w}\|_1 = \sum_{i=1}^{p} |w_i|$.
The cost function is:

$$J_{lasso}(\mathrm{w}) = \|\mathrm{y} - \mathrm{X}\mathrm{w}\|^2 + \lambda\|\mathrm{w}\|_1$$

The Effect

The L1 norm is not differentiable at zero, so there is no simple closed-form solution like in Ridge. The solution must be found with iterative algorithms. A key property of the L1 penalty is that it promotes **sparsity**: it tends to drive many of the components of the weight vector $\hat{\mathrm{w}}$ to be exactly zero. This is useful for feature selection.

L2 vs. L1 Regularization: A Geometric View

Minimizing the regularized cost is equivalent to minimizing the original cost subject to a constraint on the penalty norm:

Ridge (L2)

$$\min \|y - Xw\|^2 \quad \text{s.t.} \quad \|w\|_2^2 \leq t$$

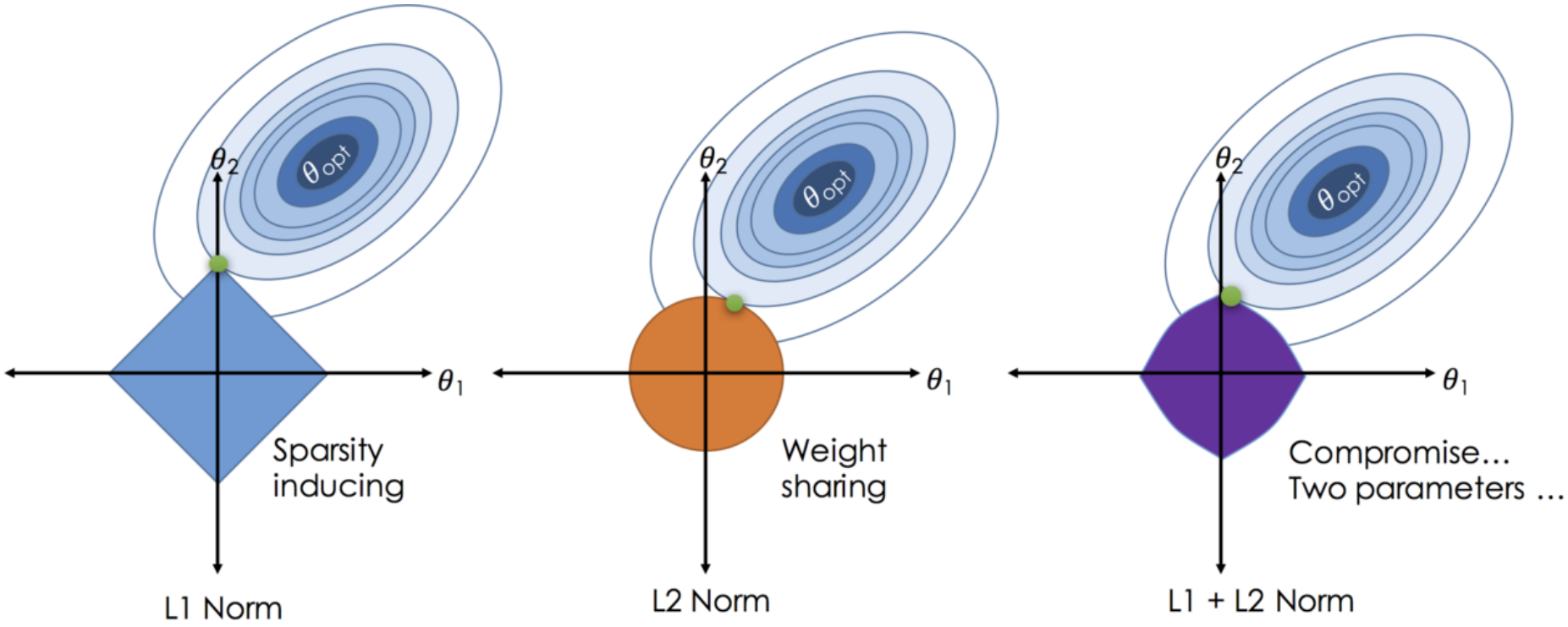
The constraint region is a circle/sphere. The solution is found where the elliptical contours of the LS error first touch the circle. This rarely happens on an axis.

Lasso (L1)

$$\min \|y - Xw\|^2 \quad \text{s.t.} \quad \|w\|_1 \leq t$$

The constraint region is a diamond/rhombus. The elliptical contours are much more likely to touch the diamond at one of its sharp corners, which lie on the axes. A corner-touch means one component of w is zero.

L2 vs. L1 Regularization: A Geometric View



Elastic Net: The Best of Both Worlds

Elastic Net regularization combines the L1 and L2 penalties.

Cost Function

The cost function is a linear combination of the Ridge and Lasso penalties:

$$J_{elastic}(\mathrm{w}) = \|\mathrm{y} - \mathrm{X}\mathrm{w}\|^2 + \lambda_1 \|\mathrm{w}\|_1 + \lambda_2 \|\mathrm{w}\|_2^2$$

This is often re-parameterized using a mixing ratio $\alpha \in [0, 1]$:

$$J(\mathrm{w}) = \|\mathrm{y} - \mathrm{X}\mathrm{w}\|^2 + \lambda \left(\alpha \|\mathrm{w}\|_1 + \frac{1-\alpha}{2} \|\mathrm{w}\|_2^2 \right)$$

- If $\alpha = 1$, it becomes Lasso.
- If $\alpha = 0$, it becomes Ridge.

L1 vs L2 vs EN

Conclusion

- **L2** shrinks all coefficients, preferring solutions with small weights (**minimum length**).
- **L1** performs feature selection, preferring solutions with many zero weights (**sparsity**).
- **EN** Performs feature selection like Lasso. Handles correlated features better than Lasso (which tends to pick one and discard others). The L2 term encourages grouped selection. Inherits the stability of Ridge regression.