

Machine Learning

Textbook: Machine Learning Refined, by Watt, Borhani, and Katsaggelos, 2nd edition:
MLRefined.com

Previous Lecture

- ◆ Linear regression predicts a numerical response using a linear model such as $w_0 + w_1x$.
- ◆ A residual is predicted value minus observed value.
- ◆ Least squares chooses parameters that minimize the mean squared residual.
- ◆ Setting the gradient of the least-squares cost equal to zero gives equations for the optimal parameters.
- ◆ For our data, the learned line is $\hat{y} = 43.7 + 8.3x$ with MSE 0.86.
- ◆ Linear-regression MSE is quadratic in the parameters, so its unique minimum can be found by gradient methods, Newton's method, or direct linear algebra.



§5.2 - Least Squares in Several Variables

Last time:

one input x ,
one output y ,
model $\hat{y} = w_0 + w_1x$.

Today:

many inputs at once.

Linear algebra will organize the entire regression problem.

Several Features

Suppose each observation has N features:

$$\mathbf{x}_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ \vdots \\ x_{pN} \end{bmatrix}.$$

Example:

predict exam score using
hours studied,
prior-course score,
attendance,
etc.

One observation:

one feature vector $\mathbf{x}_p$.

The Multivariable Linear Model

Use

$$f(\mathbf{x}; \mathbf{w}) = w_0 + w_1x_1 + \cdots + w_Nx_N.$$

Parameters:

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_N \end{bmatrix}.$$

There are

$$N + 1$$

parameters including the intercept.

The model is linear in the **parameters**.

Absorb the Intercept

Add a constant feature:

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ x_{p1} \\ \vdots \\ x_{pN} \end{bmatrix}.$$

Then

$$f(\mathbf{x}_p; \mathbf{w}) = \tilde{\mathbf{x}}_p^T \mathbf{w}.$$

The leading 1 multiplies w_0 .

Now intercept and slopes are handled uniformly.

Stack All Observations

Suppose there are P observations.

Place one augmented feature vector in each row:

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1N} \\ 1 & x_{21} & \cdots & x_{2N} \\ \vdots & \vdots & & \vdots \\ 1 & x_{P1} & \cdots & x_{PN} \end{bmatrix}.$$

X is the **design matrix**.

Dimensions:

$$X : P \times (N + 1).$$

Stack the Outputs

Observed responses:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{bmatrix}.$$

Predictions for every observation at once:

$$\hat{\mathbf{y}} = X\mathbf{w}.$$

Dimensions:

$$\begin{aligned} & (P \times (N + 1))((N + 1) \times 1) \\ & = P \times 1. \end{aligned}$$

One matrix multiplication produces all predictions.

The Residual Vector

Individual residual:

$$r_p = \hat{y}_p - y_p.$$

Stack them:

$$\mathbf{r} = X\mathbf{w} - \mathbf{y}.$$

Then

$$\|\mathbf{r}\|^2 = r_1^2 + \dots + r_P^2.$$

So mean squared error becomes

$$g(\mathbf{w}) = \frac{1}{P} \|X\mathbf{w} - \mathbf{y}\|^2.$$

Same least-squares problem.

Now in vector form.

Expand the Cost

Because

$$\|X\mathbf{w} - \mathbf{y}\|^2 = (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}),$$

we have

$$g(\mathbf{w}) = \frac{1}{P} [\mathbf{w}^T X^T X \mathbf{w} - 2\mathbf{y}^T X \mathbf{w} + \mathbf{y}^T \mathbf{y}].$$

Thus:

least-squares cost is quadratic in $\mathbf{w}$.

That fact drove the previous lesson.

Differentiate the Matrix Form

Gradient:

$$\nabla g(\mathbf{w}) = \frac{2}{P} X^T (X\mathbf{w} - \mathbf{y}).$$

Set equal to zero:
 $X^T(X\mathbf{w} - \mathbf{y}) = \mathbf{0}.$

Therefore
 $X^TX\mathbf{w} = X^T\mathbf{y}.$

These are the **normal equations**.

One matrix equation replaces all parameter-by-parameter equations.

Recover Last Lesson

For the study-hours data:
 $x = 1, 2, 3, 4, 5.$

The design matrix is

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{bmatrix},$$

and

$$\mathbf{y} = \begin{bmatrix} 52 \\ 61 \\ 67 \\ 78 \\ 85 \end{bmatrix}.$$

Then

$$X^TX = \begin{bmatrix} 5 & 15 \\ 15 & 55 \end{bmatrix},$$

and

$$X^T\mathbf{y} = \begin{bmatrix} 343 \\ 1112 \end{bmatrix}.$$

The Normal Equations Are the Same Equations

Solve

$$\begin{bmatrix} 5 & 15 \\ 15 & 55 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 343 \\ 1112 \end{bmatrix}.$$

This is exactly

$$5w_0 + 15w_1 = 343,$$

$15w_0 + 55w_1 = 1112.$

Therefore

$$\mathbf{w}^* = \begin{bmatrix} 43.7 \\ 8.3 \end{bmatrix}.$$

Nothing changed mathematically.

The notation scaled.

When Is the Solution Unique?

If the columns of X are linearly independent,

then

$$X^T X$$

is positive definite.

Therefore:

$$X^T X$$

is invertible,

and the least-squares minimizer is unique.

Formally,

$$\mathbf{w}^* = (X^T X)^{-1} X^T \mathbf{y}.$$

But computationally:

do not form the inverse unless there is a specific reason.

Solve a linear system instead.

Now Two Predictors

Consider four observations:

Observation	1	2	3	4
x_1	0	1	0	1
x_2	0	0	1	1
y	1	3	4	7

Model:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2.$$

Design matrix:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Response:

$$\mathbf{y} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 7 \end{bmatrix}.$$

Build the Normal Equations

Compute

$$X^T X = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}.$$

Also

$$X^T \mathbf{y} = \begin{bmatrix} 15 \\ 10 \\ 11 \end{bmatrix}.$$

Therefore

$$\begin{bmatrix} 4 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 10 \\ 11 \end{bmatrix}.$$

Solution:

$$\mathbf{w}^* = \begin{bmatrix} 3/4 \\ 5/2 \\ 7/2 \end{bmatrix}.$$

So

$$\hat{y} = 0.75 + 2.5x_1 + 3.5x_2.$$

Predictions and Residuals

Predictions:

$$0.75, 3.25, 4.25, 6.75.$$

Residual vector:

$$\mathbf{r} = X\mathbf{w}^* - \mathbf{y}$$

$$= \begin{bmatrix} -0.25 \\ 0.25 \\ 0.25 \\ -0.25 \end{bmatrix}.$$

MSE:

$$\begin{aligned} & \frac{1}{4} [4(0.25)^2] \\ &= 0.0625. \end{aligned}$$

The line from one predictor has become a plane in feature space.

But the algebra is unchanged.

The Geometry of Least Squares

The prediction vector

$$X\mathbf{w}$$

is a linear combination of the columns of X .

Therefore

$$X\mathbf{w} \in \text{Col}(X).$$

But the observed vector

$$\mathbf{y}$$

may not lie in $\text{Col}(X)$.

Least squares chooses the vector in

$$\text{Col}(X)$$

closest to $\mathbf{y}$.

So

$$X\mathbf{w}^* = \text{proj}_{\text{Col}(X)} \mathbf{y}.$$

Least squares is an orthogonal projection problem.

Why the Residual Is Orthogonal

At the optimum:

$$X^T(X\mathbf{w}^* - \mathbf{y}) = \mathbf{0}.$$

But

$$\mathbf{r} = X\mathbf{w}^* - \mathbf{y}.$$

Therefore

$$X^T \mathbf{r} = \mathbf{0}.$$

That means:

the residual is orthogonal to every column of X .

Therefore:

residual $\perp \text{Col}(X)$.

This is exactly the projection condition.

Check the Orthogonality

For our two-predictor example:

$$\mathbf{r} = \begin{bmatrix} -1/4 \\ 1/4 \\ 1/4 \\ -1/4 \end{bmatrix}.$$

First column of X :

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Dot product:

$$-\frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{4} = 0.$$

Second column:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Dot product:

$$\frac{1}{4} - \frac{1}{4} = 0.$$

Third column:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Dot product:

$$\frac{1}{4} - \frac{1}{4} = 0.$$

Normal equations

=

residual orthogonality.

What if the Columns Are Dependent?

Suppose one feature is exactly twice another:

$$x_2 = 2x_1.$$

Then one column of X is a multiple of another.

The columns are linearly dependent.

Therefore

$$X^T X$$

is singular.

The data cannot distinguish uniquely between the coefficients of x_1 and x_2 .

Example:

$$w_1 x_1 + w_2 x_2 = (w_1 + 2w_2) x_1.$$

Many pairs (w_1, w_2) give the same predictions.

This is an extreme form of **multicollinearity**.

Rank Matters

For a unique least-squares parameter vector:

we want

$$\text{rank}(X) = N + 1.$$

Full column rank:

unique coefficients.

Rank deficient:

coefficients may not be unique.

But:

a best prediction vector can still exist.

Later we will see methods that help when predictors are redundant or nearly redundant.

Matrix Form in Python

For the study-hours data:

```
import numpy as np

x = np.array([1., 2., 3., 4., 5.])
y = np.array([52., 61., 67., 78., 85.])

X = np.column_stack((np.ones(len(x)), x))
```

New Python ideas:

`np` creates a column of 1s;

`np` combines columns into a matrix.

Now

X

produces the 5×2 design matrix.

Solve the Normal Equations in Python

Compute

```
A = X.T @ X
b = X.T @ y
```

Then solve:

```
w = np.linalg.solve(A, b)
w
```

Output:

```
[43.7  8.3]
```

New Python idea:

```
X.T
```

is the transpose of X .

Again:

```
solve the system;
do not explicitly compute  $(X^T X)^{-1}$ .
```

A Better Numerical Command

NumPy can solve the least-squares problem directly:

```
w, residuals, rank, singular_values = \
    np.linalg.lstsq(X, y, rcond=None)
```

Then

```
w
```

returns

```
[43.7  8.3]
```

Why use this?

It avoids explicitly forming $X^T X$.

It also behaves better when X is nearly rank deficient.

For numerical work:

```
np is usually preferable to forming the normal equations by hand.
```

Two Predictors in Python

Enter the design matrix directly:

```
X = np.array([[1., 0., 0.],
              [1., 1., 0.],
              [1., 0., 1.]])
```

```
[1., 1., 1.]])
```

```
y = np.array([1., 3., 4., 7.])
```

Solve:

```
w, residuals, rank, singular_values = \
    np.linalg.lstsq(X, y, rcond=None)
```

```
w
```

Output:

```
[0.75 2.50 3.50]
```

Check residuals:

```
r = X @ w - y
X.T @ r
```

Output:

approximately 0.

The computer confirms the projection geometry.

From One Predictor to Many

One predictor:

$$\hat{y} = w_0 + w_1 x.$$

Many predictors:

$$\hat{y} = w_0 + w_1 x_1 + \cdots + w_N x_N.$$

But in matrix form:

both are simply

$$\hat{\mathbf{y}} = X\mathbf{w}.$$

And learning is

$$\underset{\mathbf{w}}{\text{minimize}} \quad \|X\mathbf{w} - \mathbf{y}\|^2.$$

The dimension changes.

The structure does not.

What did we learn?

- ◆ The design matrix X stores all observations and all linear features.
 - ◆ All predictions are produced at once by $\hat{\mathbf{y}} = X\mathbf{w}$.
 - ◆ Least squares minimizes $\|X\mathbf{w} - \mathbf{y}\|^2$.
 - ◆ Setting the gradient to zero gives the normal equations $X^T X \mathbf{w} = X^T \mathbf{y}$.
 - ◆ Geometrically, $X\mathbf{w}^*$ is the orthogonal projection of $\mathbf{y}$ onto the column space of X .
 - ◆ Full column rank gives a unique least-squares parameter vector; dependent columns create nonuniqueness.
- 

Next

- Least squares uses squared residuals.
- Squaring is mathematically convenient, but it is not the only way to measure error.
- Different loss functions respond differently to large residuals and outliers.
- **1.11 - Alternatives to Least Squares**