Scientific Computing in Python Lecture 5

Least squares, principal components, and clustering

Zan Ahmad

Winter 2026

Linear least squares

$$J(\beta) = \|A\beta - y\|_2^2.$$

Linear least squares

$$J(\beta) = \|A\beta - y\|_2^2.$$

$$\nabla J = 2A^\mathsf{T}(A\beta - y) = 0 \quad \Longrightarrow \quad A^\mathsf{T}A\beta = A^\mathsf{T}y.$$

Residual orthogonality

At a least-squares minimizer,

$$A^{\mathsf{T}} r = 0, \qquad r = A\widehat{\beta} - y.$$

Residual orthogonality

At a least-squares minimizer,

$$A^{\mathsf{T}} r = 0, \qquad r = A\widehat{\beta} - y.$$

With an intercept column, $\sum_i r_i = 0$ on the training set. There is no corresponding identity for held-out residuals.

Units of a regression coefficient

$$\widehat{y} = \beta_0 + \beta_1 x.$$

If $x' = ax$ with $a \neq 0$, then

Units of a regression coefficient

$$\widehat{y} = \beta_0 + \beta_1 x.$$

If $x' = ax$ with $a \neq 0$, then

$$\widehat{y} = \beta_0 + \frac{\beta_1}{a} x'.$$

Training and test error

$$\mathrm{RMSE}_{\mathrm{test}} = \sqrt{\frac{1}{n_{\mathrm{test}}} \sum_{i \in \mathrm{test}} (y_i - \widehat{y}_i)^2}.$$

Training and test error

$$\mathrm{RMSE}_{\mathrm{test}} = \sqrt{\frac{1}{n_{\mathrm{test}}} \sum_{i \in \mathrm{test}} (y_i - \widehat{y}_i)^2}.$$

Fit the coefficients using training rows only. Test RMSE has the same physical unit as y.

Principal component directions

$$\max_{v^\top v=1} \|X_c v\|_2^2.$$

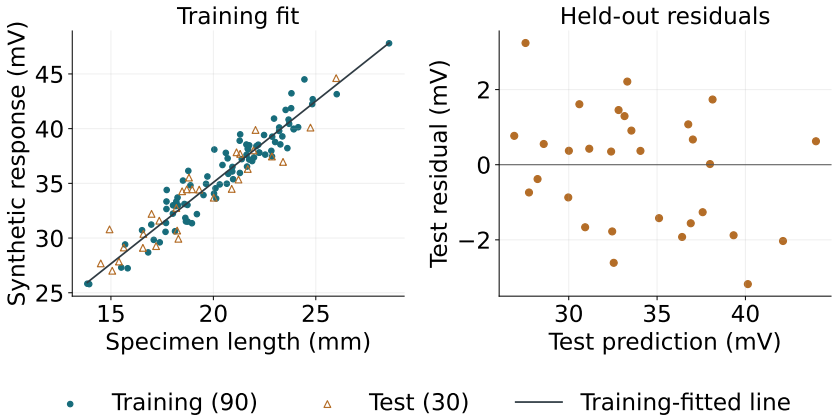
Principal component directions

$$\max_{v^\mathsf{T} v = 1} \|X_c v\|_2^2.$$

$$X_c^\mathsf{T} X_c v = \lambda v.$$

The leading eigenvector supplies the first principal direction.

Fitted response and held-out residuals



Projection and reconstruction

$$z = (x - \mu)v_1, \qquad \widehat{x} = \mu + zv_1^{\mathsf{T}}.$$

Projection and reconstruction

$$z = (x - \mu)v_1, \qquad \widehat{x} = \mu + zv_1^{\mathsf{T}}.$$

The reconstruction error is orthogonal to v_1. A large retained variance need not preserve information useful for predicting a separate response.

Scaling and distance

$$d^2(x, x') = \sum_j (x_j - x'_j)^2.$$

Scaling and distance

$$d^2(x, x') = \sum_j (x_j - x'_j)^2.$$

$$z_j = \frac{x_j - \mu_{j,\text{train}}}{s_{j,\text{train}}}, \qquad s_{j,\text{train}} > 0.$$

A zero-variance coordinate

If $s_{j,\text{train}} = 0$, division is undefined.

A zero-variance coordinate

If $s_{j,\mathrm{train}} = 0$, division is undefined. A fixed rule must remove or separately handle that coordinate. Test data must not determine the training scale.

The k-means objective

$$J(c, \mu) = \sum_i \|x_i - \mu_{c_i}\|_2^2.$$

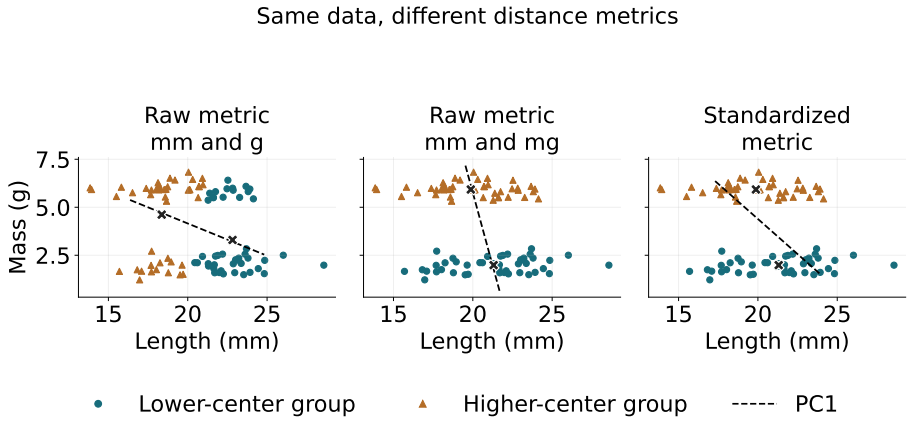
The k-means objective

$$J(c, \mu) = \sum_i \|x_i - \mu_{c_i}\|_2^2.$$

$$c_i = \arg\min_k \|x_i - \mu_k\|_2^2, \qquad \mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i.$$

The update requires $|C_k| > 0$; empty clusters need a separate rule.

Effect of feature units



Group labels

Permuting the center labels leaves the objective unchanged.

Group labels

Permuting the center labels leaves the objective unchanged. Group 0 and group 1 have no intrinsic biological meaning. Interpretation requires information beyond the distance calculation.

Fitted transformations and evaluation

$$\text{training data} \longmapsto (\mu, s, v_1, \widehat{\beta}, \widehat{\mu}_k).$$

Fitted transformations and evaluation

$$\text{training data} \longmapsto (\mu, s, v_1, \widehat{\beta}, \widehat{\mu}_k).$$

The fitted quantities are then held fixed when transforming and evaluating test observations.

A numerical implementation

```
beta = np.linalg.lstsq(A_train, y_train, rcond=None)[0]
y_pred = A_test @ beta
rmse = np.sqrt(np.mean((y_test-y_pred)**2))
```

No test response enters the coefficient solve.