Linear Regression II: Generalization and Statistical Inference

Methodology Recap: Fitting Regression with 1, 2, ≥3 Points in Higher Dimensions

1. Detailed Conceptual Overview

This is a continuation of what we saw in **Linear Regression I** — but now extended into **higher-dimensional spaces**.

The methodology for regression depends on the **number of data points relative to the number of features (dimensions)**.

a) Case 1: 1 Point

- With only one data point, infinitely many regression lines/hyperplanes can fit.
- The system is **underdetermined** — no unique solution.
- Example: In 2D, one point doesn't define a line.

b) Case 2: 2 Points

- Two points in 2D define a **unique straight line**.
- In higher dimensions, two points define a line (but not a plane/hyperplane).
- Still underdetermined if we want to fit a full regression with multiple features.

c) Case 3: ≥3 Points

- When we have at least as many points as required, we can fit regression meaningfully.

- But typically, with ≥3 points, **no perfect fit exists** (points don't all fall on one line/plane).
- Regression solves this by finding the **best-fit hyperplane** that minimizes error.

---

d) Higher Dimensions (Multivariate Regression)

- Inputs: $x \in \mathbb{R}^d$.
- Outputs: $y \in \mathbb{R}$.
- Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d$$

- For n data points and dd features:
  - If n < d → underdetermined (many possible solutions).
  - If n = d → exact solution may exist (if data is well-conditioned).
  - If n > d → regression finds a **best-fit solution** using least squares.

---

e) General Rule of Thumb:

- You need **more data points than features** for stable regression.
- Otherwise, the solution may be non-unique or unstable (high variance in coefficients).

---

2. Key Takeaways

- **1 point:** No unique regression (underdetermined).
- **2 points (2D):** Unique line, but not generalizable.
- **≥3 points:** Typically, no perfect fit → regression finds best-fit hyperplane.
- In higher dimensions:
  - If n < d: underdetermined (infinite solutions).
  - If n = d: unique solution possible.
  - If n > d: least squares finds the best-fit.

- **Golden rule:** More data points than features are needed for meaningful regression.

Generalization in Machine Learning (In-sample vs Out-of-sample Error, Higher Dimensions)

1. Detailed Conceptual Overview

The ultimate goal of regression (and ML in general) is not just to fit the training data, but to **generalize** — i.e., perform well on unseen data.

This section extends the concept of **generalization** we saw earlier, but now with emphasis on **higher-dimensional settings**.

---

a) In-sample Error (Training Error):

- Error computed on the **training data**.
- Regression aims to minimize this via least squares.
- Typically **low** because the model has already seen this data.
- Formula:

$$E_{in} = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} (y_i - h(x_i))^2$$

---

b) Out-of-sample Error (Test/Generalization Error):

- Error on **new, unseen data** from the same distribution.
- True measure of model quality.
- Formula (expected error):

$$E_{out} = \mathbb{E}_{(x,y)\sim P} \left[(y - h(x))^2 \right]$$

---

c) Generalization Challenge in Higher Dimensions:

- As the **number of features (d)** increases, the model becomes more flexible.
- This often reduces in-sample error (model fits training data better).
- But risk of **overfitting** increases → out-of-sample error may rise.

---

d) Geometric View:

- In low dimensions, we project the target vector yy onto a line/plane.
- In high dimensions, we project onto a subspace spanned by many features.
- If the subspace is too large relative to data size, projection becomes unstable (sensitive to noise).

---

e) Balancing Generalization:

- Need to avoid both **underfitting** (high bias) and **overfitting** (high variance).
- More features ≠ always better.
- Rule: We need **sufficiently more data points than features** for the regression to generalize well.

---

2. Key Takeaways

- **In-sample error:** Training error, usually low.
- **Out-of-sample error:** Test error, true measure of generalization.
- In higher dimensions:
  - More features reduce training error but risk overfitting.
  - Out-of-sample error increases if the model memorizes the training data.
- **Projection perspective:** Regression projects data onto subspaces; too many features make the projection unstable.
- **Goal:** Balance bias (underfitting) and variance (overfitting) for good generalization.

Using Error Metrics to Generalize (Matrix Solution, Noise in Data, Probability Bounds)

1. Detailed Conceptual Overview

So far, we've described the **in-sample vs out-of-sample error** conceptually.

This topic digs deeper into **how regression error behaves mathematically**, especially when noise and higher dimensions come into play.

---

a) Matrix Solution for Regression Error

- Recall regression solution:
  $\hat{\theta} = (X^T X)^{-1} X^T y$
- Prediction:
  $\hat{y} = X\hat{\theta}$
- Residual error vector:
  $e = y - \hat{y}$
- In least squares, this error is **orthogonal** to the column space of X.

---

b) Noise in Data

Assume data comes from:

$$y = X\theta^* + \epsilon$$

where θ^* is the true parameter vector and ϵ is random noise.

---

1. y (Target Vector):

- The actual outputs/labels we observe.
- Example: House prices, student scores, etc.
- Size: n×1 (where n = number of data points).

---

2. X (Design Matrix):

- Contains all input features.
- Each row = one data point, each column = one feature (plus a column of 1's for intercept).
- Size: n $\times$ d (n = data points, d = features).

Example with 2 features:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{bmatrix}$$

---

3. θ^* (True Coefficients):

- The real, underlying weights that nature uses to generate y.
- Regression tries to **estimate these** using data.
- Size: d×1.

Example:

- If the true model is $y = 2x_1 + 3x_2 + 5$, then

  $\theta^* = [5, 2, 3]^T$.

---

4. ϵ (Noise):

- Random error term captures things not explained by the model.
- Represents measurement error, randomness, or hidden factors.
- Typically assumed:

$$\epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

  → Mean = 0, variance = σ^2, independent across data points.

Example:

- If house price is affected by "luck" (e.g., buyer really likes the house), this randomness goes into ϵ.

---

5. $\hat{\theta}$ (Estimated Coefficients):

- What regression actually computes from data.
- Formula:
  $\hat{\theta} = (X^T X)^{-1} X^T y$

---

6. Error in Coefficients ($\hat{\theta} - \theta^*$):

- The difference between the estimated weights and the true underlying weights.
- Can be written as:
  $\hat{\theta} - \theta^* = (X^T X)^{-1} X^T \epsilon$

Meaning:

- The **noise in data (ϵ) directly affects coefficient accuracy**.
- If features in X are highly correlated, this error gets amplified (since $X^T X$ becomes unstable).

✅ So in summary:

- y: observed outputs
- X: features
- θ^*: true model weights
- ϵ: noise/randomness
- $\hat{\theta}$: estimated weights from regression

Noise → propagates into $\hat{\theta}$ → affects generalization ability.

---

c) Probability Bounds on Error

- With random noise ϵ\epsilon, regression error can be expressed probabilistically.

- Example (variance of coefficients):

  $\mathrm{Var}(\hat{\theta}) = \sigma^2 (X^T X)^{-1}$
- Implications:
  - More data points (nn) → smaller variance → better generalization.
  - Highly correlated features → larger variance → poor generalization.

---

d) Error Convergence

We want to understand: as we collect **more data**, how do regression errors (in-sample and out-of-sample) behave?

1. Model Setup

True model:

$y = X\theta^* + \epsilon$

where:

- X = feature matrix (n×d)
- θ^* = true underlying coefficients
- $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ = random noise

Regression estimate:

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

Prediction:

$\hat{y} = X\hat{\theta}$

---

2. Expected Error

We're interested in **mean squared error (MSE):**

$E\left[(y - \hat{y})^2\right]$

Break it down into:

- **Bias** (systematic error from wrong model assumptions)
- **Variance** (error due to randomness in training data/noise).

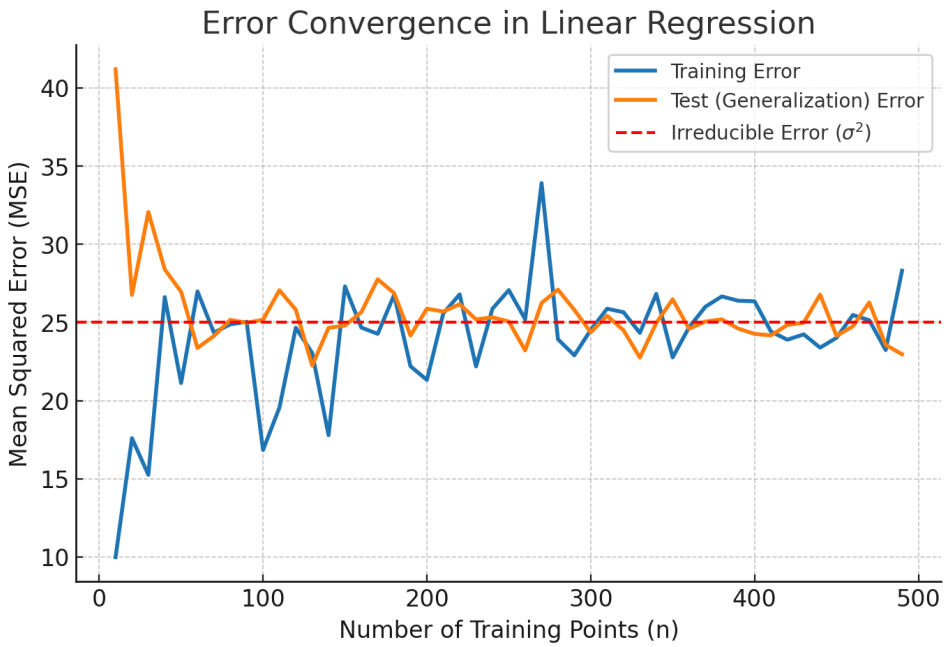
In linear regression with the correct model, **bias = 0**, so error comes mainly from **variance due to noise**.

---

3. Error Convergence Intuition

- With **few data points (small n)**:
  - The estimated coefficients $\hat{\theta}$ fluctuate a lot due to noise.
  - High **variance** in predictions → generalization error is large.
- With **many data points (large n)**:
  - The term $(X^T X)^{-1}$ becomes more stable.
  - The effect of random noise averages out.
  - Coefficient estimates $\hat{\theta}$ converge to the true θ^*.
  - Out-of-sample error approaches the irreducible noise variance σ^2.

  Here's the **graphical example of Error Convergence**



  - **Training Error (blue):** starts low (since the model can "memorize" small datasets), then gradually stabilizes.

- **Test Error (orange):** initially high with few points (unstable coefficients), but decreases as nnn grows.
- **Irreducible Error (red dashed line):** the floor at σ2\sigma^2σ2 — even with infinite data, test error can't go below this, because noise is unavoidable.

This illustrates that **as the sample size increases, out-of-sample error converges to the noise variance**.

---

4. Mathematical Expression

The variance of estimated coefficients:

$$\mathrm{Var}(\hat{\theta}) = \sigma^2 (X^T X)^{-1}$$

- As $n \to \infty$:
  - $X^T X$ grows larger → $(X^T X)^{-1} \to 0$.
  - $\mathrm{Var}(\hat{\theta}) \to 0$.
  - $So,\ \hat{\theta} \to \theta^*$.
- Out-of-sample error converges to:

  $E_{out} \to \sigma^2$

This σ^2 is called the **irreducible error** — the noise inherent in the system that no model can remove.

---

5. Analogy

Think of regression as trying to **guess the average height of people in a city**:

- If you sample **5 people**, your estimate fluctuates wildly.
- If you sample **5000 people**, the law of large numbers ensures your estimate converges to the true population mean.
- Similarly, regression coefficients stabilize as data grows.

---

6. Key Takeaways

- With small data → error is high due to noise + unstable coefficients.

- As data increases, coefficients stabilize and error decreases.
- Eventually, error converges to the **irreducible noise level (σ^2)**.
- More data **cannot reduce error below** σ^2, since noise is unavoidable.
- This explains why **adding data improves generalization** (up to the noise limit).

---

2. Key Takeaways

- Regression solution = $\hat{\theta} = (X^T X)^{-1} X^T y$.
- Noise in data propagates into coefficient estimates.
- Variance of coefficients depends on:
  - Noise level (σ^2)
  - Conditioning of feature matrix ($X^T X$)
- With enough data, out-of-sample error converges to noise variance.
- Generalization improves as:
  - $n \to \infty$(more data)
  - Features are less correlated (well-conditioned design matrix).

Data Complexity vs Model Complexity (Guidelines for Number of Data Points vs Features)

1. Detailed Conceptual Overview

In regression (and ML broadly), the balance between the **amount of data available (data complexity)** and the **number of features used (model complexity)** is crucial.

If the model is too complex for the amount of data, it **overfits**.

If the model is too simple for the data, it **underfits**.

This topic lays out **practical guidelines** for how many data points are needed relative to the number of features.

---

a) Dimensionality and Sample Size

- Suppose we have dd features (input variables).
- To estimate regression coefficients θ:
  - We need at least dd data points to find a unique solution.
  - In practice, we need **much more than dd** for stable generalization.

---

b) Rules of Thumb

1. **Underdetermined Case (n < d):**
   - More features than data points.
   - Infinite possible regression solutions → unstable coefficients.
   - Overfitting is guaranteed.
2. **Barely Determined (n ≈ d):**
   - Unique solution exists (if X is full rank).
   - But extremely sensitive to noise.
   - Poor generalization.
3. **Overdetermined Case (n >> d):**
   - Many more data points than features.
   - Regression finds stable coefficients.
   - Best generalization.

---

c) Practical Guidelines in ML

- **At least 10× more data points than features** is a common heuristic for linear regression.
  - Example: If d = 20 features, aim for n $\geq$ 200 samples.
- If dataset is smaller, consider:

- **Regularization** (Ridge, Lasso).
- **Feature selection** (removing irrelevant predictors).
- **Dimensionality reduction** (PCA, etc.).

---

d) Complexity Tradeoff

- **More features (higher d):**
  - Model becomes more flexible → fits data better.
  - But risks overfitting → higher variance.
- **More data points (higher n):**
  - Reduces variance of coefficients.
  - Improves generalization.
- Goal = Find the right balance between data and features.

---

2. Key Takeaways

- Regression stability depends on **ratio of data points (n) to features (d)**.
- **n < d:** Underdetermined, infinite solutions, overfitting.
- **n ≈ d:** Unique solution, but unstable due to noise.
- **n >> d:** Stable, generalizable regression.
- **Rule of thumb:** $n \geq 10d$ for reliable regression.
- If data is limited → use **regularization, feature selection, or dimensionality reduction** to control complexity.

Statistical Inference in Regression (t-statistic and p-value)

1. Detailed Conceptual Overview

So far, regression has given us **coefficients ($\hat{\theta}$)** that describe the relationship between features and the target. But how do we know **which coefficients are truly meaningful**?

This is where **statistical inference** comes in — it helps us move from *just fitting a model* to *evaluating whether the model's parameters are significant or just noise*.

The two key tools here are:

- **t-statistic:** Measures how strongly a coefficient differs from zero (relative to its variability).
- **p-value:** Quantifies the probability that the observed effect could have occurred by random chance.

Together, these allow us to test hypotheses about the importance of features.

---

a) Standard Error of Coefficients

- When we estimate a regression coefficient $\hat{\theta}_j$, it is not exact.
- Different samples (drawn from the same population) would give slightly different $\hat{\theta}_j$.
- The **Standard Error (SE)** tells us how much $\hat{\theta}_j$ tends to vary across repeated samples.
- Think of it as the "uncertainty" or "spread" of our estimated coefficient.

So:

- **Small SE** → coefficient estimate is stable, reliable.
- **Large SE** → coefficient estimate is noisy, less trustworthy.

Mathematically:

$$SE(\hat{\theta}_j) = \sqrt{\sigma^2 \cdot \left[(X^T X)^{-1}\right]_{jj}}$$

where:

- σ^2 = estimated variance of noise,
- $(X^T X)^{-1}_{jj}$ = diagonal element corresponding to feature j.

Where Does SE Come From?

In linear regression:

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

If the true data-generating process is

$y = X\theta^* + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I),$

then the variance of the estimated coefficients is:

$\mathrm{Var}(\hat{\theta}) = \sigma^2 (X^T X)^{-1}$

---

b) Hypothesis Testing for Coefficients

For each coefficient θ_j:

- **Null hypothesis H_0:** θ_j = 0 (feature has no effect).
- **Alternative hypothesis H_1:** $\theta_j \neq$ 0 (feature is important).

We want to know if the data provides enough evidence to reject H_0.

---

c) t-Statistic

The **t-statistic** is defined as:

$t_j = \frac{\hat{\theta}_j}{SE(\hat{\theta}_j)}$

Interpretation:

- A large t_j (positive or negative) means the coefficient is many standard errors away from zero → strong evidence against H_0.
- A small t_j means the coefficient could plausibly be zero.

---

d) p-Value

The **p-value** is a probability. Specifically, it answers this question:

> If the null hypothesis is true (coefficient = 0, no effect), how likely is it that we would observe a coefficient estimate as extreme (or more extreme) than the one we got?

So:

- **Small p-value** → It would be very unlikely to see such a strong coefficient if the feature truly had no effect → Evidence *against* the null → Feature is significant.
- **Large p-value** → The observed effect could easily happen by chance noise → No evidence that the feature matters.
- If p < α (commonly α = 0.05): Reject H_0, feature is statistically significant.
- If p $\geq \alpha$: Fail to reject H_0, the feature is not significant.

Example interpretations:

- p = 0.001: Strong evidence that the feature matters.
- p = 0.3: Weak evidence, feature likely unimportant.

Common Misconceptions

❌ **"A small p-value means the effect is large."**

- Wrong. It means the effect is unlikely to be zero, but it doesn't measure *how big* it is.

❌ **"A p-value tells us the probability that the null hypothesis is true."**

- Wrong. The p-value is computed **assuming the null is true**. It tells us how surprising the data would be under that assumption.

✅ **Correct interpretation:**

- The p-value measures the compatibility of the data with the null hypothesis.

---

e) Why We Need This

- **Not all coefficients matter:** Some may appear large due to noise, but not truly significant.
- **Helps with feature selection:** By checking significance, we can identify which predictors add real value.
- **Avoids over-interpretation:** Prevents us from drawing conclusions based on random fluctuations in data.

---

f) Example Scenario

Imagine a regression predicting **Savings** from features: Salary, Age, Education, Inherited Wealth.

- **Salary:** Coefficient = 0.8, SE = 0.1 → t = 8, p < 0.001 → Highly significant.
- **Education:** Coefficient = 0.05, SE = 0.08 → t = 0.62, p = 0.54 → Not significant.
- **Inherited Wealth:** Coefficient = 0.4, SE = 0.15 → t = 2.67, p = 0.008 → Significant.

From this, we conclude: Salary and Inherited Wealth are strong predictors, while Education has little effect after accounting for others.

---

2. Key Takeaways

- Regression coefficients come with **uncertainty**, captured by **standard errors**.
- **t-statistic** compares the coefficient size relative to its variability.
- **p-value** quantifies the probability of seeing such a coefficient if the true effect were zero.
- **Small p-values (< 0.05)** → feature is significant.
- **Large p-values** → feature likely not important.
- Statistical inference helps us distinguish **real predictors from noise**, guiding better model interpretation and feature selection.
