Final Project Submission: Statistical Estimation through SVD Analysis

Due: 2/25/2025

Overview

The final submission builds on your work from Milestones 1 and 2, completing the implementation and analysis of SVD-based statistical estimation. You'll demonstrate both theoretical understanding and practical implementation skills.

Final Deliverables

1. Complete Implementation (30%)

- Full SVD and PCA implementations
- Comparison of direct SVD vs sklearn PCA
- Analysis of explained variance ratios
- Component selection analysis

2. Analysis and Results (30%)

- Scree plot analysis and interpretation
- Component selection methodology comparison:
  - Elbow method
  - Percentage of variance explained
  - Kaiser criterion
- Convergence analysis with sample size
- Dimensionality effects study

3. Connection to Fundamental Theorem (20%)

- Explain how the four fundamental subspaces relate to covariance analysis:
  - Row Space: Directions of non-zero variance in feature space

  - Column Space: Span of possible covariance combinations
  - Null Space: Directions of zero variance (linear dependencies)
  - Left Null Space: Orthogonal complement to sample covariance structure
- Demonstrate how FTLA dimensions apply to covariance matrices:
  - dim(Row) = dim(Col) = rank
  - dim(Null) + rank = n
  - Relationship to sample size and feature count
- Analyze implications for estimation:
  - How sample size affects rank
  - When covariance matrix becomes singular
  - Connection to principal components

4. Presentation and Documentation (20%)

- Final presentation slides
- Technical documentation
- Code demonstration
- Analysis results visualization

Technical Requirements

Code Structure

```
# Final implementation structure
def analyze_explained_variance(X: np.ndarray) -> Tuple[np.ndarray, List[float]]:
    """
    Analyze explained variance ratios

    Args:
        X: Data matrix (n_samples, n_features)
    Returns:
        Cumulative explained variance, ratios
    """
    pass

def create_scree_plot(explained_variance_ratios: np.ndarray) -> None:
```

```
    """
    Create and save scree plot

    Args:
        explained_variance_ratios: Array of variance ratios
    """
    pass

def compare_component_selection_methods(
    X: np.ndarray,
    explained_variance_ratios: np.ndarray
) -> dict:
    """
    Compare different component selection methods

    Args:
        X: Data matrix
        explained_variance_ratios: Variance ratios
    Returns:
        Dictionary with results from each method
    """
    pass
```

Analysis Requirements

1. Scree Plot Analysis
   - Clear visualization
   - Interpretation of results
   - Justification of choices
2. Component Selection
   - Implementation of multiple methods
   - Comparison of results
   - Recommendations with justification
3. Convergence Analysis
   - Sample size effects

- Stability analysis
- Error bounds

Presentation Requirements

1. Slides (15-20 minutes)
    - Theoretical foundation
    - Implementation approach
    - Key results
    - Conclusions
2. Technical Documentation
    - Mathematical derivations
    - Implementation details
    - Analysis methodology
    - Results interpretation

Evaluation Criteria

- Technical correctness
- Analysis depth
- Presentation clarity
- Documentation quality

Submission Guidelines

1. Code files
    - Well-documented implementations
    - Test cases
    - Example usage
2. Analysis document
    - Methodology description
    - Results presentation
    - Interpretation discussion
3. Presentation materials
    - Slides

- Demo code
- Visualizations

Tips for Success

- Start with clear visualizations
- Compare methods systematically
- Document all decisions
- Practice presentation
- Test with various datasets