# Chapter 7: Moving Beyond Linearity - Expanded Notes

## Introduction

Linear models offer simplicity and interpretability but are limited in their ability to capture complex relationships between predictors and responses. This chapter explores methods that extend beyond linear models to fit non-linear relationships while maintaining a balance between flexibility and interpretability.

The techniques covered include:

1. Polynomial regression
2. Step functions
3. Basis functions
4. Regression splines
5. Smoothing splines
6. Local regression
7. Generalized additive models (GAMs)

Each approach offers different trade-offs between flexibility, interpretability, and computational complexity. The chapter presents these methods as part of a progression that builds from simple extensions of linear models to sophisticated non-parametric approaches.

## 1. Polynomial Regression

Polynomial regression is one of the simplest ways to extend linear regression to model non-linear relationships.

### Core Concept

The standard linear model:

Is replaced with a polynomial function:

### Implementation and Estimation

Despite modeling non-linear relationships, polynomial regression remains a linear model in terms of its parameters (the β coefficients). The predictors are transformations of the original variable X, and the model can be estimated using ordinary least squares.

### Key Considerations

* **Degree selection**: While higher-degree polynomials (larger values of d) can produce extremely flexible fits, degrees higher than 3 or 4 are uncommon in practice.
* **Boundary behavior**: High-degree polynomials can exhibit erratic behavior near the boundaries of the predictor range.
* **Coefficient interpretation**: Individual coefficients in polynomial regression are typically less interpretable than in simple linear regression. The focus shifts to the overall shape of the fitted curve.
* **Statistical inference**: Standard linear model techniques can be used to estimate coefficient variances and compute confidence intervals for the fitted function.

### Advantages and Limitations

**Advantages:** - Simple extension of linear regression - Easily implemented with existing software - Global model with smooth transitions

**Limitations:** - High-degree polynomials can overfit and produce unrealistic predictions, especially at the boundaries - Single global polynomial may not adequately capture complex patterns - Difficult to interpret coefficients directly

## 2. Step Functions

Step functions offer a different approach by converting continuous predictors into discrete intervals, fitting a separate constant within each interval.

### Core Concept

The range of predictor X is divided into K distinct intervals using K+1 cutpoints (c₁, c₂, ..., cₖ). A new set of binary variables is created, where each indicates whether X falls within a specific interval:

Where I(·) is an indicator function returning 1 if the condition is true and 0 otherwise.

The model then becomes:

### Implementation and Interpretation

* Only one indicator variable is "active" (equals 1) for any given value of X.
* The coefficient β₀ represents the mean response when X < c₁.
* Each βⱼ represents the mean difference in response between region j and the reference region (X < c₁).

### Key Considerations

* **Cutpoint selection**: The choice of cutpoints can significantly impact the model's performance. They may be chosen based on domain knowledge, quantiles of the data, or through formal optimization approaches.
* **Number of intervals**: More intervals increase flexibility but risk overfitting.
* **Discontinuities**: The resulting function is discontinuous at the cutpoints, which may be unrealistic for many natural processes.

### Advantages and Limitations

**Advantages:** - Simple to implement and interpret - Can capture threshold effects where response changes abruptly - Makes no assumptions about the functional form within each interval

**Limitations:** - Creates discontinuities at cutpoints - Lacks smoothness, which may be unrealistic for many natural processes - Inefficient use of data—relationships within intervals are ignored

## 3. Basis Functions

Both polynomial regression and step functions are special cases of a more general approach using basis functions, which provides a flexible framework for non-linear modeling.

### Core Concept

Instead of using X directly in the model, we apply a set of fixed, known transformations or basis functions to X:

Where each bⱼ(X) is a basis function applied to the predictor X.

### Examples of Basis Functions

* **Polynomial basis**: bⱼ(x) = xʲ
* **Step function basis**: bⱼ(x) = I(cⱼ ≤ x < cⱼ₊₁)
* **Spline basis**: More complex functions that enforce smoothness constraints

### Key Considerations

* The choice of basis functions determines the flexibility and shape of the fitted curve.
* The number of basis functions controls the model's complexity.
* Since this approach reduces to a linear model once the basis functions are computed, all standard linear model inference tools (standard errors, confidence intervals, hypothesis tests) remain applicable.

### Theoretical Significance

The basis function approach provides a unified framework for understanding many non-linear modeling techniques. It highlights that once appropriate transformations of the predictors are defined, the problem returns to familiar linear model territory.

## 4. Regression Splines

Regression splines combine the flexibility of polynomials with the local adaptability of step functions, addressing limitations of both approaches.

### Core Concept

A regression spline is a piecewise polynomial function with pieces joined at points called knots. The polynomial pieces are constrained to ensure the function is continuous and smooth at the knots.

### Construction

1. Divide the range of X into regions using K knots.
2. Fit a separate polynomial within each region.
3. Constrain the polynomials to ensure continuity at the knots.
4. For higher-degree splines, additional constraints ensure continuity of derivatives at the knots.

### Formal Definition

A degree-d spline is a piecewise polynomial of degree d with continuity in derivatives up to degree d-1 at each knot.

For a cubic spline (d=3), this means: - The function is continuous at each knot - The first derivative is continuous at each knot - The second derivative is continuous at each knot

### Mathematical Representation

A cubic spline with K knots can be represented using basis functions:

A common basis representation for cubic splines includes: - The polynomial basis (X, X², X³) - Truncated power basis functions: (x - ξ)³₊, where ξ is a knot and the "+" subscript indicates the function equals zero when x ≤ ξ.

### Degrees of Freedom

A cubic spline with K knots uses K+4 degrees of freedom: - 4 degrees for the cubic polynomial - K additional degrees for the K knots

### Natural Splines

Natural splines add boundary constraints to regression splines, forcing the function to be linear in the boundary regions (beyond the outermost knots). This reduces variance at the boundaries where data is typically sparse.

### Knot Selection

Several approaches exist for selecting knot locations: - Uniform spacing across the range of X - Placement at quantiles of the data distribution (ensuring equal numbers of observations between knots) - Adaptive placement with more knots in regions with more complex patterns - Cross-validation to determine optimal number and placement

### Advantages and Limitations

**Advantages:** - More flexible than single polynomials - Maintains smoothness, unlike step functions - Can adapt to local features by placing knots strategically - Behaves better at boundaries than high-degree polynomials

**Limitations:** - Requires choices about degree, number of knots, and knot placement - Can still suffer from high variance at boundaries (unless natural splines are used) - Complexity increases with number of knots

## 5. Smoothing Splines

Smoothing splines take a fundamentally different approach to fitting non-linear functions by directly addressing the bias-variance tradeoff through penalized optimization.

### Core Concept

Instead of pre-specifying knots, smoothing splines emerge from an optimization problem that balances fit to the data against function smoothness:

This consists of: - Loss term: Residual sum of squares (RSS) - Penalty term: Integral of squared second derivative (roughness penalty) - Tuning parameter: λ controls the tradeoff between fit and smoothness

### Interpretation of the Penalty Term

* The second derivative g''(t) measures the "curvature" or "wiggliness" of g at point t.
* When g''(t) is large in absolute value, the function is changing rapidly at t.
* When g''(t) is close to zero, the function is nearly linear at t.
* The integral ∫g''(t)²dt measures total curvature across the entire range.

### Role of the Tuning Parameter

* λ = 0: No penalty for roughness, resulting in an interpolating function (passes through all data points).
* λ → ∞: Infinite penalty for roughness, resulting in a linear least squares fit.
* Intermediate values of λ produce progressively smoother functions as λ increases.

### Mathematical Properties

A smoothing spline g(x) has special properties: 1. It is a natural cubic spline with knots at all unique values of the training data. 2. Unlike regression splines, it's not directly constructed from basis functions but emerges from the optimization. 3. It can be shown that the solution to the optimization problem is indeed a natural cubic spline.

### Effective Degrees of Freedom

Rather than counting parameters, smoothing splines are characterized by "effective degrees of freedom" (df\_λ):

Where S\_λ is the "smoother matrix" that maps the response vector y to the fitted values.

* As λ increases from 0 to ∞, df\_λ decreases from n to 2.
* df\_λ need not be an integer.
* df\_λ provides a more intuitive measure of model complexity than λ.

### Selecting the Smoothing Parameter

Cross-validation can be used to select an optimal value of λ (or equivalently, df\_λ). For smoothing splines, leave-one-out cross-validation can be computed efficiently:

This formula avoids refitting the model n times, making cross-validation computationally feasible.

### Advantages and Limitations

**Advantages:** - Automatically places knots at data points, adapting to data density - Explicit control over smoothness through λ or effective degrees of freedom - Elegant mathematical formulation that directly addresses the bias-variance tradeoff - Efficient cross-validation for tuning parameter selection

**Limitations:** - Computationally intensive for large datasets (requires n knots) - Less interpretable than simpler models - May overfit in sparse regions unless constrained (as with natural splines)

## 6. Local Regression

Local regression (also known as locally estimated scatterplot smoothing or LOESS) takes a different approach by fitting models locally using weighted subsets of the data.

### Core Concept

Instead of fitting a global model, local regression fits separate models in local neighborhoods around each point where prediction is desired:

1. For each target point x₀:
2. Select k nearest neighbors (the span).
3. Assign weights based on distance from x₀.
4. Fit a weighted regression model using only these neighbors.
5. Use this model to make a prediction at x₀.

### Algorithm

1. Select the span size s = k/n, the fraction of training points to include in each local neighborhood.
2. For each point x₀ where prediction is needed:
3. Find the k training points closest to x₀.
4. Assign weights K\_i0 to each nearby point, with weight decreasing as distance from x₀ increases.
5. Fit a weighted least squares regression by minimizing:
6. The fitted value at x₀ is:

### Key Parameters

1. **Span size (s)**: Controls the neighborhood size and thus the smoothness of the fit:

* Smaller span: More flexible, potentially more variable fit
* Larger span: Smoother, potentially more biased fit

1. **Weighting function (K)**: Determines how influence decreases with distance:

* Common choice: Tri-cube function (smooth decrease to zero at the boundary)
* Alternative: Gaussian kernel or other kernel functions

1. **Degree of local fit**: The polynomial degree used in each local regression:

* Degree 0: Locally constant (weighted average)
* Degree 1: Locally linear (most common)
* Degree 2: Locally quadratic (more flexible)

### Computational Considerations

Local regression is "memory-based" because the entire training dataset must be retained for making predictions. This contrasts with methods that estimate a fixed set of parameters and discard the training data.

### Extensions to Multiple Predictors

Local regression can be extended to multiple dimensions, but suffers from the curse of dimensionality as the number of predictors increases: - Local neighborhoods become less "local" in high dimensions - More data is needed to maintain the same level of local density - Practical applications typically limited to 2-3 predictors

### Varying Coefficient Models

A useful extension is varying coefficient models, where some coefficients are constant while others vary with certain predictors:

Where T is a variable that influences the coefficients, and each βⱼ(T) is a function rather than a constant. This can be useful for: - Time-varying effects in longitudinal data - Spatial variation in regression coefficients - Interaction effects with continuous moderators

### Advantages and Limitations

**Advantages:** - Highly adaptive to local patterns in the data - Makes few assumptions about global functional form - Can handle complex, non-monotonic relationships - Easy to understand conceptually

**Limitations:** - Computationally intensive, especially for large datasets - Sensitive to outliers in sparse regions - Suffers from curse of dimensionality with multiple predictors - Requires careful selection of tuning parameters

## 7. Generalized Additive Models (GAMs)

Generalized Additive Models extend the flexibility of the previous methods to multiple predictors while preserving interpretability through an additive structure.

### Core Concept

GAMs replace the linear terms in a multiple regression model with flexible non-linear functions:

From:

To:

Where each fⱼ is a non-linear function of a single predictor, estimated from the data.

### Key Properties

1. **Additivity**: Effects of different predictors are modeled separately and then added together.
2. **Flexible function representation**: Each fⱼ can be represented using any of the non-linear techniques (splines, local regression, etc.).
3. **Function-specific smoothing**: Different levels of smoothness can be applied to different predictors.
4. **Visual interpretability**: Each function can be plotted separately to understand its effect.

### Fitting GAMs

GAMs can be fit using different approaches:

1. **Backfitting algorithm**: An iterative procedure that updates each function in turn while holding others fixed:
2. Initialize: fⱼ(X) = 0 for all j
3. Cycle through predictors, updating each fⱼ by:
   * Compute partial residuals: r\_i = y\_i - β₀ - ∑*{k≠j} f\_k(x*{ik})
   * Fit fⱼ to these residuals using any smoothing method
4. Repeat until convergence
5. **Penalized likelihood**: With appropriate basis expansions, GAMs can be expressed as penalized regression problems and solved directly.

### Incorporating Interactions

While additivity is a key feature of GAMs, interactions can be incorporated:

1. **Parametric interactions**: Adding standard interaction terms like x\_j × x\_k
2. **Non-parametric interactions**: Adding bivariate smoothers f\_jk(x\_j, x\_k) for selected variable pairs
3. **Semi-parametric approach**: Combining parametric and non-parametric components

### Extension to Classification

GAMs can be extended to classification problems by replacing the linear predictor in logistic regression with an additive model:

This preserves the interpretability advantages while allowing non-linear effects on the log-odds scale.

### Advantages and Limitations

**Advantages:** - Balance between flexibility and interpretability - Can identify complex non-linear relationships - Modular approach—different smoothing methods can be used for different predictors - Model components can be examined individually - Can incorporate parametric and non-parametric components

**Limitations:** - Additivity assumption may miss important interactions - More complex to fit than standard models - Inference more complicated than for parametric models - Potential for overfitting if smoothness is not properly controlled

## Conclusion: Comparing Non-Linear Methods

When selecting among these non-linear methods, several considerations are important:

### Interpretability vs. Flexibility Trade-off

* **More interpretable**: Polynomial regression, step functions, additive models
* **More flexible**: Smoothing splines, local regression with small span

### Computational Requirements

* **Lower computational cost**: Polynomial regression, regression splines with few knots
* **Higher computational cost**: Local regression, smoothing splines with large datasets

### Ease of Implementation

* **Easier to implement**: Polynomial regression, regression splines
* **More complex**: Smoothing splines, GAMs with interactions

### Statistical Properties

* **Inference well-developed**: Regression splines, polynomial regression
* **Inference more challenging**: Local regression, complex GAMs

### Practical Considerations

1. **Sample size**: For small samples, simpler methods like polynomial regression may be preferred to avoid overfitting.
2. **Number of predictors**: With many predictors, GAMs become essential for manageable non-linear modeling.
3. **Noise level**: Higher noise may warrant smoother fits (larger λ in smoothing splines, larger span in local regression).
4. **Domain knowledge**: Prior understanding of the functional relationships can guide the choice of method.

### Model Selection Approach

Regardless of the chosen method, cross-validation remains the most reliable approach for selecting tuning parameters that control the flexibility of the fit. This ensures an appropriate balance between underfitting and overfitting for the specific dataset at hand.

Most modern statistical software implements all these methods, making it feasible to compare different approaches and select the one that best balances predictive accuracy, interpretability, and computational requirements for a given application.