Improving and Extending Linear Models

Shyue Ping Ong

Aiiso Yufeng Li Family Department of Chemical and Nano Engineering
University of California, San Diego
http://materialsvirtuallab.org

Overview

- Preliminaries
- Improving on linear models
  - Subset selection
  - Shrinkage
  - Derived input directions
- Extending linear methods
- Transformation of inputs
- Piece-wise polynomials
- Gaussian basis functions
- Wavelet and Fourier basis functions

Preliminaries

Preliminaries

- In this lecture, we will look at various approaches to improving and extending simple linear models.
- It is important to note that techniques and concepts such as regularization, shrinkage and transformation of inputs are general and extend to other models.

Improving on linear models

Feature selection

- Often, we want to improve on the least squares model.
  - To improve prediction accuracy by sacrificing some bias for reduced variance.
  - To improve interpretability by reducing number of features or descriptors.
- Three main approaches:
  1. Subset selection
  2. Shrinkage methods
  3. Dimension reduction

Subset selection

Best subset selection

- Brute force approach.
- From p parameters, find the subset of k parameters that results in the smallest RSS.
- Combinatorially expensive for large p and large k.
- Note that the best subset for a larger k does not necessarily include the best subset for a smaller k.

Forward- or backward-stepwise selection

- Forward: Start with intercept, and iteratively add feature that most improves the fit.
- Backward: Start with full model, and sequentially deletes the feature with least impact on the fit.

Shrinkage methods

- Subset methods is discrete, i.e., retains/discards variables, and tends to exhibit high variance.
- Shrinkage methods are more continuous and do not suffer as much from high variability.
- Basic concept: instead of finding the parameters that minimizes the RSS only, we add a penalty term that penalizes more complex models, e.g., models with larger coefficients or larger number of coefficients. This “shrinks” the coefficients, in some cases, to 0.

Ridge regression (L_2 regularization)

$$\beta^{\hat{ridge}} = \underset{\beta}{\text{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

- $\lambda \geq 0$ is the shrinkage parameter. The larger the λ, the greater the shrinkage.
- Also equivalent to:

$$\beta^{\hat{ridge}} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j)^2$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 \leq t$$

Ridge regression - Key details

- Intercept (β_0) is not part of penalty term.
- Inputs should be scaled prior to performing ridge regression, typically by centering to the mean and scaling to unit variance:

$$z_j = \frac{x_j - \mu_{x_j}}{s_{x_j}}$$

LASSO (L_1 regularization)

$$\beta^{L\hat{A}SSO} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

- Least Absolute Shrinkage and Selection Operator
- $\lambda \geq 0$ is the shrinkage parameter. The larger the λ, the greater the shrinkage.
- Also equivalent to:

$$\beta^{L\hat{A}SSO} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j)^2$$

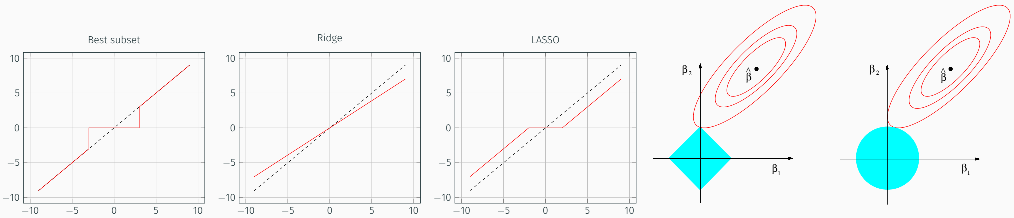
$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t$$

LASSO regression - Key details

- Intercept (β_0) is not part of penalty term.
- Inputs should be scaled prior to performing lasso regression, just as in ridge regression.

Subset vs ridge vs LASSO

- Consider a set of orthonormal features.
    - Ridge: proportional shrinkage. No coefficients are set to zero.
    - LASSO: "soft" thresholding. Translates coefficients by a factor, truncating at zero.
    - Best-subset: "hard" thresholding. Drops all coefficients below a certain threshold.



Other variants of shrinkage methods

- Elastic net penalty:

$$\lambda \left(\alpha \sum_{j=1}^{p} \beta_j^2 + (1 - \alpha) \sum_{j=1}^{p} |\beta_j| \right)$$

- Least angle regression

Derived input directions

- General concept: transforms input $\mathbf{X}$ into a smaller subset of $\mathbf{z_m}$ and regress on $\mathbf{z_m}$
- Principal component regression:
    - Transform non-orthonormal features into orthonormal directions using Principal Component Analysis (PCA).
    - Choose M directions that have the highest eigenvalues (explains the most variance) and discards the rest.
    - Will revisit at a later lecture.

Partial Least Squares (PLS)

- Algorithm:
  1. Compute $\phi_{1j} =< \mathbf{x_j}, \mathbf{y} >$ for each j.
  2. First transformed direction $\mathbf{z_1} = \sum_j \phi_{1j}\mathbf{x_j}$, i.e., each direction is weighted by strength of effect on $\mathbf{y}$.
  3. Regress $\mathbf{y}$ on $\mathbf{z_1}$ to obtain θ_1, orthogonalize $\mathbf{x_1}, \ldots \mathbf{x_p}$ wrt $\mathbf{z_1}$ via $x_j' = x_j - \frac{<\mathbf{z_1}, \mathbf{x_j}>}{<\mathbf{z_1}, \mathbf{z_1}>}\mathbf{z_1}$.
  4. Repeat until $M \leq p$ coefficients are obtained.
- Finds directions with high variance and high correlation with response.

Extending linear methods

Preliminaries

- It is highly unlikely that the true function $f(X)$ is linear in X.
- In some cases, linearity is a reasonable assumption, e.g., a first order Taylor series expansion:

$$f(x) = f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2!} + f'''(a)\frac{(x-a)^3}{3!} + \ldots$$

- Examples where this is used in materials science - linear elasticity (Hooke's law), etc.
- More frequently, we perform a transformation of inputs to create a linear basis expansion.

Transformation of inputs

General concept

- Express:

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$$

  where h_m is the m^{th} transformation of X.
- This is known as a linear basis expansion in X.
- The key lies in choice of the basis functions h_m.

Examples of basis expansions

- $h_m(X) = X_j^2, h_m(X) = X_iX_j$
  - Polynomial expansion to higher-order Taylor series terms.
  - No. of terms increases exponentially with degree of polynomial. For p variables, we have $O(p^2)$ square and cross-product terms in a quadratic model. For a degree d polynomial, we have $O(p^d)$.
- $h_m(X) = log(X_j), sqrt(X_j), exp(iX_j)$: non-linear transformations in X.
- $h_m(X) = I(L_m \leq X_k < U_m)$: Piece-wise division of regions of X. E.g., cubic splines.
- $h_m(X) = RBF(||X - X_m||)$: radial basis function, e.g., Gaussian.
- Typically, basis functions are used simply to allow a more flexible representation of the data. The basis functions can span a very large (sometimes infinite) set, from which a selection has to be made:
  - Restriction - Truncate the choice of basis functions using some criteria.
  - Selection - Choose basis functions that contribute significantly to the fit.
  - Regularization - Use the whole and/or very large subset and apply

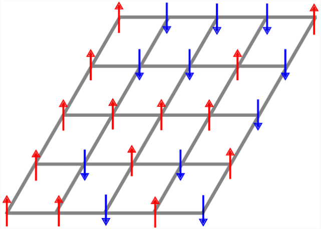
Linearization from physical laws

- Arrhenius law:

$$r = A\,\mathbf{exp}(-\frac{E_a}{RT}) \longrightarrow log(r) = log(A) - \frac{E_a}{RT}$$

- Ising model:

$$H(\sigma) = -\sum_{<i,j>} J_{ij}\sigma_i\sigma_j - \mu\sum_j h_j\sigma_j$$

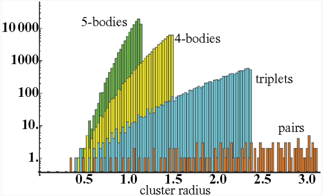


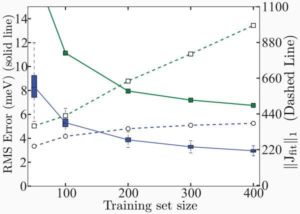
Compressive sensing for cluster expansions

- Cluster expansion of energy on lattice points:

$$H(\sigma) = E_0 + \sum_f J_f \prod_f(\sigma)$$

- σ is the vector representing occupation of lattice sites, $\prod_f$ are the cluster basis functions, J_f are effective cluster interactions (ECIs).
- Compressive sensing: essentially a LASSO to solve for ECIs.[1]

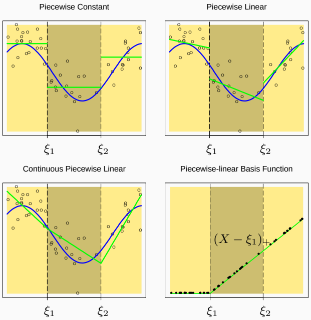




Piece-wise polynomials

Piecewise polynomials

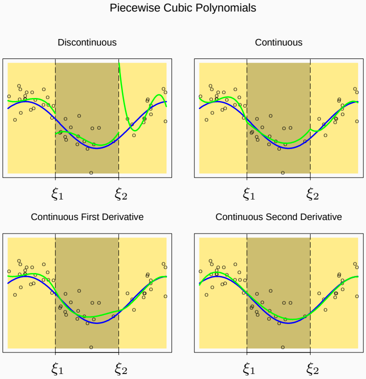
$$h_1(X) = I(X < \xi_1), h_2(X) = I(\xi_1 \leq X < \xi_2), h_3(X) = I(X \geq \xi_2)$$



Parameters:

- No. of knots
- Order of polynomial
- Continuity at knots (value, first derivative, second derivative, etc.). For a polynomial of order N, we usually want all derivatives $< N$ to be continuous.

Cubic splines



- Probably the most commonly used.
- Continuous 1st and 2nd derivatives.
- Natural cubic spline: polynomial is linear beyond boundaries.
- Smoothing spline: Use regularization to control complexity:

$$RSS(f, \lambda) = \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda \int \{f''(t)\}^2 dt$$

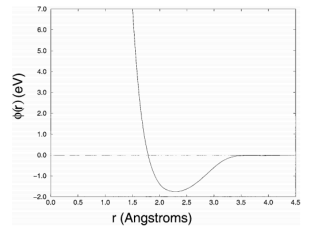
Examples of cubic spline fitting

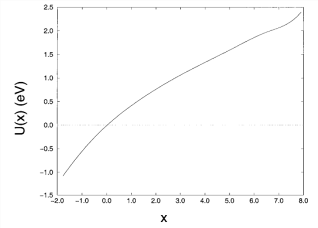
- Spline-based Modified Embedded Atom Method (MEAM)

$$E = \sum_{i<j} \phi(r_{ij}) + \sum_{i} U(n_i),$$

$$n_i = \sum_{j} \rho(r_{ij}) + \sum_{i<k,j,k!=i} f(r_{ij}) f(r_{ik}) g[cos(\theta_{jik})]$$

where ϕ, U, ρ, f and g can be approximated by cubic splines.





Demo: Cubic spline fitting in scipy

```
import numpy as np

## Import CubicSpline from scipy
from scipy.interpolate import CubicSpline

## x, y data for generating the spline fitting
x = np.arange(10)
y = np.sin(x)
## Fit the spline
cs = CubicSpline(x, y)
## Generate new x values
xs = np.arange(-0.5, 9.6, 0.1)
## Perform the interpolation on the new points
ys = cs(xs)
```

Gaussian basis functions

Gaussian basis functions

$$h_m(x) = \exp(-k(x - x_m)^2)$$

- Gaussian functions centered at x_m.
- Other similar types of functions include Lorentzian ($h_m(x) = \frac{1}{1+kx^2}$), Gaussian-Lorentzian, Voigtian, Pearson type IV, and beta profiles.

Example: Rietveld refinement

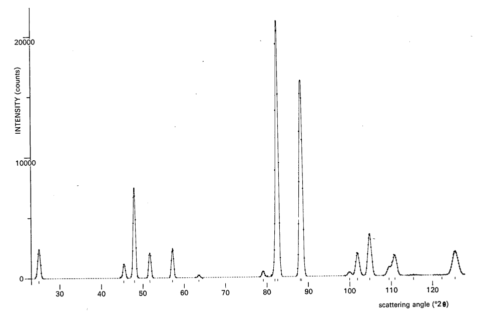


Figure 1: Neutron powder diffraction diagram of $CaUO_4$

- Least squares fitting of theoretical line profile to match a measured diffraction pattern (e.g., X-ray, neutron).[2]

Example: Rietveld refinement, contd.

- Peak shape function:

$$PSF(\theta) = \Omega(\theta) \otimes \Lambda(\theta) \otimes \Psi(\theta) + b(\theta)$$

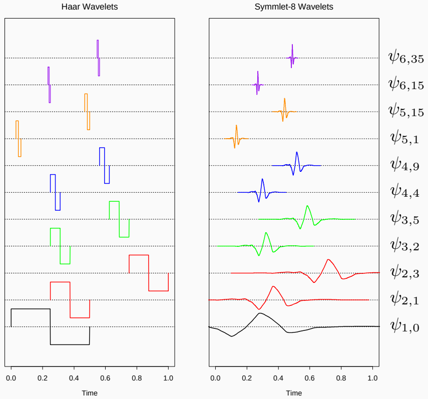
- Ω: Instrument broadening, Λ: Wavelength dispersion, Ψ: Specimen function.
- For single phase, minimize:

$$\Phi = \sum_{i=1}^{N} w_i \left(Y_i^{obs} - \left(b_i + K \sum_{j=1}^{m} I_j y_j(x_j) \right) \right)^2$$

- where $y_j(x_j)$ is typically a pseudo-Voigt (mix of Gaussian and Lorentizan function) function.
- Note that the background (b_i) holds no useful structural information and should be minimized in experiments.

Wavelet and Fourier basis functions

Wavelet smoothing



- Complete orthonormal basis
- Shrink and select toward **sparse** representation.
- Able to represent both time and frequency localization efficiently (Fourier basis can only do frequency localization).

Example: NMR Spectroscopy

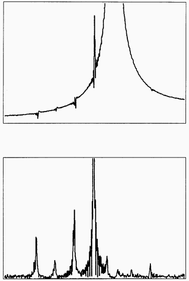
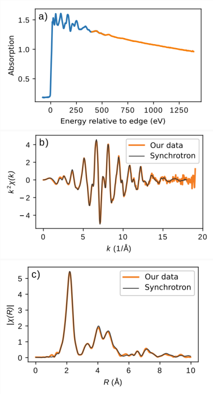


Figure 2: Subtraction of a large spectral line: (top) the original spectrum of polyethylene, (bottom) reconstructed spectrum after removal of CH_2 peak.[3]

Applications:

- Suppression of large unwanted spectral line (left).
- Rephasing spectrum perturbed by time-dependent magnetic field.
- Noise filtering
- Detecting phases in a mixture

Example: Fourier transform for analysis of extended X-ray absorption fine structure (EXAFS)



- (a) The extended edge (orange part) contains information of atom chemical environment.
- (b) Subtract the background, convert energy to k-space unit, and multiply the normalized intensity by k^2
- (c) Fourier transform k-space information to real space and obtain the first shell bond length.

Bibliography i

Lance J. Nelson, Gus L. W. Hart, Fei Zhou, and Vidvuds Ozoliņš.
Compressive sensing as a paradigm for building physics models.
Physical Review B, 87(3):035125, January 2013.

H. M. Rietveld.
A profile refinement method for nuclear and magnetic structures.
Journal of Applied Crystallography, 2(2):65–71, June 1969.

D. Barache, J-P. Antoine, and J-M. Dereppe.
The Continuous Wavelet Transform, an Analysis Tool for NMR Spectroscopy.
Journal of Magnetic Resonance, 128(1):1–11, September 1997.

The End