Functions and Data Fitting
Outline

1. Functions
2. Features
3. Polynomial Fitting: Univariate
   Least Squares Fitting
   Choosing a Degree
4. Polynomial Fitting: Multivariate
5. Limitations of Polynomials
6. The Curse of Dimensionality
Functions Everywhere

- **SPAM**
  
  \[ A = \{ \text{all possible emails} \} \]
  \[ Y = \{ \text{true, false} \} \]
  \[ f : A \rightarrow Y \quad \text{and} \quad y = f(a) \in Y \quad \text{for} \ a \in A \]

- **Virtual Tennis**
  
  \[ A = \{ \text{all possible video frames} \} \subseteq \mathbb{R}^d \]
  \[ Y = \{ \text{body configurations} \} \subseteq \mathbb{R}^e \]

- **Medical diagnosis, speech recognition, movie recommendation**

- **Predictor = Regressor or Classifier**
Classic and ML

- Classic:
  - Design *features* by hand
  - Design $f$ by hand

- ML:
  
  Define $A, Y$
  Collect $T_a = \{(a_1, y_1), \ldots, (a_N, y_N)\} \subset A \times Y$
  Choose $\mathcal{F}$
  Design $\lambda : \{\text{all possible } T_a\} \rightarrow \mathcal{F}$
  *Train*: $f = \lambda(T_a)$
  Hopefully, $y \approx f(a)$ *now and forever*

- Technical: $A$ can be anything. Too difficult to work with.
Features

- From $A$ to $X \subseteq \mathbb{R}^d$

  \[
  \begin{align*}
  \mathbf{x} &= \phi(a) \\
  y &= h(\mathbf{x}) = h(\phi(a)) = f(a)
  \end{align*}
  \]

  \[
  h : X \subseteq \mathbb{R}^d \rightarrow Y \subseteq \mathbb{R}^e
  \]

  \[
  \mathcal{H} \subseteq \{X \rightarrow Y\}
  \]

  \[
  T = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \subseteq X \times Y
  \]

- Just numbers!
Features for SPAM

\[ d = 20,000 \]

\[ \phi \] also useful in order to make \( d \) smaller or \( \textbf{x} \) more informative
Fitting and Learning

- **Loss** $\ell(y, h(x)) : Y \times Y \rightarrow \mathbb{R}^+$
- **Empirical Risk** (ER): average loss on $T$
- **Fitting and Learning:**
  - Given $T \subset X \times Y$ with $X \subseteq \mathbb{R}^d$
    \[ \mathcal{H} = \{ h : X \rightarrow Y \} \text{ (hypothesis space)} \]
  - Fitting: Choose $h \in \mathcal{H}$ to minimize ER over $T$
  - Learning: Choose $h \in \mathcal{H}$ to minimize some risk over previously unseen $(x, y)$
Summary

- Features insulate ML from domain vagaries
- Loss function insulates ML from price considerations
- Empirical Risk (ER) averages loss for $h$ over $T$
- ER measures average performance of $h$
- **A learner picks an** $h \in \mathcal{H}$ **that minimizes some risk**
- Data fitting minimizes ER and stops here
- **ML wants $h$ to do well also tomorrow**
- The risk for ML is on a bigger set
Polynomial Fitting: Univariate

Data Fitting: Univariate Polynomials

\[ h : \mathbb{R} \to \mathbb{R} \]
\[ h(x) = c_0 + c_1 x + \ldots + c_k x^k \]

with \( c_i \in \mathbb{R} \) for \( i = 0, \ldots, k \)

- The definition of the structure of \( h \) defines the hypothesis space \( \mathcal{H} \)
- \( T = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R} \times \mathbb{R} \)
- Quadratic loss \( \ell(y, \hat{y}) = (y - \hat{y})^2 \)
- ER: \( L_T(h) \overset{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, h(x_n)) \)
- Choosing \( h \) is the same as choosing \( \mathbf{c} = [c_0, \ldots, c_k]^T \)
- \( L_T \) is a quadratic function of \( \mathbf{c} \)
Rephrasing the Loss

\[ \text{NL}_T(h) = \sum_{n=1}^{N} [y_n - h(x_n)]^2 = \sum_{n=1}^{N} \{y_n - [c_0 + c_1 x_n + \ldots + c_k x_n^k]\}^2 \]

\[ = \| \begin{bmatrix} y_1 - [c_0 + c_1 x_1 + \ldots + c_k x_1^k] \\ \vdots \\ y_N - [c_0 + c_1 x_N + \ldots + c_k x_N^k] \end{bmatrix} \|^2 \]

\[ = \| \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} - \begin{bmatrix} 1 & x_1 & \ldots & x_1^k \\ \vdots \\ 1 & x_N & \ldots & x_N^k \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_k \end{bmatrix} \|^2 \]

\[ = \| b - Ac \|^2 \]
Linear System in $c$

$c_0 + c_1 x_n + \ldots + c_k x_n^k = y_n$

$$A c = b$$

$$A = \begin{bmatrix} 1 & x_1 & \ldots & x_1^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \ldots & x_N^k \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

- Where are the unknowns?
- Why is this linear?
Least Squares

\[ \hat{c} \in \arg \min_c \| Ac - b \|^2 \]

Thus, we are minimizing the empirical risk \( L_T(h) \) (with the quadratic loss) over the training set.
Choosing a Degree

- Underfitting, overfitting, interpolation
Data Fitting: Multivariate Polynomials

• The story is not very different:

\[ h(x) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1^2 + c_4 x_1 x_2 + c_5 x_2^2 \]

• Polynomial of degree up to 2

\[
A = \begin{bmatrix}
1 & x_{11} & x_{12} & x_{11}^2 & x_{11} x_{12} & x_{12}^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & x_{N1} & x_{N2} & x_{N1}^2 & x_{N1} x_{N2} & x_{N2}^2 \\
\end{bmatrix},
\]

\[
b = \begin{bmatrix}
y_1 \\
\vdots \\
y_N \\
\end{bmatrix}, \quad c = \begin{bmatrix}
c_0 \\
\vdots \\
c_5 \\
\end{bmatrix}
\]

• The rest is the same

• Why are we not done?
Counting Monomials

- Monomial of degree $k' \leq k$ in $d$ variables:
  \[ x_1^{k_1} \ldots x_d^{k_d} \text{ where } k_1 + \ldots + k_d = k' \]

- How many are there?
  \[ m(d, k') = \binom{d + k'}{k'} \]

(See an Appendix for a proof)
Asymptotics: Too Many Monomials

\[ m(d, k) = \binom{d+k}{k} = \frac{(d+k)!}{d!k!} = \frac{(d+k)(d+k-1)\ldots(d+1)}{k!} \]

- \( k \) fixed: \( O(d^k) \)
- \( d \) fixed: \( O(k^d) \)

- When \( k \) is \( O(d) \), look at \( m(d, d) \):

\[ m(d, d) \text{ is } O\left(\frac{4^d}{\sqrt{d}}\right) \]

- Except when \( k = 1 \) or \( d = 1 \), growth is polynomial (with typically large power) or exponential (if \( k \) and \( d \) grow together)

- This difficulty is specific to polynomials
The Curse of Dimensionality

- A large $d$ is typically troublesome
- We want $T$ to be “representative”
- “Filling” $\mathbb{R}^d$ with $N$ samples
  
  $X = [0, 1]^2 \subset \mathbb{R}^2$
  
  10 bins per dimension, $10^2$ bins total

  $X = [0, 1]^d \subset \mathbb{R}^d$
  
  10 bins per dimension, $10^d$ bins total

- $d$ is often hundreds or thousands (SPAM $d \approx 20,000$)
- $10^{80}$ atoms in the universe
- **We will always have too few data points**
- This difficulty is general