Overview
-- What is Machine Learning?
-- Simple toy problems in regression.

Fundamentals
-- Bayesian Inference
-- Regularization and Localization
-- Bias-Variance
-- Model selection and Cross Validation
-- Feature and Input Selection
-- Optimization in Learning

Learning Tasks
-- Regression, Classification, Detection,
-- Recognition, Recommendation
-- Clustering
Deep Learning
   -- 27 different types of networks
   -- Learning algorithms used to train networks
Problem Solving
   -- Reduced Modeling
   -- Parameterization
   -- Data Assimilation
   -- Quantifying Uncertainty
   -- Discovery Machines (Learning + Sampling)
   -- Predictability and Learnability
Advanced (* optional)
   -- Dynamic Data Driven Learning for Geosience
Learn Physics?

• Let’s re-examine the Poisson Equation problem

\[ \nabla^2 \phi = f \]

On a periodic domain
In 1D

\[ \phi = D^{-1} f \]

\[ \phi = Lf \]

Machine Learning problem:
Estimate \( D, L \)

Given: \( \{\phi_n, f_n\}, the training data \)

Sound Easy?
Step 1

How many samples?  Why $L_2$?

$$J(L^\#) := \frac{1}{2N} \sum_{i=1}^{N} \left\| \phi_i - L^\# f_i \right\|_2^2$$

Matrix form of operator $L$

$L^\#$ degrees of freedom: $n^2$
Step 2, OLS

- $J(L^o) := \frac{1}{2N} \sum_{i=1}^{N} \left\| \phi_i - L^o f_i \right\|^2_2$

- $L^o$ has $n$ degrees of freedom;
  block circulant matrix for our problem

Oracle: Let’s invoke invariance as a principle

$L^# \Rightarrow L^o$

- $L^o = \begin{pmatrix} l_1 & \cdots & l_n \\ \vdots & \ddots & \vdots \\ l_2 & \cdots & l_1 \end{pmatrix}$ -- special Toeplitz matrix
Equivalently

\[ J(l) \overset{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} J_i(l); \]
\[ \Rightarrow \hat{l} = (\sum_{i=1}^{N} F_i^T F_i)^{-1} \left( \sum_{j=1}^{N} F_j^T \phi_j \right) \]

- Where, \( l = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix} \) and \( F_i = \begin{pmatrix} f_{i,1} & \cdots & f_{i,n} \\ \vdots & \ddots & \vdots \\ f_{i,2} & \cdots & f_{i,1} \end{pmatrix} \)

- \( J_i(l) \overset{\text{def}}{=} \frac{1}{2} \left\| \phi_i - F_i l \right\|_2^2 \)

- Local solution is: \( \hat{l}_i = (F_i^T F_i)^{-1} F_i^T \phi_i \)

- Can estimate \( \hat{l} = (F_i^T F_i)^{-1} F_i^T \phi_i \) work?

What is the approximation?
What kind of training data?
How many samples N/n?
Regularization

\[ J_i(l) \overset{\text{def}}{=} \frac{1}{2} \| \phi_i - F_i l \|_2^2 + \frac{\lambda}{2} \| l \|_2^2 \]

\[ \hat{l}_i = (F_i^T F_i + \lambda I)^{-1} F_i^T \phi_i \]

\[ F_i = U_i S_i U_i^T \]

\[ U_{\#i} \overset{\text{def}}{=} U_i(\cdot, 1:S < N \ll n) \]

- **Leading S directions**

\[ \hat{\phi}_i = U_{\#i} [S_{\#i}^2 + \lambda I]^{-1} S_{\#i} U_{\#i}^T \phi_i \]

Why L_2? Why L_2? aka Ridge

What does \( \lambda \) do? Why this form? What are they?

How many “leading” dimensions?

Try Matlab Code
Recursion

• \( J(l) \) \( \overset{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{N} \| \phi_i - F_i l \|_2^2 \)

• \( Q_1 = F_1^T F_1 \) and \( \hat{l}_1 = [F_1^T F_1]^{-1} F_1^T [\phi_1] \)

• \( \hat{l}_{i+1} = \hat{l}_i + [Q_i + F_{i+1}^T F_{i+1}]^{-1} F_{i+1}^T [\phi_{i+1} - F_{i+1} \hat{l}_i] \)

• \( Q_{i+1} = Q_i + F_{i+1}^T F_{i+1} \) (cumulant;)

• Assignment 1: Derive this result.
• Compare with batch solution
• Compare with naïve average
• Optional: Compare with Kalman Filter
Estimation Beyond OLS

• Maximum Likelihood Estimation (MLE)
  
  Let \( J_i(l) \) def \( \frac{1}{2} \left\| \phi_i - F_i l \right\|_R^2 = \frac{1}{2} [\phi_i - F_i l]^T R^{-1} [\phi_i - F_i l] \)
  
  then \( \hat{l}_i = (F_i^T R^{-1} F_i)^{-1} F_i^T R^{-1} \phi_i \)

• Maximum A Posteriori Estimation (MAP):
  
  Let \( J_i(l) \) def \( \frac{1}{2} \left\| \phi_i - F_i l \right\|_R^2 + \frac{1}{2} \left\| l - \bar{l} \right\|_C^2 \)
  
  \( \hat{l}_i = (F_i^T R^{-1} F_i + C^{-1})^{-1} [F_i^T R^{-1} \phi_i + C^{-1} \bar{l}] \)

• Can you combine this with regularization? Truncation?

• Why are these called MAP and MLE?
Notes

• The best result did not come by
  • Increasing sample size
  • Using L2 regularization

• It immediately emerged when we chose a “subspace”

• The Kernel ($\nabla^2$) is sparse, so $L^0$ is non-sparse. $U$ is the Fourier Transform (in the continuous case), and therefore truncating $U$ projects the residual onto very low frequencies, promoting sparse $D$.

• What we assumed was
  • Linearity, Invariance, small magnitude perturbations.
Bayesian Perspective

• \( \hat{\beta} = \arg \max_\beta P (\beta | d_1 \ldots d_n); \ d \overset{\text{def}}{=} \{x, y\} \)

• \( P (\beta | d_1 \ldots d_n) = \frac{P(\beta, d_1 \ldots d_n)}{P(d_1 \ldots d_n)} = \frac{P(d_1 \ldots d_n | \beta)P(\beta)}{P(d_1 \ldots d_n)} \)

\[
= \frac{P(d_n | \beta, d_{1:n-1}) P(d_{n-1} | \beta, d_{1:n-2}) \ldots P(d_3 | \beta, d_{1:2}) P(d_2 | \beta, d_1) P(d_1 | \beta)}{P(d_n | d_{1:n-1}) P(d_{n-1} | d_{1:n-2}) \ldots P(d_3 | d_{1:2}) P(d_2 | d_1) P(d_1)}
\]

\( y_i = f(x_i; \beta) + \epsilon_i \)

\( d_i \) is fixed, given

Heteroskedastic
Recursive Expression

- Let \( P(\beta_1) \stackrel{\text{def}}{=} P(\beta_1 | d_1) = \frac{P(d_1 | \beta)P(\beta)}{P(d_1)} = \frac{P(d_1 | \beta)}{\sum_{\beta} P(d_1 | \beta)P(\beta)} \)

- \( P(\beta_2) \stackrel{\text{def}}{=} \frac{P(d_2 | \beta_1)P(\beta_1)}{\sum_{\beta_1} P(d_2 | \beta_1)P(\beta_1)} = \frac{P(d_2 | \beta, d_1)}{P(d_2 | d_1)} \left[ \frac{P(d_1 | \beta)P(\beta)}{P(d_1)} \right] \)

- \( P(\beta_i = b) = \frac{P(d_i | \beta_{i-1} = b)P(\beta_{i-1} = b)}{\sum_{\beta_{i-1}} P(d_i | \beta_{i-1})P(\beta_{i-1})} \)
Solution

• Recursion:

\[
P(\beta | d_{1:n}) = \frac{P(d_n | \beta) P(\beta | d_{1:n-1})}{\sum_{\beta_{n-1}} P(d_n | \beta_{n-1}) P(\beta_{n-1})}
\]

• Are the data \(d_{1:n}\) assumed to be independent samples?
• Are the data \(d_{1:n}\) from identical distributions?
• Is the joint distribution exchangeable, e.g.
  \[P(D_1 = d_1 \ldots D_n = d_n) = P(D_1 = d_n \ldots D_n = d_1)\]?
• Are the random variables \(D_1 \ldots D_n\) conditionally independent, given \(\beta\)?
  • What is the difference between \(P(\beta | d_{1:n-1})\) and \(P(\beta)\)?

• How to solve?
  • \(P(\beta)\) sample \(\rightarrow P(\beta_1)\) sample \(\rightarrow P(\beta_2)\) sample \(\rightarrow P(\beta_3)\) sample \(\rightarrow \ldots \) sample \(\rightarrow P(\beta_N)\)

• Dimensionality a problem for Sampling, with \(\beta\) large.
Conditional Independence

• Why not just write:

\[
P(\beta \mid d_{1:n}) \propto P_n(D_n = d_n \mid \beta) P_{n-1}(D_{n-1} = d_{n-1} \mid \beta) \cdots P_1(D_1 = d_1 \mid \beta) P(\beta)
\]

• \(P(\beta \mid d_1 \ldots d_n) = \frac{\prod_{i=1}^{N} p_i(D_i = d_i \mid \beta) P(\beta)}{\sum_{\beta} \prod_{i=1}^{N} p_i(D_i = d_i \mid \beta) P(\beta)}\)

• i.e. Conditioned on \(\beta\), \(D_1 \ldots D_n\) are independent, but they need not be identical. They may have different variances for example.

• If \(D_1 \ldots D_n\) are also identically distributed, same parameters then \(P(d_1 \ldots d_n)\) is exchangeable

• I.I.D. is a very strong and common assumption!

• Usually only conditional independence is needed in regression (and other problems)
Solving

- $P(\beta \mid d_{1:n}) \propto P(D_n = d_n \mid \beta) P(D_{n-1} = d_{n-1} \mid \beta) \ldots P(D_1 = d_1 \mid \beta) / P(\beta)$

- Sample $\beta \sim P(\beta)$, Evaluate Likelihoods, Infer $P(\beta \mid d_{1:n})$

- This can also be done sequentially because recursion is true here also, but a major convenience even in batch over the previous scheme.
Linear Gaussian Case

• Suppose

\[ P_i \left( D_i = d_i \mid \beta \right) \propto e^{-J_i(\beta)} ; ; ; \]

\[ J_i(\beta) \stackrel{\text{def}}{=} \frac{1}{2} \left\| \Phi_i - F_i \beta \right\|_R^2 \]

• \[ P(\beta) = e^{-\frac{1}{2} \left\| \beta - \overline{\beta} \right\|_C^2} \]

• Thus \[ \arg \max_{\beta} P(\beta \mid d_1 \ldots d_n) = \arg \min_{\beta} - \ln P(\beta \mid d_1 \ldots d_n) \]

• \textbf{MAP}: \[ \sum J_i(\beta) + \frac{1}{2} \left\| \beta - \overline{\beta} \right\| \]

• \textbf{MLE}: \[ \sum J_i(\beta) \]

• \textbf{OLS}: \[ \sum J_i(\beta) ; \text{with } R = I \]  

What Is the statistical Model for OLS?
How general is (Nonlinear) Regression?

- Neural Nets
- Kernel Methods
- Basis Functions
- Many other problems can be thought this way.
- We will return to regression machines later and look at regression from a Bayesian perspective.
Summary

- Regression
  - Cost function, OLS
    - Batch
    - Sequential
  - With Spectral Truncation
  - With Regularization
- Bayesian
  - Sampling posterior sequence
  - Conditional Independence
    - MAP $\rightarrow$ MCMC
    - Gaussian Linear Case $\rightarrow$ Batch
    - MLE $\rightarrow$ OLS $\rightarrow$ Sequential Solution
    - MAP $\rightarrow$ Sequential Solution
- Graphical Models with Conditional Independence
Ex’s and Oh’s and Classification

Loss Functions

https://youtu.be/0uLI6BnVh6w
How Complex?

Model Complexity

Bias-Variance Dilemma
Nearest Neighbors vs Global Regression

• **Regression:** \( y \sim f(x) \); use all training data to estimate parameters of \( f \), then

\[
\text{Classification: } c = \begin{cases} 
1 & \text{if } y > 0.5 \\
0 & \text{otherwise}
\end{cases}
\]

• **Nearest Neighbor regression:** \( y \sim f_x(x) \), using training data \( \{x_i, c_i \mid x_i \in \mathcal{N}_k(x)\} \); then classify

• **Nearest Neighbor classifier:** \( c(x) = g (\{c_i \mid x_i \in \mathcal{N}_k(x)\}) \). The training data is \( \{x_i, c_i\} \)

• Here \( \mathcal{N}_k(x) \) is the set of \( k \) nearest neighbors of \( x \), whose labels \( c_i \) are used by function \( g \) to make a decision. For example \( g \) could be majority vote.
Statistical Model Regression

\[
\mathbf{E}(y - f(x))^2 = \int \int (y - f(x))^2 P(x, y) dx \, dy \\
= \int \left[ \int (y - f(x))^2 P(y|x) dy \right] P(x) dx \\
= \int [\hat{e}(x)]^2 P(x) dx \\
\arg \min_{\hat{f}_x} \mathbf{E}_{y|x} (y - \hat{f}_x)^2 = \arg \min_{\hat{f}} \int (y - \hat{f}_x)^2 P(y|x) dy \\
\Rightarrow \hat{f}_x = \mathbf{E}_{y|x}(y)
\]

Take the conditional mean of \( y \) at \( x \) as the estimate for \( f(x) \)

For nearest neighbor regression: \( \hat{f}_x = \mathbf{E}_{y|\mathcal{N}_k(x)}(y) \); using some neighborhood
Statistical Model for Classification

• The variables have $|C|$ categories, say $C = \{1 \ldots S\}$.

• $L$ is a matrix of dissimilarities, the Loss function, e.g. 0-1 loss function
  
  $$L[c, g(x)] = \begin{cases} 
  0 & \text{if } c = g(x) \\
  1 & \text{otherwise} 
  \end{cases}$$

  hit or miss, 1s off diagonal, 0s on it.

  Expected Prediction Error is:

  $$E_x \left[ \sum_{s=1}^{S} L[s, g(x)] P(s|x) \right]$$

  For 0-1 loss function, solution for $g(x)$ is

  $$\hat{g}(x) = \arg \max_c P(c|x)$$

  Choose the most likely class! ➔ Bayes Classifier
Kernels vs. Nearest Neighbors

\[ y \sim \sum_{i=1}^{N} K_i(x, x_i; \beta_i) f(x_i) = \sum_{i=1}^{N} K_i(x, x_i; \beta_i) y_i \]

Kernels:

\[ K_i(x, x_i) = \frac{w_i}{\sum_j(w_j)}, \text{ where } w_i = \frac{1}{\|x - x_i\|^2 + \epsilon > 0} \]

\[ K_i(x, x_i; \sigma_i) = \frac{1}{Z} e^{-\frac{1}{2}(x-x_i)\sigma_i^{-2} (x-x_i)} \] (normalized)
Kernels and Neural Nets
Bases, and other forms of regression

Bases

\[ y \sim \sum_{b=1}^{B} f_b(x) \beta_b \]

\[ = \left[ f^T \beta \right] (x) \]

Neural Nets (a single internal layer):

\[ y \sim f(W^T[x 1]^T) \]

\[ f(z) = \sum_i \beta_i \left[ \frac{1}{1 + \exp(-z_i)} \right] \]

Localization through Kernels
Basis weights
Regularization
Sparsity
...
Is a key facet of Learning
Regression vs NN

• Regression
  • If a straight line
  • “high bias”

• K-NN
  • Effective degrees of freedom: $\approx \frac{N}{k}$
  • Larger $k \Rightarrow$ more bias, smaller $k \Rightarrow$ more “variance”

• Bias $\rightarrow$ the parameters change little from one training sample to the next, there is a steady, large error between training data

• Variance $\rightarrow$ huge changes in parameters, but always fits training data well.
Bias Variance Decomposition

\[ y = f(x) + n \]

\[ \frac{1}{N} \sum_i \left( y_i - \hat{f}(x_i) \right)^2 = \sum_i y_i^2 + \hat{f}_i^2 - 2y_i\hat{f}_i \]

\[ = \left[ \frac{1}{N} \sum_i y_i^2 - \bar{y}^2 \right] + \left[ \frac{1}{N} \sum_k \hat{f}_k^2 - \bar{f}^2 \right] + \bar{y}^2 - 2\bar{y}\bar{f} + \bar{f}^2 \]

\[ = \left[ \frac{1}{N} \sum_i y_i^2 - \bar{y}^2 \right] + \left[ \frac{1}{N} \sum_k \hat{f}_k^2 - \bar{f}^2 \right] + (\bar{y} - \bar{f})^2 \]

TV = noise + variance + bias^2
Issues: Poor Features, Bad Labels?

Feature Selection

Data Selection

Noisy Labels

These can be common issues

Even generating training data can be difficult
Issues: Model Error and Uncertainty

Uncertainty induced by data labels may imply probabilistic regression and classification is needed.
Dimensionality Issues and K-nearest dearest

• Assume data is uniform on a unit hypercube

• For dimension $d$ capturing a fraction $r$ of neighbors requires $r^\frac{1}{d}$ edge. As $d$ goes up, this is most of the volume

• If $r$ is reduced, then increased variance.

• In a unit ball, the median distance from origin to closest point of $N$ points is $\left(1 - \left(\frac{1}{2}\right)^{\frac{1}{N}}\right)^\frac{1}{d}$ so that as $d$ increases, the points are farther apart.

• Finally the sampling density is $N\frac{1}{d}$, so that keeping the same density requires an enormous number of points with increased dimensionality

• ML in high dimensions is a challenge.
Learning Tasks

**Supervised**
- Training Data \{X,Y\}
- Task: \(Y := f(X)\)
  - \{X,Y\} \xrightarrow{Learning} M
  - \(M(X = x) \Rightarrow Y = \hat{Y}\)

**Unsupervised**
- Training Data \{X\}
- Task: \(O(X) – \text{Organization of } X\).
  - \{X\} \xrightarrow{Learning} O
  - \(O(X = x) \Rightarrow \text{mode/category}\)

**Semi-Supervised:** Some Data have labels and/or requires internal organization
Modeling by Machine Learning

- Physical Model (Use of physical models to make a prediction)
  - + Statistical Model (Use of data to improve prediction)
  - + Stochastic Model (Represent the distribution of residuals/joint probabilities)
A mostly complete chart of Neural Networks

©2016 Fjodor van Veen - asimovinstitute.org
# Machine Learning

<table>
<thead>
<tr>
<th>Learning Principles</th>
<th>Learning Knobs</th>
<th>Major Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invariance</td>
<td>Loss Functions</td>
<td>Prediction</td>
</tr>
<tr>
<td>Parsimony</td>
<td>Regularization</td>
<td>Characterization</td>
</tr>
<tr>
<td>Smoothness</td>
<td>Dimensionality Reduction</td>
<td>Explanation</td>
</tr>
<tr>
<td>Sparsity</td>
<td>Bias-Variance</td>
<td></td>
</tr>
<tr>
<td>Entropy</td>
<td>Model Selection</td>
<td></td>
</tr>
<tr>
<td>Compactness</td>
<td>Feature Selection</td>
<td></td>
</tr>
<tr>
<td>Generalization</td>
<td>Data Selection</td>
<td></td>
</tr>
</tbody>
</table>