Localization and Other Issues

Sai Ravela
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
  
  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 \( \{x_i, c_i \mid x_i \in N_k(x)\} \); then classify

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

- Here \( N_k(x) \) 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

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

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

For nearest neighbor regression: \( \hat{f}_x = \mathbb{E}_{y|\mathcal{N}_k(x)}(y) \); 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

$$L[c, g(x)] = \begin{cases} 0 & \text{if } c = g(x); \text{ hit or miss, 1s off diagonal, 0s on it.} \\ 1 & \text{otherwise} \end{cases}$$

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! $\Rightarrow$ 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:

\[ K_i(x, x_i) = \frac{w_i}{\sum_j (w_j)}, \text{ where } w_i = \frac{1}{\|x - x_i\|_2^2 + \epsilon} > 0 \]

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

Bases

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

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

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 \( k \) ⇒ more bias, smaller \( k \) ⇒ more “variance”

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

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

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

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

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

TV = noise + variance + bias^2

Bias  Variance

complexity \( \frac{N}{k} \)
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 $\frac{1}{r^d}$ 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 $\frac{1}{N^{\frac{1}{d}}}$, so that keeping the same density requires an enormous number of points with increased dimensionality.
- ML in high dimensions is a challenge.
# 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>
Types of Learning

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

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

**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)
<table>
<thead>
<tr>
<th>The Matrix</th>
<th>Regress</th>
<th>Classify</th>
<th>Estimate</th>
<th>Cluster</th>
<th>Sample Selection</th>
<th>Feature Select</th>
<th>Model Complexity</th>
<th>Loss</th>
<th>Regularize</th>
<th>Generalize</th>
<th>Explain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Squares</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-NN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayesian MCMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trees</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Forests</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensemble Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manifold Learning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixtures/Kernels</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression Machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel Machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph Spectra</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>