Machine Learning

Vasant Honavar
Artificial Intelligence Research Laboratory
Informatics Graduate Program
Computer Science and Engineering Graduate Program
Bioinformatics and Genomics Graduate Program
Neuroscience Graduate Program
Data Sciences Undergraduate Program
Center for Big Data Analytics and Discovery Informatics
Huck Institutes of the Life Sciences
Institute for Cyberscience
Clinical and Translational Sciences Institute
Northeast Big Data Hub
Pennsylvania State University

vhonavar@ist.psu.edu
http://faculty.ist.psu.edu/vhonavar
http://ailab.ist.psu.edu
Now, on to some real content …
Classification

• How would you write a program to distinguish a picture of you from a picture of someone else?
  – Provide examples pictures of you and pictures of other people and let a classifier learn to distinguish the two.
• How would you write a program to determine whether a sentence is grammatical or not?
  – Provide examples of grammatical and ungrammatical sentences and let a classifier learn to distinguish the two.
• How would you write a program to distinguish cancerous cells from normal cells?
  – Provide examples of cancerous and normal cells and let a classifier learn to distinguish the two.
Example: To play or not to play tennis

- **Example dataset**

<table>
<thead>
<tr>
<th>Class</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Windy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play</td>
<td>Sunny</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>No play</td>
<td>Sunny</td>
<td>High</td>
<td>Yes</td>
</tr>
<tr>
<td>No play</td>
<td>Sunny</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>No play</td>
<td>Rainy</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Play</td>
<td>Rainy</td>
<td>Low</td>
<td>No</td>
</tr>
</tbody>
</table>

- **Three key elements**
  - Class label ("label", denoted by \( y \))
  - Features ("attributes")
  - Feature values ("attribute values", denoted by \( x \))

  Feature values can be **binary, nominal or continuous**

- A **labeled dataset** is a collection of \((x, y)\) pairs
Example: To play or not to play tennis?

- Example dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Windy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play</td>
<td>Sunny</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>No play</td>
<td>Sunny</td>
<td>High</td>
<td>Yes</td>
</tr>
<tr>
<td>No play</td>
<td>Sunny</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Play</td>
<td>Overcast</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>No play</td>
<td>Rainy</td>
<td>Low</td>
<td>Yes</td>
</tr>
<tr>
<td>Play</td>
<td>Rainy</td>
<td>Low</td>
<td>No</td>
</tr>
</tbody>
</table>

- Task:

- Predict the **class** of this “test” sample
- Requires us to **generalize** from the training data
What is a good *representation* for images? Pixel values? Edges?
Example (chair detection)
Ingredients for classification

- **Idea:** Incorporate your knowledge of the problem into a learning system

- **Sources of knowledge:**
  1. Feature representation
     - Important for success of machine learning
     - Can be “problem specific”
     - Designing a good representation will take you half way
  2. Training data: labeled examples
     - High quality labeled data can be hard to find
     - Often have to make do with the available data
  3. Model
     - No single learning algorithm outperforms all others on every task ("no free lunch")
     - Different learning algorithms come with different inductive biases
Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it’s probably a duck
Nearest-Neighbor Classifiers

● Requires three things
  – The set of stored training samples and their labels
  – Distance Metric to compute distance between samples
  – The value of K, the number of nearest neighbors to retrieve

● To classify a query sample:
  – Compute distance to training samples
  – Identify K nearest neighbors
  – Use class labels of the K nearest neighbors to determine the class label of the query sample (e.g., by taking majority vote)
Definition of Nearest Neighbor

K-nearest neighbors of a sample x are data points that have the k smallest distance to x

(a) 1-nearest neighbor  (b) 2-nearest neighbor  (c) 3-nearest neighbor
K nearest neighbor classifier

Data samples are assumed to lie in an $n$-dimensional space – e.g., the Euclidean space.

An instance $X$ is described by a feature vector

$$X_p=[x_{1p} \cdots x_{Np}]$$

Where $x_{ip}$ denotes the value of the $i$th feature in $X_p$

$$d(X_p, X_r) = \left( \sum_{i=1}^{N} (x_{ip} - x_{ir})^2 \right)$$

Defines the Euclidean distance between two points in the Euclidean space.
Standardization

Standardization can be important when the variables are not all measured on the same scale

• 0-1 scaling
  
  4, 3, 1 2
  
  e.g. 3 $\rightarrow$ $(3-\text{min})/(\text{max}-\text{min})=(3-1)/(4-1)=2/3$

• Z-score scaling: subtract out the mean, divide by std. deviation
K nearest neighbor Classifier

Learning Phase

For each training example \((X_i, f(X_i))\), store the example in memory

Classification phase

Given a query instance \(X_q\), identify the \(k\) nearest neighbors \(X_1 \ldots X_k\) of \(X_q\)

Assign \(X_q\) the label of the majority class

\[
g(X_q) = \arg\max_\omega \sum_{i=1}^{k} \delta(\omega, f(x_i)) \quad \text{where}
\]

\[
\delta(a, b) = 1 \text{ iff } a = b \text{ and } \delta(a, b) = 1.
\]
Distance weighted K nearest neighbor Classifier

Learning Phase
For each training example \((X_i, f(X_i))\), store the example in memory

Classification phase
Given a query instance \(X_q\), identify the \(k\) nearest neighbors of \(X_q\) - \(KNN(\ X_q) = \{X_1 \ldots X_k\}\)

And obtain a weighted vote, with each nearest neighbor getting a vote in favor of its class label that is weighted by the distance to the query

\[ w_i = \frac{1}{d(X_i, X_q)^2} \]
Distance Measures

- Distance
  - Depends on the data representation
  - Distance measure chosen

An Employee DB

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Age</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F</td>
<td>27</td>
<td>19,000</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
<td>51</td>
<td>64,000</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>52</td>
<td>100,000</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>33</td>
<td>55,000</td>
</tr>
<tr>
<td>5</td>
<td>M</td>
<td>45</td>
<td>45,000</td>
</tr>
</tbody>
</table>

Word Frequencies for Documents

<table>
<thead>
<tr>
<th></th>
<th>w1</th>
<th>w2</th>
<th>w3</th>
<th>w4</th>
<th>w5</th>
<th>w6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doc1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Doc2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Doc3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Doc4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Doc5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Representation has to be chosen with some care
Distance measure should be chosen to work with the representation
Distance measures

- \( d(p, q) \) between two points \( p \) and \( q \) is a proper distance measure if it satisfies:

1. Positive definiteness:
   \[
   d(p, q) \geq 0 \quad \text{for all } p \text{ and } q \text{ and } \\
   d(p, q) = 0 \text{ only if } p = q.
   \]

2. Symmetry: \( d(p, q) = d(q, p) \) for all \( p \) and \( q \).

3. Triangle Inequality:
   \[
   d(p, r) \leq d(p, q) + d(q, r) \quad \text{for all points } p, q, \text{ and } r.
   \]
Cosine Distance

• If \( d_1 \) and \( d_2 \) are two document vectors, then
  \[
  1-\cos( d_1, d_2 ) = 1 - (d_1 \cdot d_2) / \| d_1 \| \| d_2 \| ,
  \]
  where \( \cdot \) indicates vector dot product and \( \| d \| \) is the length of vector \( d \).

• Example:
  \[
  d_1 = 3 2 0 5 0 0 0 2 0 0 \\
  d_2 = 1 0 0 0 0 0 1 0 2 \\
  d_1 \cdot d_2 = 3x1 + 2x0 + 0x0 + 5x0 + 0x0 + 0x0 + 0x0 + 2x1 + 0x0 + 0x2 = 5 \\
  \| d_1 \| = (3x3+2x2+0x0+5x5+0x0+0x0+0x0+2x2+0x0+0x0+0x0)^{0.5} = (42)^{0.5} = 6.481 \\
  \| d_2 \| = (6)^{0.5} = 2.245 \\
  \cos( d_1, d_2 ) = .3150
  \]
Distance Measures

Distances in vector spaces

- Euclidean distance $\sqrt{\sum_{j=1}^{d} (p_j - q_j)^2}$

- Minkowski distance
  - a generalization of Euclidean distance
    
    $\sqrt[n]{\sum_{j=1}^{d} |p_j - q_j|^n}$

Distance measures in Boolean spaces

- $n=1$ Manhattan distance
- $n=2$ Euclidean distance
Distance measures for nominal attributes

- Nominal attributes can take 2 or more values, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Simple matching – distance between two objects is simply the number of mismatched attributes divided by the total number of attributes
- One hot encoding – Encode each $M$-valued nominal attribute an $M$-bit vector
  Red: $1 0 0 0$, Yellow: $0 1 0 0$; Blue: $0 0 1 0$ ...
- Use distance measures designed for vectors ...
Decision Boundary induced by the 1 nearest neighbor classifier

Voronoi Diagram
Nearest Neighbor Classification...

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes
Nearest neighbor classifiers

- Nearest neighbor classifiers are conceptually simple
- Learn by simply memorizing training examples
- The computational effort of learning is low
- The storage requirements of learning is high
  - need to memorize the examples in the training set
- Cost of classifying new instances can be high
  - Use efficient data structures and algorithms for nearest neighbor search, k-d trees, e.g., locality sensitive hashing
- A distance measure needs to be defined over the input space
- Performance degrades when there are many irrelevant attributes:
  - Perform feature selection or dimensionality reduction
Regression or Function approximation

- **Regression** is like **classification** except the **labels** are **real valued**

- **Example applications:**
  - Predicting
    - Stock value
    - Income
    - Power consumption
K nearest neighbor Function Approximator

Learning Phase
For each training example \((X_i, f(X_i))\), store the example in memory

Approximation phase
Given a query instance \(X_q\), identify the \(k\) nearest neighbors \(X_1 \ldots X_k\) of \(X_q\)

\[
g(X_q) \leftarrow \frac{\sum_{i=1}^{K} f(X_i)}{K}
\]
Regression

- For classification the output is nominal
- In regression the output is continuous
  - Function Approximation
- Linear regression is perhaps the simplest approach
  - Fit data with the best hyper-plane which "goes through" the points
Regression

• For classification the output(s) is nominal
• In regression the output is continuous
  – Function Approximation
• Many models could be used – Simplest is linear regression
  – Fit data with the best hyper-plane which "goes through" the points

\[ y \text{ dependent variable (output)} \]
\[ x - \text{independent variable (input)} \]
Simple Linear Regression

• For now, assume just one (input) independent variable \( x \), and one (output) dependent variable \( y \)
  - Multiple linear regression assumes an input vector \( x \)
  - Multivariate linear regression assumes an output vector \( y \)
• We will "fit" the points with a linear hyper-plane (line in the simplest case)
• Which line should we use?
  - Choose an objective function
  - For simple linear regression we choose sum squared error (SSE)
    • \( \sum (d_i - y_i)^2 = \sum (e_i)^2 \)
    - Thus, find the line which minimizes the sum of the squared residues (e.g. least squares)
Digression – Minimizing / Maximizing Functions

Consider $f(x)$, a function of a scalar variable $x$ with domain $D_x$. $f(x)$ is convex over some sub-domain $D \subseteq D_x$ if $\forall X_1, X_2 \in D$, the chord joining the points $f(X_1)$ and $f(X_2)$ lies above the graph of $f(x)$.

$f(x)$ has a local minimum at $x = X_a$ if $\exists$ neighborhood $U \subseteq D_x$ around $X_a$ such that $\forall x \in U, f(x) > f(X_a)$.

We say that $\lim_{x \to a} f(x) = A$ if, for any $\epsilon > 0$, $\exists \delta > 0$ such that $|f(x) - A| < \epsilon$ for all $x$ such that $|x - a| < \delta$. 
Minimizing/Maximizing Functions

We say that $f(x)$ is continuous at $x = a$ if
$$\lim_{\varepsilon \to 0} \left\{ \lim_{x \to a+\varepsilon} f(x) \right\} = \lim_{\varepsilon \to 0} \left\{ \lim_{x \to a-\varepsilon} f(x) \right\}$$

The derivative of the function $f(x)$ is defined as
$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\frac{df}{dx} \bigg|_{x=X_0} = 0 \text{ if } X_0 \text{ is a local maximum or a local minimum}$$
Minimizing/Maximizing Functions

\[ \frac{d(u + v)}{dx} = \frac{du}{dx} + \frac{dv}{dx} \]

\[ \frac{d(uv)}{dx} = u \frac{dv}{dx} + v \frac{du}{dx} \]

\[ \frac{d\left(\frac{u}{v}\right)}{dx} = \frac{v \left( \frac{du}{dx} \right) - u \left( \frac{dv}{dx} \right)}{v^2} \]
Examples

\[ f(x) = x^2 + 3x \]

\[ \frac{d(u + v)}{dx} = \frac{du}{dx} + \frac{dv}{dx} \]

\[ \frac{df}{dx} = \]
Examples

\[ f(x) = x(x + 3) \]

\[
\frac{d(uv)}{dx} = u \frac{dv}{dx} + v \frac{du}{dx}
\]

\[
\frac{df}{dx} =
\]
Examples

\[ f(x) = \frac{x(x + 3)}{x^2} \]

\[ \frac{d}{dx} \left( \frac{u}{v} \right) = \frac{v \left( \frac{du}{dx} \right) - u \left( \frac{dv}{dx} \right)}{v^2} \]

\[ \frac{df}{dx} = \]
Partial derivatives and chain rule

Let \( f(X) = f(x_0, x_1, x_2, \ldots, x_n) \)

\[
\frac{\partial f}{\partial x_i}
\]

is obtained by treating all \( x_i \) \( i \neq j \) as constant.

Chain rule

Let \( z = \varphi(u_1, \ldots, u_m) \)

Let \( u_i = f_i(x_0, x_1, \ldots, x_n) \)

Then \( \forall k \)

\[
\frac{\partial z}{\partial x_k} = \sum_{i=1}^{m} \left( \frac{\partial z}{\partial u_i} \right) \left( \frac{\partial u_i}{\partial x_k} \right)
\]
Example

- \( z = f(u, v) = u^2 + 2v \)
- \( u = f_1(x, y) = 2x + y \)
- \( v = f_2(x, y) = x^2 + y \)
- \( \frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial z}{\partial v} \frac{\partial v}{\partial x} \)
Taylor Series Approximation of Functions

Taylor series approximation of \( f(x) \)
If \( f(x) \) is differentiable i.e., its derivatives
\[
\frac{df}{dx}, \quad \frac{d^2 f}{dx^2} = \frac{d}{dx} \left( \frac{df}{dx} \right), \quad \ldots \quad \frac{d^n f}{dx^n}
\]
exist at \( x = X_0 \) and \( f(x) \) is continuous in the neighborhood of \( x = X_0 \), then

\[ f(x) = f(X_0) + \left( \frac{df}{dx} \right)_{x=X_0} (x-X_0) + \cdots + \frac{1}{n!} \left( \frac{d^n f}{dx^n} \right)_{x=X_0} (x-X_0)^n \]

\[ f(x) \approx f(X_0) + \left( \frac{df}{dx} \right)_{x=X_0} (x-X_0) \]
Example
Taylor Series Approximation of Multivariate Functions

Let \( f(\mathbf{X}) = f(x_0, x_1, x_2, \ldots, x_n) \) be differentiable and continuous at \( \mathbf{X}_0 = (x_{00}, x_{10}, x_{20}, \ldots, x_{n0}) \).

Then

\[
f(\mathbf{X}) \approx f(\mathbf{X}_0) + \sum_{i=0}^{n} \left. \left( \frac{\partial f}{\partial x} \right) \right|_{\mathbf{X}=\mathbf{X}_0} (x_i - x_{i0})
\]
Minimizing / Maximizing Multivariate Functions

To find $X^*$ that minimizes $f(X)$, we change current guess $X^C$ in the direction of the negative gradient of $f(X)$ evaluated at $X^C$.

$$X^C \leftarrow X^C - \eta \begin{pmatrix} \frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \end{pmatrix}\bigg|_{X=X^C}$$

(why?)

for small (ideally infinitesimally small)