Machine Learning Algorithms for Bioinformatics

Hyeoncheol Kim

Contents

• Overview – Classification –
• Information-based Methods
• Decision Tree
• Neural Networks
  – Perceptron
  – MLP (Multi-Layered Perceptron)
  – Example: HIV Domain
• Association Rule Mining

Machine Learning Algorithms

• Data-driven approach
• Non-parametric approach
• Human flavor to knowledge representation (structure)
• Can deal with missing and noisy data, and uncertainty

Classification

• Decision Tree
• Neural Networks (MLP)
• Support Vector Machine
• Rule Induction

Information-based Method

BITS

• We observed symbols of variable X, and probabilities of each symbol to occur are:

<table>
<thead>
<tr>
<th>X = A</th>
<th>X = B</th>
<th>X = C</th>
<th>X = D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>

• To encode a sequence BAACBADCDDADA... efficiently, we need 2 bits for each symbol
  – i.e., A=00, B=01, C=10, D=11

Classifiers

New Data
ccatagcgtca...

Classifier
Decision Tree
Bayes Classifier
Neural Network

Description of Domain

Known Data
aatgattc...
→ cancer A

If “g” @ p4, then Cancer A

Class of the new data
Cancer A

Available Data
Attributes, class

Domain
Attributes, class

Decision Tree
MLP
SVM
Rules

Class A
Class B
Fewer BITS

We observed the symbols of variable X, and the probabilities for each symbol are:

- \( P(X=A) = \frac{1}{8} \)
- \( P(X=B) = \frac{1}{8} \)
- \( P(X=C) = \frac{1}{4} \)
- \( P(X=D) = \frac{1}{2} \)

We can encode each symbol with just 1.75 bits (<2bits).

- i.e., A=0, B=10, C=110, D=111

\[
\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = 1.75
\]

Information and Uncertainty

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Probability</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>Yes</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>EE</td>
<td>No</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>CSE</td>
<td>Yes</td>
<td>0.25</td>
<td>0.5</td>
</tr>
</tbody>
</table>

On a Sequence

- Given a sequence of amino acids, Entropy at each position provides
  - Which position is more important than others
  - Which symbol at each position is more important than others
- Quiz) 20 different symbols for amino acid. What is the lowest entropy?

Assuming Uniform distribution, \( \log_2(20)=4.xxx \) bits

Specific Conditional Entropy

- \( H(Y|X=v) = \) The entropy of Y among only those instances in which X has value v
- Example)
  - \( H(Y|X=\text{math}) = 1 \)
  - \( H(Y|X=\text{EE}) = 0 \)
  - \( H(Y|X=\text{CSE}) = 0 \)

\[
H(X) = - \sum (p_j \cdot \log_2(p_j)) = \text{The entropy of } X
\]

Low Information

Lowest Uncertainty

High Information

Highest Uncertainty

General Case

- Suppose X can have one of m values... \( V_1, X_2,...V_m \)
- \( P(X=V_i)=p_i \)
- Smallest possible number of bits, on average, per symbol?
  - \( H(X) = - \sum (p_j \cdot \log_2(p_j)) \)
  - The entropy of X

Specific Conditional Entropy

- Given Input X, output Y?
- \( \text{e.g.} \)
  - \( P(Y)=0.5 \)
  - \( P(\text{Math} & \sim Y)=0.25 \)
  - \( P(\text{Math})=0.5 \)
  - \( P(Y | \text{EE}) = 0 \)
- Entropy
  - \( H(X)=1.5 \)
  - \( H(Y)=1 \)

\[
H(Y|X) = \frac{1}{2} \log_2(\frac{1}{2}) + \frac{1}{4} \log_2(\frac{1}{4}) + \frac{1}{4} \log_2(\frac{1}{4}) + \frac{1}{2} \log_2(\frac{1}{2}) = 1.5
\]

\[
H(Y|X) = \frac{1}{2} \log_2(\frac{1}{2}) + \frac{1}{4} \log_2(\frac{1}{4}) + \frac{1}{4} \log_2(\frac{1}{4}) + \frac{1}{2} \log_2(\frac{1}{2}) = 1.5
\]
**Conditional Entropy**

\[ H(Y|X) = \text{The average specific conditional entropy of } Y \]

\[ H(Y|X) = \sum_P P(X=v_j) \cdot H(Y|X=v_j) \]

\[ H(Y|X) = 0.5 \cdot 1 + 0.25 \cdot 0 + 0.25 \cdot 0 = 0.5 \]

---

**Information Gain**

\[ IG(Y|X) : \text{Given information } X, \text{ How much entropy of } Y \text{ can be decreased?} \]

\[ IG(Y|X) = H(Y) - H(Y|X) \]

**Example**

- \( H(Y) = 1 \)
- \( H(Y|X) = 0.5 \)
- \( IG(Y|X) = 1 - 0.5 = 0.5 \)

---

**Sequence Logos**

- graphical representation of an amino acid or nucleic acid multiple sequence alignment
- Each logo consists of stacks of symbols, one stack for each position in the sequence.
- define the sequence conservation at a particular position in the alignment
- as the difference between the maximum possible entropy and the entropy of the observed symbol distribution:

**Idea of Sequence Logos**

\[ H_{max} - H(S) = \sum_{n=1}^{N} \log_2 N \]

- \( P_n \) is the observed frequency of symbol \( n \) at a particular sequence position
- \( N \) is the number of distinct symbols for the given sequence type, either four for DNA/RNA or 20 for protein.
- maximum sequence conservation per site is \( \log_2 4 = 2 \) bits for DNA/RNA and \( \log_2 20 \approx 4.32 \) bits for proteins.

**Create**

- The logo generation form (http://weblogo.berkeley.edu/logo.cgi) can process RNA, DNA, or protein multiple sequence alignments provided in either FASTA or CLUSTAL formats.

---

**Decision Tree (ID3)**

<table>
<thead>
<tr>
<th>Fruit Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
</tr>
<tr>
<td>Small</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>Big</td>
</tr>
<tr>
<td>Medium</td>
</tr>
<tr>
<td>Medium</td>
</tr>
</tbody>
</table>

- Which features of a fruit are more important to decide its class?
- What value of the features?
**Decision Tree (ID3)**

Fruit dataset

<table>
<thead>
<tr>
<th>Size</th>
<th>Color</th>
<th>Surface</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Yellow</td>
<td>Smooth</td>
<td>A</td>
</tr>
<tr>
<td>Medium</td>
<td>Red</td>
<td>Smooth</td>
<td>A</td>
</tr>
<tr>
<td>Big</td>
<td>Red</td>
<td>Rough</td>
<td>A</td>
</tr>
<tr>
<td>Medium</td>
<td>Yellow</td>
<td>Smooth</td>
<td>B</td>
</tr>
<tr>
<td>Medium</td>
<td>Yellow</td>
<td>Smooth</td>
<td>B</td>
</tr>
</tbody>
</table>

- $H(Class|Xi)$?
  - $H(Class|Size)=\frac{1}{6}(-1/1 \times \log(1/1)) + \frac{2}{3}(-2/2 \times \log(2/2)) = 0.452$
  - $H(Class|Color) = 0.318$
  - $H(Class|Surface) = 0.56$

- $H(Class|Size) = \sum (-p_i \times \log(p_i))$

- Re-calculate the $H$ with the rest
  - $H of size = \frac{1}{3}(-1/1 \times \log(1/1)) + \frac{2}{3}(-2/2 \times \log(2/2)) = 0$
  - $H of surface = 0.636$

**Decision Tree (ID3)**

Fruit dataset

<table>
<thead>
<tr>
<th>Size</th>
<th>Color</th>
<th>Surface</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Yellow</td>
<td>Smooth</td>
<td>A</td>
</tr>
<tr>
<td>Medium</td>
<td>Red</td>
<td>Smooth</td>
<td>A</td>
</tr>
<tr>
<td>Big</td>
<td>Red</td>
<td>Rough</td>
<td>A</td>
</tr>
<tr>
<td>Medium</td>
<td>Yellow</td>
<td>Smooth</td>
<td>B</td>
</tr>
<tr>
<td>Medium</td>
<td>Yellow</td>
<td>Smooth</td>
<td>B</td>
</tr>
</tbody>
</table>

- If Color = red, then A
- If Color = yellow & Size = small, then A
- If Color = yellow & Size = medium, then B

**Neural Networks**

- MLP (Multi-layered Perceptrons)
  - Supervised network
  - Feed-forward network
  - Backpropagation network
  - Classification
- SOM (Self-Organizing Feature Map)
  - Unsupervised
  - Clustering

**Perceptron and Neuron**

- Simple information processor
- Connection weight values are updated adaptively.
- Updating the weight values to decrease error (i.e., difference between actual output and predicted output)

**Learning: Delta Rule**

- $\delta = \frac{\text{Actual output} - \text{Expected output}}{\text{Sum of weights}}$
- $w_i' = w_i + \delta \times x_i$
Example: Perceptron that learns OR

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

How to learn the weight values of incoming connections?

Beyond Perceptron

• Linear-Separability is enough?
• Can Perceptron learn XOR?

MLP (Multi-Layered Perceptron)

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Multi-Layered Perceptron

Neural Network (MLP, Backpropagation)

• One of the most powerful classification/prediction tools!
• Provides arbitrary non-linear discriminant
• Deal with noisy data very well
• Weakness:
  – Proper Architecture (hidden units)
  – Interpretation of what the network has learned.
**MLP (Multi-Layered Perceptron)**

Input vector: $x_1, x_2, x_3, x_4, x_5$

Output vector: $y_1, y_2$

**Domain Data $(X, Y)$**

$x_1, x_2, y_1$ belongs to class $y_1$!

**Interpretation of what NN has learned...**

If $x_1$ & $x_2$, then $y_1$
If $x_3$, then $y_2$
If $x_5$ & $\neg x_3$, then $y_1$

**Human understandable**

**Terms**

- MLP (Multi-Layered Perceptron) ➔ Architecture & Elements
- Feed-forward Neural Network ➔ Process
- Back-propagation Network ➔ Algorithm

**So many different Neural Networks**

**NN Applications**

- **[App1]** The coding region recognition and gene identification problem
  - Tackled by two approaches
    - Search by content
    - Search by signal
  - NN can use both
    - GRAIL system
      - Classifying DNA sequences into two groups, exons and not exons; i.e., the network is trained to identify coding regions (1997)
      - $13 \times 3 \times 1$ MLP
      - The 13 features include sequence length, exon GC composition, Markov scores, and other complex scales describing the candidate sequence.
- **[App2]** Identification of signals, sites, etc.
  - NN
    - Allows the incorporation of both positive and negative examples as well as the detection of higher-order and long-range correlation,
    - Are not based on the assumption of positional independence.

**[App3]** Sequence classification and feature detection

**Learning Domain Data Will be discussed!!!**
Sequence classification and feature detection

- Detecting significant sequence features and understanding biological rules that govern gene structure and gene regulation

Example:

HIV and HCV Protease cleavage site prediction

- HIV dataset
  - 160-8-1 MLP used
- HCV dataset
  - 250-20-1 MLP used.

Applications from Recent Literatures

- Protein secondary structure prediction
- Signal peptide prediction
- Gene finding and gene structure prediction
- T-cell epitope prediction
- RNA secondary structure prediction
- Toxicity prediction
- Disease diagnosis and outcome prediction
- Gene expression analysis
- Protein translation initiation site recognition

Association Rule Mining

- Association rules from protein-protein interaction data
  - "the protein having a feature A interacts with the protein having the feature B"
  - "This domain interacts with that domain"
- Expression associations from gene expression profiling data

Association Rule Overview

- Problem Statement
  - Transactional Data
  - Support, Confidence
- Application: Market Basket Analysis
- Complexity
- Interestingness of the rule
## Frequent Itemsets / Association Rules

### Complexity
- Search complexity for itemsets of size 2?
- How about itemsets of size 3?
- How about size 4?

### A-Priori Algorithm

#### A-priori property
- If a set is frequent itemset, all its subsets are also frequent itemsets. 

=> Reducing Search Space

### Interestingness
- Is the items in the frequent itemset are really associated? Or just happened by accident?
- Isn’t the rule meaningless?
- How do we evaluate their interestingness?

### Problems
- High frequent single item – very likely to become a rule.
- Is that real association between items?
- Support and confidence are not enough.
- Interestingness!
  - Correlation/Lift
  - Conviction
  - Etc.
Measures for rule (A->B)

- Support: \( P(A, B) \)
- Confidence: \( P(B|A) \)

More measures for A->B

- Correlation (Lift)
  - \( \frac{P(A, B)}{P(A)*P(B)} \)
  - If A and B are independent, \( P(A,B) = P(A)*P(B) \)
  - Lift=1 tells “they are independent”
  - Lift>1 tells “positive association”
  - Lift<1 tells “negative association”

More measures for A->B

- Conviction
  - \( \frac{P(A)*P(\neg B)}{P(A, \neg B)} \)
  - A->B = (\neg A or B)

Association rules from protein-protein interaction data

- Oyama’s experiment (2002)
  - 4307 pairs of yeast interaction proteins from 4 source databases.
  - 5241 features categorized into 7 types
  - Discovered 6367 rules
  - Ex)
    - “a SH3 domain binds to a proline-rich region”

associations from gene expression profiling data

- To understand gene functions, biological networks, and cellular states.
- Creighton and Hanash’s experiment (2003)
  - Revealing biologically relevant associations between different genes or between environmental effects and gene expression.
  - 6316 transcripts as features, corresponding 300 diverse mutations and chemical treatment in yeast.
  - More useful than clustering methods to uncover gene networks.