COSC 522 – Machine Learning

Lecture 12 – Decision Tree and Random Forest

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Course Website: http://web.eecs.utk.edu/~hqi/cosc522/
Racap - Decision Rules

- Supervised learning
  - Baysian based - Maximum Posterior Probability (MPP): For a given \( x \), if \( P(w_1|x) > P(w_2|x) \), then \( x \) belongs to class 1, otherwise 2.
    - Parametric Learning
      - Case 1: Minimum Euclidean Distance (Linear Machine), \( \Sigma_i = \sigma^2 I \)
      - Case 2: Minimum Mahalanobis Distance (Linear Machine), \( \Sigma_i = \Sigma \)
      - Case 3: Quadratic classifier, \( \Sigma_i = \) arbitrary
      - Estimate Gaussian parameters using MLE
    - Nonparametric Learning
      - K-Nearest Neighbor
  - Neural network
    - Perceptron
    - BPNN
  - Kernel-based approaches
    - Support Vector Machine
  - Decision tree
    - Least-square based

- Unsupervised learning
  - Kmeans
  - Winner-takes-all

- Supporting preprocessing techniques
  - Normalization
  - Dimensionality Reduction (FLD, PCA)

- Performance Evaluation (metrics, confusion matrices, ROC, cross validation)
COSC 522 - Machine Learning (Fall 2020) Syllabus

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>Content</th>
<th>Tests</th>
<th>Assignment</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8/20</td>
<td>Introduction</td>
<td></td>
<td>HW0</td>
<td>8/25</td>
</tr>
<tr>
<td>2</td>
<td>8/25</td>
<td>Baysian Decision Theory - MPP</td>
<td>HW1</td>
<td>Proj1 - Supervised Learning</td>
<td>9/1</td>
</tr>
<tr>
<td>3</td>
<td>8/27</td>
<td>Discriminant Function - MD</td>
<td></td>
<td></td>
<td>9/10</td>
</tr>
<tr>
<td>4</td>
<td>9/1</td>
<td>Parametric Learning - MLE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9/3</td>
<td>Non-parametric Learning - kNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9/8</td>
<td>Unsupervised Learning - kmeans</td>
<td>HW2</td>
<td>Proj2 - Unsupervised Learning and DR</td>
<td>9/24</td>
</tr>
<tr>
<td>7</td>
<td>9/10</td>
<td>Dimensionality Reduction - FLD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9/15</td>
<td>Dimensionality Reduction - PCA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9/17</td>
<td>Linear Regression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9/22</td>
<td>Performance Evaluation</td>
<td>HW3</td>
<td></td>
<td>9/29</td>
</tr>
<tr>
<td>11</td>
<td>9/24</td>
<td>Fusion</td>
<td>Proj3 - Regression</td>
<td></td>
<td>10/8</td>
</tr>
<tr>
<td>12</td>
<td>9/29</td>
<td>Midterm Exam</td>
<td>Final Project - Milestone 1: Forming Team</td>
<td>9/29</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10/1</td>
<td>Gradient Descent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10/6</td>
<td>Neural Network - Perceptron</td>
<td>HW4</td>
<td></td>
<td>10/13</td>
</tr>
<tr>
<td>15</td>
<td>10/8</td>
<td>Neural Network - BPNN</td>
<td>Proj4 - BPNN</td>
<td></td>
<td>10/22</td>
</tr>
<tr>
<td>16</td>
<td>10/13</td>
<td>Neural Network - Practices</td>
<td>Final Project - Milestone 2: Choosing Topic</td>
<td>10/13</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10/15</td>
<td>Kernel Methods - SVM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>10/20</td>
<td>Kernel Methods - SVM</td>
<td>HW5</td>
<td></td>
<td>10/27</td>
</tr>
<tr>
<td>19</td>
<td>10/22</td>
<td>Kernel Methods - SVM</td>
<td>Proj5 - SVM &amp; DT</td>
<td></td>
<td>11/5</td>
</tr>
<tr>
<td>20</td>
<td>10/27</td>
<td>Decision Tree</td>
<td>Final Project - Milestone 3: Literature Survey</td>
<td>10/27</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>10/29</td>
<td>Random Forest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>11/3</td>
<td></td>
<td>HW6</td>
<td></td>
<td>11/10</td>
</tr>
<tr>
<td>23</td>
<td>11/5</td>
<td>From PCA to t-SNE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>11/10</td>
<td>From Gaussian to Mixture and EM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>11/12</td>
<td>From Supervised/Unsupervised to RL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>11/17</td>
<td>From Classification/Regression to Generation</td>
<td>Final Project - Milestone 4: Prototype</td>
<td>11/17</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>11/19</td>
<td>From NN to CNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>11/24</td>
<td>Final Exam</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/3</td>
<td>8:00-10:15</td>
<td>Final Presentation Final Project - Report</td>
<td>12/4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions

• What does generalization and capacity mean?
• What is VC dimension?
• What is the principled method?
• What is the VC dimension for perceptron?
• What are support vectors?
• What is the cost function for SVM?
• What is the optimization method used?
• How to handle non-separable cases using SVM?
• What is kernel trick?
Nominal Data

- Descriptions that are discrete and without any natural notion of similarity or even ordering
Some Terminologies

Decision tree
- Root
- Link (branch) - directional
- Leaf
- Descendent node
FIGURE 8.1. Classification in a basic decision tree proceeds from top to bottom. The questions at each node concern a particular property of the pattern, and the downward links correspond to the possible answers. Successive nodes are visited until a terminal or leaf node is reached, where the category label is determined. Note that the same question, Size?, appears in different places in the tree and that different questions can have different numbers of branches. Moreover, different leaf nodes, shown in pink, can be labeled.
CART

- Classification and regression trees
- A generic tree growing methodology

Issues studied

- How many splits from a node?
- Which property to test at each node?
- When to declare a leaf?
- How to prune a large, redundant tree?
- If the leaf is impure, how to classify?
- How to handle missing data?
Number of Splits

- Binary tree
- Expressive power and comparative simplicity in training
Node Impurity – Occam’s Razor

• The fundamental principle underlying tree creation is that of simplicity: we prefer simple, compact tree with few nodes
• Occam's (or Ockham's) razor is a principle attributed to the 14th century logician and Franciscan friar; William of Occam. Ockham was the village in the English county of Surrey where he was born.
• The principle states that “Entities should not be multiplied unnecessarily.”
• "when you have two competing theories which make exactly the same predictions, the one that is simpler is the better.“
• Stephen Hawking explains in A Brief History of Time: “We could still imagine that there is a set of laws that determines events completely for some supernatural being, who could observe the present state of the universe without disturbing it. However, such models of the universe are not of much interest to us mortals. It seems better to employ the principle known as Occam's razor and cut out all the features of the theory which cannot be observed.”
• Everything should be made as simple as possible, but not simpler
Property Query and Impurity Measurement

We seek a property query $T$ at each node $N$ that makes the data reach the immediate descendant nodes as pure as possible.

We want $i(N)$ to be 0 if all the patterns reach the node bear the same category label.

Entropy impurity (information impurity)

$$i(N) = - \sum_{j} P(\omega_j) \log_2 P(\omega_j)$$

$P(\omega_j)$ is the fraction of patterns at node $N$ that are in category $\omega_j$. 
Other Impurity Measurements

- **Variance impurity (2-category case)**
  \[
i(N) = P(ω_1)P(ω_2)
\]

- **Gini impurity**
  \[
i(N) = \sum_{i \neq j} P(ω_i)P(ω_j) = 1 - \sum_j P^2(ω_j)
\]

- **Misclassification impurity**
  \[
i(N) = 1 - \max_j P(ω_j)
\]

*FIGURE 8.4.* For the two-category case, the impurity functions peak at equal class frequencies and the variance and the Gini impurity functions are identical. The entropy, variance, Gini, and misclassification impurities (given by Eqs. 1–4, respectively) have been adjusted in scale and offset to facilitate comparison here; such scale and offset do not directly affect learning or classification. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification.* Copyright © 2001 by John Wiley & Sons, Inc.
Choose the Property Test?

Choose the query that decreases the impurity as much as possible

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R)$$

- $N_L$, $N_R$: left and right descendant nodes
- $i(N_L)$, $i(N_R)$: impurities
- $P_L$: fraction of patterns at node $N$ that will go to $N_L$

Solve for extrema (local extrema)
Example

Node N:
- 90 patterns in $\omega_1$
- 10 patterns in $\omega_2$

Split candidate:
- 70 $\omega_1$ patterns & 0 $\omega_2$ patterns to the right
- 20 $\omega_1$ patterns & 10 $\omega_2$ patterns to the left
When to Stop Splitting?

Two extreme scenarios
- Overfitting (each leaf is one sample)
- High error rate

Approaches
- Validation and cross-validation
  - 90% of the data set as training data
  - 10% of the data set as validation data
- Use threshold
  - Unbalanced tree
  - Hard to choose threshold
- Minimum description length (MDL)
  - $i(N)$ measures the uncertainty of the training data
  - Size of the tree measures the complexity of the classifier itself

$$MDL = \alpha \cdot \text{size} + \sum_{\text{leaf nodes}} i(N)$$
When to Stop Splitting? (cont’)

- Use stopping criterion based on the statistical significance of the reduction of impurity
  - Use chi-square statistic
  - Whether the candidate split differs significantly from a random split

\[
\chi^2 = \sum_{i=1}^{2} \frac{(n_{iL} - n_{ie})^2}{n_{ie}}
\]
Pruning

- Another way to stop splitting
- Horizon effect
  - Lack of sufficient look ahead
- Let the tree fully grow, i.e. beyond any putative horizon, then all pairs of neighboring leaf nodes are considered for elimination
Instability
Other Methods

- Quinlan’s ID3
- C4.5 (successor and refinement of ID3)
  http://www.rulequest.com/Personal/
Random Forest

• Potential issue with decision trees
• Prof. Leo Breiman
• Ensemble learning methods
  – Bagging (Bootstrap aggregating): Proposed by Breiman in 1994 to improve the classification by combining classifications of randomly generated training sets
  – Random forest: bagging + random selection of features at each node to determine a split
Reference

