Machine Learning

Anna Salzberg
Machine Learning

• Subfield of artificial intelligence concerned with the design and development of algorithms that allow computers to improve their performance over time based on data.

• A major focus of machine learning is to automatically produce models (such as rules and patterns) from data.
Human Interaction

• Some machine learning algorithms adopt a collaborative approach between human and machine (e.g. **Bayesian Networks**)

• Others attempt to eliminate the need for human intuition in data analysis (e.g. **Random Forest**)

• But human intuition cannot be entirely eliminated, since the system’s designer must specify how the data is to be represented.
Algorithm Types

• **Supervised Learning**: predicts the value of a function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output)
  – Neural Network (Multi-layer Perceptron)
  – Support Vector Machines (SVM)
  – Decision Tree
  – Random Forest
  – Gaussian Process

• **Unsupervised Learning**: models a set of inputs where labeled examples are not available
  – Clustering
  – Self-Organizing Map (SOM)
Algorithm Types - continued

• **Semi-supervised Learning**: combines both labeled and unlabeled examples to generate an appropriate function or classifier
  – Bayesian Networks: parameter & structure learning
  – Expectation-maximization (EM) algorithm

• **Reinforcement Learning**: concerned with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward
  – Simulated annealing
  – Genetic algorithms

• **Learning to Learn**: algorithm learns its own inductive bias based on previous experience
Hierarchical Clustering

• Builds a hierarchy of clustering
  – "bottom up": each observation starts in its own cluster, and pairs of clusters are merged as it moves up the hierarchy.
  – "top down": all observations start in one cluster, and splits are performed recursively as it moves down the hierarchy.

• In general, the merges and splits are determined in a greedy manner.
Hierarchical Clustering Parameters

- Distance measure between pairs of observations. Examples:
  - Euclidian:  \[ \sqrt{\sum_i(a_i - b_i)^2} \]
  - Cosine similarity:  \[ \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \]

- Linkage criterion specifies dissimilarity of sets as a function of pairwise distances of observations in the sets. Examples:
  - Maximum (complete linkage):  \[ \max \{ d(a, b) : a \in A, \ b \in B \} \]
  - Average:  \[ \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b) \]
Welcome to GenePattern

Analyzing genomic data in GenePattern

what do you want to do?

- Click a protocol to run an analysis. GenePattern guides you step by step.
- Click Quick Start for instructions on how to run any module in GenePattern.

Protocols for running common analyses in GenePattern:

- Run an Analysis in GenePattern
  Learn how to run an analysis in GenePattern by preprocessing gene expression data and visualizing the resulting data as a heat map.
- Differential Expression Analysis
  Find genes that are significantly differentially expressed between classes of samples.
- Clustering
  Group genes and/or samples by similar expression profiles.
- Prediction
  Create a model, also referred to as a classifier or class predictor, that correctly classifies unlabeled samples into known classes.
- SNP Copy Number and Loss of Heterozygosity Estimation
  Compute SNP copy number (CN) and loss of heterozygosity (LOH) based on Affymetrix SNP chip data for paired target/normal samples.

See also:

- Tutorial [HTML | PDF] Hands-on introduction to GenePattern.
- GenePattern User Guide Full description of this web application.
- Modules List of all installed modules with links to their documentation.

http://genepattern.broadinstitute.org/gp/pages/index.jsf
GenePattern Hierarchical Clustering

Input file: all_aml_test.gct
Input files to HierarchicalClusteringViewer are the output files from the HierarchicalClustering in the previous step.
GenePattern Clustering

Column Clustering: samples based on all genes

Row Clustering: genes based on sample expression profiles
GenePattern Biclustering
MATLAB Hierarchical Clustering

Clustering of Data Samples based on Gene Expression

Molecular Biomarkers

Alpha-2 Macroglobulin  Tissue Factor  MIF-1-alpha  Thrombopoietin  EGF  IL-2

Training and Test Samples

Z-score

Biol Psychiatry (2010)
Class Prediction

- Dataset comprised of rows (samples) and columns (features such as gene expression values)
- Divide the data into 2 data sets: train and test
- Choose a class prediction method and build a classifier using the train dataset, including the known class labels (e.g. tumor or normal tissue)
- Use the test dataset to predict the classification labels and compare predictions with the known ones to determine accuracy of classifier.
CART Class Prediction

- Data set describing objects by vectors of *features* and a *class*

<table>
<thead>
<tr>
<th>Age</th>
<th>Sex</th>
<th>Chest Pain</th>
<th>Rest BP</th>
<th>Cholesterol</th>
<th>Blood Sugar</th>
<th>Max Heart Rate</th>
<th>Angina</th>
<th>Old Peak</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>2</td>
<td>154</td>
<td>271</td>
<td>0</td>
<td>0</td>
<td>162</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>3</td>
<td>140</td>
<td>280</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>3</td>
<td>160</td>
<td>199</td>
<td>0</td>
<td>0</td>
<td>179</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>1</td>
<td>2</td>
<td>115</td>
<td>153</td>
<td>0</td>
<td>0</td>
<td>132</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>3</td>
<td>130</td>
<td>156</td>
<td>1</td>
<td>2</td>
<td>142</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>2</td>
<td>156</td>
<td>245</td>
<td>0</td>
<td>2</td>
<td>143</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>4</td>
<td>132</td>
<td>184</td>
<td>0</td>
<td>2</td>
<td>105</td>
<td>1</td>
<td>2.1</td>
</tr>
</tbody>
</table>

- Find a function $F$: *features* $\rightarrow$ *class* to classify a new object

(Nir Friedman)
Class Prediction Tree

Age Sex Pain BP Chol BS ECG MH Ang OpPeak
Vector₁ = <49, 0, 2, 154, 271, 0, 0, 162, 0, 0> Presence
Vector₂ = <42, 1, 3, 140, 280, 0, 0, 150, 0, 0> Presence
Vector₃ = <39, 0, 3, 160, 199, 0, 0, 179, 0, 0> Presence
Vector₄ = <41, 1, 2, 115, 153, 0, 0, 132, 0, 0> Absence
Vector₅ = <56, 1, 3, 130, 156, 1, 2, 142, 1, 0.6> Absence
Vector₆ = <70, 1, 2, 156, 245, 0, 2, 143, 0, 0> Presence
Vector₇ = <56, 1, 4, 132, 184, 0, 2, 105, 1, 2.1> Absence

Chol > 200

false
Vector₃
Vector₄
Vector₅
Vector₇
true
Vector₁
Vector₂
Vector₆

Gini Impurity Criterion

Sex < 1

false
Vector₄
Vector₅
Vector₇
true
Vector₃

Absence

Presence
SVM Class Prediction

- Separate the p-dimensional data points with a (p-1)-dimensional hyperplane.
- Choose hyperplane with the largest separation (margin).
- A new example is mapped to the same space and predicted to belong to a category depending on which side of the hyperplane it falls on.

H3 (green) doesn’t separate the 2 classes. H1 (blue) does, with a small margin and H2 (red) with the maximum margin.
GenePattern Result Files

<table>
<thead>
<tr>
<th>Samples</th>
<th>True Class</th>
<th>Predicted Class</th>
<th>Confidence</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL 19769</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>TA+ Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 406</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>TA+ (ML) Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 4466</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 1245</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>TA- Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 16125</td>
<td>ALL</td>
<td>ALL</td>
<td>1</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

**CART: 3 errors, 32 correct predictions**

<table>
<thead>
<tr>
<th>Samples</th>
<th>True Class</th>
<th>Predicted Class</th>
<th>Confidence</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL 19769 TA+ Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.9919481</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 406 TA+ (ML) Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.7321699</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 4466 Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.5648526</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 1245 TA- Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.5164803</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 16125 TA- Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.8851563</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL 23368 TA- Norel</td>
<td>ALL</td>
<td>ALL</td>
<td>0.9570557</td>
<td>TRUE</td>
</tr>
<tr>
<td>ALL R28 (ML) Relap</td>
<td>ALL</td>
<td>ALL</td>
<td>0.6765786</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

**SVM: 5 errors, 30 correct predictions**
Beware of Selection Bias!

- It is ok to tweak the parameters of the classifier, including selecting a subset of the features (e.g. genes), as long as these decisions are based on the training set alone.

- It is not ok to use the test set to evaluate the effects of such tweaking, because otherwise the test set data is indirectly being incorporated into the building of the classifier.
Random Forest (RF) Class Prediction

- **Bootstrapping**: Train an ensemble of decision trees (usually 500), each using a sampling of the data set.
- **Tree Construction**: Only consider a subset of possible descriptors when choosing each split.
- **Prediction**: Each of the 500 trees gets a single vote. The most popular vote is the predicted class.

**Impact**: the predictive power of each tree is weakened but the predictive power of the ensemble increases.
Random Forest Sample Prediction

Age Sex Pain BP Chol BS ECG MH Ang OPeak
Vector_{New} = <59, 1, 2, 158, 273, 0, 0, 164, 1, 0.5>

Each of the 500 trees gets a single vote.
RF Variable Importance

- Random Forest calculates an estimate of variable importance:
  1. In every tree grown in the forest, put down the test cases and count the number of votes cast for the correct class.
  2. Now randomly permute the values of variable m in the test cases and put these cases down the tree.
  3. Subtract the number of votes for the correct class in the variable-m-permuted test data from the number of votes for the correct class in the untouched test data.
  4. The average of this number over all trees in the forest is the raw importance score for variable m
Leukemia RF Results

Type of random forest: classification
Number of trees: 5000
No. of variables tried at each split: 5000

OOB estimate of error rate: 5.26%
Confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>FALSE</th>
<th>TRUE</th>
<th>class.error</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>27</td>
<td>0</td>
<td>0.00000000</td>
</tr>
<tr>
<td>TRUE</td>
<td>2</td>
<td>9</td>
<td>0.181818182</td>
</tr>
</tbody>
</table>

Test set error rate: 2.86%
Confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>FALSE</th>
<th>TRUE</th>
<th>class.error</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>21</td>
<td>0</td>
<td>0.00000000</td>
</tr>
<tr>
<td>TRUE</td>
<td>1</td>
<td>13</td>
<td>0.07142857</td>
</tr>
</tbody>
</table>

Random Forest beats CART and SVM for this example!
Leukemia RF Results (cont’d)

Model Prediction of ALL

- ALL: 21
- AML: 14

Model Prediction of AML

- ALL: 0
- AML: 13

Result: 97% Accuracy
Leukemia – RF Variable Importance

V4847 = Zyxin
V1882 = CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
V2020 = FAH Fumarylacetoacetate
V1926 = PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta

Zywin, CST3, and FAH were also indentified by Science paper of this dataset (using various methods including non-optimal RADVIZ) to be in the top 20 of the most important genes.

PTX3 was not listed in this Science paper but it was reported elsewhere as specifically binding membranes of human leukemia (Neuroscience, 2001)
EXTRA: Bayesian Networks
Example: Medical Diagnosis

[Image of a Bayesian network model with nodes labeled Smoking, Bronchitis, Lung Cancer, Fatigue, and Chest X-ray. The network shows relationships and probabilities between these conditions.]
Medical Diagnosis - Continuation

• Suppose we want to know the conditional probability of an individual having bronchitis given that he smokes, is fatigued, and has a positive chest x-ray

\[ P(b^+ | s^+, f^+, c^+) = \frac{P(b^+, s^+, f^+, c^+)}{P(s^+, f^+, c^+)} = \frac{\sum_{l} P(b^+, s^+, f^+, c^+, l)}{\sum_{b, l} P(b, s^+, f^+, c^+, l)} \]

• Problems:
  1. values in joint probability distributions ordinarily not readily available

  2. exponential number of terms based on number of variables
Markov Condition

• Suppose we have a joint probability distribution $P$ of the random variables in some set $V$ and a DAG $G=(V,E)$. We say $(G,P)$ satisfies the Markov condition if for each variable $X \in V$, $\{X\}$ is conditionally independent of the set of all its nondescendants given the set of all its parents.

• If we denote the sets of parents and nondescendants of $X$ by $PA_X$ and $ND_X$ respectively, then $(G,P)$ satisfies the Markov condition if for each variable $X \in V$, $I_p(\{X\}, ND_X \mid PA_X)$

• Let $P$ be a joint probability distribution of the random variables in some set $V$, and $G=(V,E)$ be a DAG. We call $(G,P)$ a Bayesian network if $(G,P)$ satisfies the Markov condition.
Bayesian Network

- Structure: **DAG**

- Meaning: a child is **conditionally independent** on its non-descendants, given the value of its parents

- Does not **impose causality**, but **suits** modeling of causal processes

(Nir Friedman)
Examples: Markov Condition

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Outcomes Mapped to this value</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>v1</td>
<td>all objects containing a “1”</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>all objects containing a “2”</td>
</tr>
<tr>
<td>S</td>
<td>s1</td>
<td>all square objects</td>
</tr>
<tr>
<td></td>
<td>s2</td>
<td>all round objects</td>
</tr>
<tr>
<td>C</td>
<td>c1</td>
<td>all blue objects</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>all violet objects</td>
</tr>
</tbody>
</table>

Then (G,P) satisfies the Markov condition for G in a), b), c), but **NOT** d) because \(\text{Ip}(\{V\}, \{S\})\) is not the case in d)
Examples Detailed

**a) Ip(V, S | C)**

\[
\begin{align*}
P(1 \mid \text{blue}) P(\text{sq} \mid \text{blue}) &= (1/3)(2/3) = P(1 \text{ sq} \mid \text{blue}) = 2/9 \\
P(2 \mid \text{blue}) P(\text{sq} \mid \text{blue}) &= (2/3)(2/3) = P(2 \text{ sq} \mid \text{blue}) = 4/9 \\
P(1 \mid \text{blue}) P(\text{cle} \mid \text{blue}) &= (1/3)(1/3) = P(1 \text{ cle} \mid \text{blue}) = 1/9 \\
P(2 \mid \text{blue}) P(\text{cle} \mid \text{blue}) &= (2/3)(1/3) = P(2 \text{ cle} \mid \text{blue}) = 2/9 \\

P(1 \mid \text{vio}) P(\text{sq} \mid \text{vio}) &= (1/2)(1/2) = P(1 \text{ sq} \mid \text{vio}) = 1/4 \\
P(2 \mid \text{vio}) P(\text{sq} \mid \text{vio}) &= (1/2)(1/2) = P(2 \text{ sq} \mid \text{vio}) = 1/4 \\
P(1 \mid \text{vio}) P(\text{cle} \mid \text{vio}) &= (1/2)(1/2) = P(1 \text{ cle} \mid \text{vio}) = 1/4 \\
P(2 \mid \text{vio}) P(\text{cle} \mid \text{vio}) &= (1/2)(1/2) = P(2 \text{ cle} \mid \text{vio}) = 1/4 \\
\end{align*}
\]

**b) Ip(V, S | C) (same as a)**

**c) Ip(V, S | C) (same as a)**

**d) Ip(V, S) NO!**

\[
P(1)P(\text{square}) = (5/13)(8/13) \neq P(1 \text{ square}) = 3/13
\]
Bayesian Network Semantics

Qualitative part
DAG specifies conditional independence statements

Quantitative part
+ local probability models

Unique joint distribution over domain

❖ The joint distribution decomposes nicely:

\[
\]

versus

\[
\]

❖ Every node is dependent on many others, BUT:

- \( \leq k \) parents \( \Rightarrow O(2^k n) \) vs. \( O(2^n) \) params
- => good for memory conservation & learning robustness

(Nir Friedman)
Constructing a Bayesian Network

- **causal DAG**: given a set of observed variables $V$, if for every $X, Y \in V$ we draw an edge from $X$ to $Y$ iff $X$ is a direct cause of $Y$ relative to $V$

- $X$ is a **direct cause** of $Y$ relative to $V$ iff a manipulation of $X$ changes the probability distribution of $Y$ and there is no subset $W \subseteq V - \{X, Y\}$ such that if we instantiate the variables in $W$ a manipulation of $X$ no longer changes the probability distribution of $Y$.

- **We can construct a Bayesian Network by creating a causal DAG** (proof not shown)
Constructing a BN: Pitfalls

1. Causes and their effects are statistically correlated.
2. But, variables can be correlated without causing one another.
3. For ex., let F & G be statistically correlated. Then all of the following causal relationships could account for F & G being correlated:
   1. 
   2. 
   3. 
   4. 
   5. 
   6. F&G are not causally related at all!

Selection bias or “discounting”: each cause explains away the occurrence of the effect.
Bayesian Network Learning

• In the simplest case, a Bayesian Network is specified by a domain expert (e.g. an M.D. for a medical diagnosis model) and is then used to perform inference.

• However, there are several algorithms for Bayesian Network learning:
  – Parameter Learning
    • Expectation-maximization algorithm
  – Structure Learning
    • Markov chain Monte Carlo
References

2. Wikipedia (Machine Learning and links within)