CS3700: Data Mining and Machine Learning

Machine Learning Overview
Prof. Chris Clifton
16 January 2020

The data mining process

- Data selection
- Preprocessing
- Knowledge extraction
- Interpretation and evaluation
- Mining patterns
- Machine learning
- Knowledge generation
Real-world example:

- 659,000 brokers
- 171,000 branches
- 5,100 firms
- 400,000 disclosures
Performance of NASD models

"One broker I was highly confident in ranking as 5...

Not only did I have the pleasure of meeting him at a shady warehouse location, I also negotiated his bar from the industry... This person actually used investors' funds to pay for personal expenses including his trip to attend a NASD compliance conference!

…If the model predicted this person, it would be right on target."

The data mining process

Data
- Use public data from NASD BrokerCheck

Selection
- Extract data about small firms in a few geographic locations

Target data

Preprocessing
- Create class label, temporal features

Processed data

Knowledge
- Evaluate objectively on historical data, subjectively with fraud analysts

Interpretation

Patterns
- Learn decision trees, output predictions and tree structure

Mining
Elements of Data Mining & Machine Learning Algorithms

- Task specification
- Data representation
- Knowledge representation
- Learning technique
  - Search + scoring
- Prediction and/or interpretation

Task specification

- **Objective of the person who is analyzing the data**
- **Description of the characteristics of the analysis and desired result**

- Examples:
  - From a set of *labeled examples*, devise an *understandable model* that will *accurately predict* whether a stockbroker will commit fraud in the near future.
  - From a set of *unlabeled examples*, cluster stockbrokers into a *set of homogeneous groups* based on their demographic information.
Exploratory data analysis

• Goal
  – Interact with data without clear objective

• Techniques
  – Visualization, ad hoc modeling

Descriptive modeling

• Goal
  – Summarize the data or the underlying generative process

• Techniques
  – Density estimation, cluster analysis and segmentation

Also known as: unsupervised learning
Predictive modeling

• Goal
  – Learn model to predict unknown class label values given observed attribute values
• Techniques
  – Classification, regression

Also known as: supervised learning

Pattern discovery

• Goal
  – Detect patterns and rules that describe sets of examples
• Techniques
  – Association rules, graph mining, anomaly detection

Model: global summary of a data set
Pattern: local to a subset of the data
Overview

- Task specification
- Data representation
- Knowledge representation
- Learning technique
  - Search + scoring
- Prediction and/or interpretation

Data representation

- *Choice of data structure* for representing individual and collections of measurements

- Individual measurements: single observations (e.g., person’s date of birth, product price)
- Collections of measurements: sets of observations that describe an instance (e.g., person, product)
- Choice of representation determines applicability of algorithms and can impact modeling effectiveness
- Additional issues: data sampling, data cleaning, feature construction
Individual measurements

- Unit measurements:
  - Discrete values — categorical or ordinal variables
  - Continuous values — interval and ratio variables
- Compound measurements:
  - \(< x, y >\)
  - \(< \text{value, time} >\)

Data representation: Table/vectors

<table>
<thead>
<tr>
<th>Fraud</th>
<th>Age</th>
<th>Degree</th>
<th>StartYr</th>
<th>Series7</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>22</td>
<td>Y</td>
<td>2005</td>
<td>N</td>
</tr>
<tr>
<td>-</td>
<td>25</td>
<td>N</td>
<td>2003</td>
<td>Y</td>
</tr>
<tr>
<td>-</td>
<td>31</td>
<td>Y</td>
<td>1995</td>
<td>Y</td>
</tr>
<tr>
<td>-</td>
<td>27</td>
<td>Y</td>
<td>1999</td>
<td>Y</td>
</tr>
<tr>
<td>+</td>
<td>24</td>
<td>N</td>
<td>2006</td>
<td>N</td>
</tr>
<tr>
<td>-</td>
<td>29</td>
<td>N</td>
<td>2003</td>
<td>N</td>
</tr>
</tbody>
</table>

\(N\) instances \(X\) \(p\) attributes
Data representation: Time series/sequences

Data representation: Relational/graph data
Overview

• Task specification
• Data representation
• Knowledge representation
• Learning technique
  – Search + scoring
• Prediction and/or interpretation

Knowledge representation

• **Underlying structure of the model or patterns that we seek from the data**
  – Specifies the models/patterns that could be returned as the results of the data mining algorithm
  – Defines the **model space** that algorithms search over (i.e., all possible models/patterns)

• Examples:
  – **If-then rule**
    If short closed car then toxic chemicals
  – **Conditional probability distribution**
    \[ P(\text{fraud} | \text{age}, \text{degree}, \text{series7}, \text{startYr}) \]
  – **Decision tree**
Each node corresponds to a feature; each leaf a class label or probability distribution.

Knowledge representation: Classification tree

Knowledge representation: Regression model

\[ y = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_0 \]

- X are predictor variables
- Y is response variable
- Example:
  - Predict number of disclosures given income and trading history
Overview

• Task specification
• Data representation
• Knowledge representation
• Learning technique
  – Search + scoring
• Prediction and/or interpretation

Learning technique

• Method to construct model or patterns from data
• **Model space**
  – Choice of knowledge representation defines a set of possible models or patterns
• **Scoring function**
  – Associates a numerical value (score) with each member of the set of models/patterns
• **Search technique**
  – Defines a method for generating members of the set of models/patterns and determining their score
Scoring function

- A numeric score assigned to each possible model in a search space, given a reference/input dataset
  - Used to judge the quality of a particular model for the domain

- Score function are statistics—estimates of a population parameter based on a sample of data

- Examples:
  - Misclassification
  - Squared error
  - Likelihood

Parameter estimation vs. structure learning

- Models have both parameters and structure
- Parameters:
  - Coefficients in regression model
  - Feature values in classification tree
  - Probability estimates in graphical model
- Structure:
  - Variables in regression model
  - Nodes in classification tree
  - Edges in graphical model

Search: Convex/smooth optimization techniques

Search: Heuristic approaches for combinatorial optimization
Example learning problem

Knowledge representation:
If-then rules

Example rule:
If \( x > 25 \) then
Else

What is the model space?
All possible thresholds

What score function?
Prediction error rate

CS3700:
Data Mining and Machine Learning

ML Overview: Continued
Prof. Chris Clifton
21 January 2020
Score function over model space

Search procedure?
Try all thresholds, select one with lowest score

Note: learning result depends on data

Classification tree

How many unique classification trees are there?
Search space

- Can we search exhaustively?
- Simplifying assumptions
  - Binary tree
  - Fixed depth
  - 10 binary attributes

<table>
<thead>
<tr>
<th>Tree depth</th>
<th>Number of trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>$8 \times 10^2$</td>
</tr>
<tr>
<td>3</td>
<td>$3 \times 10^6$</td>
</tr>
<tr>
<td>4</td>
<td>$2 \times 10^{13}$</td>
</tr>
<tr>
<td>5</td>
<td>$5 \times 10^{25}$</td>
</tr>
</tbody>
</table>

Overview

- Task specification
- Data representation
- Knowledge representation
- Learning technique
  - Search + Evaluation
- Prediction and/or interpretation
Inference and interpretation

- Prediction technique
  - Method to apply learned model to new data for prediction/analysis
  - Only applicable for predictive and some descriptive models
  - Prediction is often used during learning (i.e., search) to determine value of scoring function

- Interpretation of results
  - Objective: significance measures
  - Subjective: importance, interestingness, novelty

Example: Identifying email spam

- Task
  - Design automatic spam detector that can differentiate between labeled emails

- Data
  - Table of relative word/punctuation frequencies

- Knowledge representation
  - If/then rules with conjunctions of features

- Learning technique
  - Search over set of rules, select rule with maximum accuracy on training data

<table>
<thead>
<tr>
<th></th>
<th>george</th>
<th>you</th>
<th>your</th>
<th>hp free</th>
<th>hpl</th>
<th>!</th>
<th>our</th>
<th>re</th>
<th>edu</th>
<th>remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>spam</td>
<td>0.00</td>
<td>2.26</td>
<td>1.38</td>
<td>0.02</td>
<td>0.52</td>
<td>0.01</td>
<td>0.51</td>
<td>0.51</td>
<td>0.13</td>
<td>0.01</td>
</tr>
<tr>
<td>email</td>
<td>1.27</td>
<td>1.27</td>
<td>0.44</td>
<td>0.90</td>
<td>0.07</td>
<td>0.43</td>
<td>0.11</td>
<td>0.18</td>
<td>0.42</td>
<td>0.29</td>
</tr>
</tbody>
</table>

if ($\%\text{george} < 0.6$) & ($\%\text{you} > 1.5$) then spam
else email.
Example: Automatically identify digits on envelopes

- **Task**
  - Predict digit from image of handwritten envelopes
- **Data**
  - 16x16 matrix of pixel intensities
- **Knowledge representation**
  - Nearest neighbor classifier, defines tessellation over the feature space
- **Learning technique**
  - Implicit search for decision boundaries to minimize mis-classifications on training data

![Examples of handwritten digits from U.S. postal envelopes.](image)

Example: DNA expression levels in cancer tumors

- **Task**
  - **Unsupervised** learning:
    Examine DNA microarrays to determine which tumors are similar and which genes are similar
- **Data**
  - Expression levels [-6,6] for 6830 genes (rows) in 64 cancer tumors (columns) from different patients
- **Knowledge representation**
  - Clusters of similar genes/tumors
- **Learning technique**
  - Search over groups, minimize distance to group centroid

![DNA microarray data expression matrix.](image)
Summary: Data-Mining Pipeline

- **Define the task** you care about
- **Collect data** relevant for that task
- Get to know your data:
  - Find outliers, noise
  - What are the properties (features) that we care about for the task?
- **Learn a model** (function, hypothesis)
  - decisions: search space, search algorithm, scoring function
- **Evaluate** the learned model: How can you quantify performance?

Your First Classifier!

- Let’s consider one of the simplest classifiers out there.
- Assume we have a training set \((x_1, y_1) \ldots (x_n, y_n)\)
- Now we get a new instance \(x_{\text{new}}\), how can we classify it?
  - Example: Can you recommend a movie, based on user’s movie reviews?
- **Simple Solution:**
  - Find the most similar example \((x, y)\) in the training data and predict the same
  - If you liked “Fast and Furious” you’ll like “2 fast 2 furious”
- Only a single decision is needed: distance metric to compute similarity

\[
d(x_1, x_2) = 1 - \frac{x_1 \cap x_2}{x_1 \cup x_2} \quad d(x_1, x_2) = \sqrt{(x_1 - x_2)^2}
\]
K Nearest Neighbors

– Can you think about a better way?
– We can make the decision by looking at several near examples, not just one. Why would it be better?

K Nearest Neighbors

- **Learning**: just storing the training examples
- **Prediction**:
  - Find the K training example closest to $x$
- **Predict a label**:
  - Classification: majority vote
  - Regression: mean value
- **KNN is a type of instance based learning**
- **This is called lazy learning**, since most of the computation is done at prediction time
Let’s analyze KNN

- **What are the advantages and disadvantages of KNN?**
  - What should we care about when answering this question?

- **Complexity**
  - **Space** (how memory efficient is the algorithm?)
    - Why should we care? KNN needs to maintain all training examples!
  - **Time** (computational complexity)
    - Both at training time and at test (prediction) time
    - Datasets can be HUGE

- **Expressivity**
  - What kind of functions can we learn? Training is very fast! But *prediction is slow*
    - \(O(dN)\) for \(N\) examples with \(d\) attributes
    - *increases* with the number of examples!

Analyzing K Nearest Neighbors

- We discussed the importance of finding a good model space
  - Expressive (we can represent the right model)
  - Constrained (we can search effectively, using the data we have)

- Let’s try to characterize the model space, by looking at the **decision boundary**

- **How would it look if K=1?**

  If we define the model space to be our choice of \(K\)
  Does the complexity of the model space increase or decrease with \(K\)?
Analyzing K Nearest Neighbors

- Which model has a higher K value?
- Which model is more complex?
- Which model is more sensitive to noise?

Questions

- We know higher K values result in a smoother decision boundary.
  - Less "jagged" decision regions
  - Total number of regions will be smaller

  What will happen if we keep increasing K, up to the point that K=n?
  n = is the number of examples we have
How should we determine the value of \( K \)?

- Higher \( K \) values result in less complex functions (less expressive)
- Lower \( K \) values are more complex (more expressive)
- **How can we find the right balance between the two?**
- Option 1: Find the \( K \) that minimizes the training error.
  - Training error: after learning the classifier, what is the number of errors we get on the training data.
  - What will be this value for \( k=1, k=n, k=n/2 \)?
- Option 2: Find \( K \) that minimizes the **validation error**.
  - Validation error: set aside some of the data (validation set). what is the number of errors we get on the validation data, after training the classifier.

**In general** – using the training error to tune parameters will always result in a more complex hypothesis! *(why?)*
KNN Practical Consideration

- Finding the right representation is key
  - KNN is very sensitive to irrelevant attributes
- Choosing the right distance metric is important
  - Many options!
  - Popular choices:

\[
\begin{align*}
\text{Euclidean distance} & : \| x_1 - x_2 \|_2 = \sqrt{\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2} \\
\text{Manhattan distance} & : \| x_1 - x_2 \|_1 = \sum_{i=1}^{n} |x_{1,i} - x_{2,i}| \\
\text{L_p-norm} & : \begin{align*}
\text{Euclidean} & = L_2 \\
\text{Manhattan} & = L_1 \\
\| x_1 - x_2 \|_p & = \left( \sum_{i=1}^{n} |x_{1,i} - x_{2,i}|^p \right)^{\frac{1}{p}}
\end{align*}
\]
Beyond KNN

• KNN is not a statistical classifier.
• It memorizes the training data, and makes a majority vote over the K closest points.
• For example, these two cases are the same:

  ![Diagram showing two cases](image)

• What is the difference between the two scenarios?
• How can we reason about it?