Overview of Inductive Learning

Daniel Lowd

Ungraded Reading Quiz

1. Turn to the person next to you and introduce yourself.
2. If your name come first alphabetically, answer the first question for your neighbor. If your name comes second alphabetically, explain the answer to the second question to your neighbor.
3. If you have time, feel free to answer the last two questions as well.

Q1: What is a loss function? Give three examples.
Q2: What is the difference between the data distribution and the training data?
Q3: Give an example of a real-world classification problem. What’s the source of data? What loss function would you use?
Q4: Give an example of a real-world regression problem. What’s your source of data? What loss function would you use?
Goal: Use course attributes to predict the student’s course rating (+2 = loved it, -2 = hated it).

How should we model this?

(Data is from CML appendix.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+2</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>+2</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>+2</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>+2</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>+2</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>+1</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>+1</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>+1</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>0</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>0</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>

Warm-up Exercise

Consider the following dataset:

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- Draw a decision tree that classifies all examples correctly.
- How would your tree classify (0, 1, 1)?
Inductive Learning

Informally: Learning is about using experience to improve performance.

Formally: Given a loss function $\ell$ and a sample $D$ from some unknown distribution $\mathcal{D}$, you must compute a function $f$ that has low expected error $\varepsilon$ over $\mathcal{D}$ with respect to $\ell$.

$$\varepsilon = E_{(x,y)} \mathcal{D}[\ell(y, f(x))]$$

Inductive Bias

Since we don’t know $\mathcal{D}$, we must make do with the training error (or “empirical risk”) $\hat{\varepsilon}$ over training data $D$:

$$\hat{\varepsilon} = \frac{1}{N} \sum_{i=1}^{n} \ell(y^{(i)}, f(x^{(i)}))$$

Many different $f$ may have the same training error. Inductive bias is how we choose among them.

Without bias we cannot learn!
Tuning and Testing

• How do we know if $f$ is good?
  – Can’t use data generating distribution directly
  – Low training error could be misleading – easy to memorize data and overestimate performance.

• Average loss on previously unseen data is a much better indicator of future performance.
  – Use held out validation data to choose algorithms and hyperparameters (algorithm settings) that are likely to generalize well.
  – Use separate test data for final evaluation.

Important Concepts

• Data: labeled instances, e.g. emails marked spam/ham
  – Training set
  – Held out set
  – Test set

• Features: attribute-value pairs which characterize each $x$

• Experimentation cycle
  – Learn parameters (e.g. model probabilities) on training set
  – (Tune hyperparameters on held-out set)
  – Compute accuracy of test set
  – Very important: never “peek” at the test set!

• Evaluation
  – Accuracy: fraction of instances predicted correctly

• Overfitting and generalization
  – Want a classifier which does well on test data
  – Overfitting: fitting the training data very closely, but not generalizing well
The Big Picture

Problem Domain
- (email spam)
  - Data Representation
    - ("bag of words")
  - Objective Function
  - Evaluation
    - (error rate on training emails)
    - (error rate on future emails)

Learning Algorithm
- (C4.5 algorithm)

Predictive Model
- (decision tree)
  - Useful Predictions
    - (automated filter)
  - Domain Insight
    - (types of spam)

The class mostly focuses on the red pieces, but the blue pieces are just as important!
ML in a Nutshell

• Tens of thousands of machine learning algorithms
• Hundreds new every year
• Every machine learning algorithm has three components:
  – Model Representation
  – Evaluation
  – Optimization

Model Representation

• Decision trees
• Instances
• Linear function (hyperplane)
• Neural networks
• Support vector machines
• Model ensembles
• (Sets of rules / Logic programs)
• (Graphical models (Bayes/Markov nets))
• Etc.
Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.

Optimization

- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming