CS367: AI
Tutorial 8

Machine Learning

Feedback to: goo.gl/auRHe

Machine Learning Fundamentals

- A machine learning application can be defined by three aspects:
  - T: a class of tasks
  - P: a performance measure
  - E: experience
- eg. For a spam filter:
  - T: classifying email messages as spam / not spam
  - P: percentage of correct classifications
  - E: a database of emails and their classifications
- We want to find a *target function*, V, which allows us to predict the correct outcome in a situation.

Machine Learning Fundamentals

- There are many ways we could express V
  - eg. a weighted linear combination of attributes, a decision tree, a similarity to past examples (CBR), a neural network, a genetic algorithm...
  - Each has a trade-off in expressability vs complexity and time (or number of examples) taken to learn
- To start with, we used a weighted linear combination of attributes
  - *Attributes* are the things we are considering about the state/situation which may affect the outcome
  - In this we learn by incrementally reducing the error between our predicted outcome and the one seen in each training case, by adjusting the weights

Machine Learning Fundamentals

- Learning in machine learning is simply a search through a space of all possible hypotheses to find one that fits the data
- Depending on the representation we use, we may:
  - restrict the area in the total space able to be searched
    - This is restriction bias (completely searching an incomplete hypothesis space) – e.g. candidate elimination algorithm
  - focus the search in particular areas (preferred)
    - This is preference bias (incompletely searching a complete hypothesis space) – e.g. ID3 decision tree algorithm

Machine Learning Fundamentals

- Machine learning algorithms have *inductive bias*. It is required to make generalisations or "inductive leaps" to work on never-before-seen examples
  - An unbiased learner can be called a rote learner or database
  - Machine learning algorithms tend to work better when the assumptions underlying their biases are correct in the situation in which they are being used
  - Some machine learning algorithms have stronger bias than others. This does not necessarily mean they are worse.

Machine Learning: Question

- What is the difference between the bias of an algorithm and the *variance* of an algorithm?
- The bias of an algorithm is how far the average error (or accuracy) is from the true error. The true error is based on the whole distribution of data points in the universe (not just the training set).
- The variance is a measure of the difference between the different runs of the algorithm from one another.

Concept Learning

- Concept learning is inferring a boolean-valued function from training examples of its input and output
- Related: Classification (same but discrete-valued) and Regression (same but continuous)
- Training examples have a attribute-vector and the corresponding expected output
- By learning from many training examples these algorithms can find a way to represent them and predict an output for a given attribute-vector

Concept Learning

- Hypothesis is formed as (a) attribute vector(s) for a positive output, with two additions:
  - ? represents any value is acceptable
  - 0 represents no value is acceptable
  - eg. <Sunny,?,?,?> is general (is not specific)
<Sunny,0,0,0> is not general (is specific)
- Algorithms given to find the most-specific possible hypotheses (Find-S) and most general possible hypotheses (List-Then-Eliminate)

Concept Learning

- A useful set of hypotheses to consider are the ones between the most general and most specific possible
  - These are all hypotheses consistent with the training examples – the *Version Space*
  - The Version Space can be represented by the most-general *G* and most-specific *S* boundaries
  - These boundaries can be found using the candidate-elimination algorithm

Concept Learning: Question

- If you had the sets S and G shown below and each attribute is binary as shown, what would all the hypotheses in your version space be?
  - Attributes: <{yellow, blue}, {ball, square}, {tall, short}, {2D, 3D}, {large, tiny}>
  - S: <<yellow, ball, tall, 2D, ?>>
  - G: <<yellow, ?, ?, ?, ?><?, ?, ?, 2D, ?>>
- S and G, plus…
- 3 “?”s:
  - <yellow, ball, ?, ?, ?><yellow, ?, tall, ?, ?>
    <yellow, ?, ?, 2D, ?>
    <?, ball, ?, 2D, ?><?, ?, tall, 2D, ?>
- 2 “?”s:
  - <yellow, ball, tall, ?, ?>
    <yellow, ball, ?, 2D, ?><yellow, ?, tall, 2D, ?>
    <?, ball, tall, 2D, ?>

Candidate-Elimination Algorithm

Initialise G to the set of maximally general hypotheses in H

Initialise S to the set of maximally specific hypotheses in H

For each training example d, do:

- If d is a positive example:
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d:
    - Remove s from S
    - Add to S all minimal generalisations h of s such that h is consistent with d, and some member of G is more general than h
    - Remove from S any hypothesis that is more general than another hypothesis in S
- If d is a negative example:
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specialisations h of g such that h is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Candidate Elimination Example

Origin	Manufacturer	Color	Decade	Type	Example Type
Japan	Honda	Blue	1980	Economy	Positive
Japan	Toyota	Green	1970	Sports	Negative
Japan	Toyota	Blue	1990	Economy	Positive
USA	Chrysler	Red	1980	Economy	Negative
Japan	Honda	White	1980	Economy	Positive
Japan	Toyota	Green	1980	Economy	Positive
Japan	Honda	Red	1990	Economy	Negative

Taken from 2010 slides

Positive Example 1
(Japan, Honda, Blue, 1980, Economy)

- Initialize G to a singleton set that includes everything.

G = { (?, ?, ?, ?, ?) }

- Initialize S to a singleton set that includes the first positive example.

S = { (Japan, Honda, Blue, 1980, Economy) }

Negative Example 2
(Japan, Toyota, Green, 1970, Sports)

- Specialize G to exclude the negative example.

G = {
(?, Honda, ?, ?, ?),
(?, ?, Blue, ?, ?),
(?, ?, ?, 1980, ?),
(?, ?, ?, ?, Economy) }

- S stays the same

S = { (Japan, Honda, Blue, 1980, Economy) }

Positive Example 3
(Japan, Toyota, Blue, 1990, Economy)

- Prune G to exclude descriptions inconsistent with the positive example.

G = {
(?, ?, Blue, ?, ?),
(?, ?, ?, ?, Economy) }

- Generalize S to include the positive example.

S = { (Japan, ?, Blue, ?, Economy) }

Negative Example
(USA, Chrysler, Red, 1980, Economy)

- Specialize G to exclude the negative example

G = {
(?, ?, Blue, ?, ?),
(Japan, ?, ?, ?, Economy) }

- S stays the same

S = { (Japan, ?, Blue, ?, Economy) }

Positive Example
(Japan, Honda, White, 1980, Economy)

- Prune G to exclude descriptions inconsistent with positive example.

G = { (Japan, ?, ?, ?, Economy) }

- Generalize S to include positive example.

S = { (Japan, ?, ?, ?, Economy) }

Positive Example
(Japan, Toyota, Green, 1980, Economy)

- New example is consistent with version-space, so no change is made.

G = { (Japan, ?, ?, ?, Economy) }

S = { (Japan, ?, ?, ?, Economy) }

Negative Example
(Japan, Honda, Red, 1990, Economy)

- Example is inconsistent with the version-space.

- G cannot be specialized.
- S cannot be generalized.

- The version space **collapses**.
- <u>Conclusion:</u> No conjunctive hypothesis is consistent with the data set.

Remarks on Version Spaces and Candidate Elimination

- The version space learned by the Candidate-Elimination Algorithm will converge toward the hypothesis that correctly describes the target concept provided: (1) There are no errors in the training examples; (2) There is some hypothesis in *H* that correctly describes the target concept.
  - The candidate elimination algorithm is **biased** by the assumption that the target function can be represented as a conjunction of attribute values
- Convergence can be sped up by presenting the data in a strategic order. The best examples are those that satisfy exactly half of the hypotheses in the current version space.