Unit 5: AI & Machine Learning

Dave Abel

February 29th, 2016
Field Trip!
Field Trip!

› Three scheduled times in the Yurt! Limited space :(
  - Wednesday 3/9 from 2pm-3pm
  - Wednesday 3/16 3pm-4pm
  - Wednesday 3/23 1pm-2pm

› If you want to go:
  - Send me an email (david_abel@brown.edu) with subject “CS8 Yurt”
  - In the message, list your date/time preferences from 1-3, 1 being top preference, 3 being bottom preference (only list those that you can actually make).
Machine Learning Takeaway

- **Learning can be represented as an algorithm!**

- Several kinds of learning, each requires different teaching/training styles.
  - Classification! (Today, Wednesday)
  - Reinforcement! (Wednesday, Friday)
Outline

› Overview of AI

› Some examples of recent success.

› Classification
  - Setup, features
  - Memorize and Guess
  - Nearest Neighbor

› Training Process

› Testing Process

› Overfitting, Occam’s Razor
Overview of AI

http://www.atlasobscura.com/places/canard-digerateur-de-vaucanson-vaucansons-digesting-duck
Overview of AI

http://www.atlasobscura.com/places/canard-digerateur-de-vaucanson-vaucansons-digesting-duck

1739
Overview of AI

"without...the duck of Vaucanson, you would have nothing to remind you of the glory of France."
- Voltaire

http://www.atlasobscura.com/places/canard-digerateur-de-vaucanson-vaucansons-digesting-duck
Overview of AI
Dartmouth Summer 1956

“We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.”

- John McCarthy, 1955
Dartmouth Summer 1956

› Automatic Computers
› How Can a Computer be Programmed to Use a Language
› Neuron Nets
› Theory of the Size of a Calculation
› Self-Improvement
› Abstractions
› Randomness and Creativity
Dartmouth Summer 1956

- Automatic Computers
- How Can a Computer be Programmed to Use a Language
- Neuron Nets
- Theory of the Size of a Calculation
- Self-Improvement
- Abstractions
- Randomness and Creativity

(60 years later…)
Some Recent Success

Cloth Grasp Point Detection based on Multiple-View Geometric Cues with Application to Robotic Towel Folding

Jeremy Maitin-Shepard
Marco Cusumano-Towner
Jinna Lei
Pieter Abbeel

Department of Electrical Engineering and Computer Science
University of California, Berkeley

International Conference on Robotics and Automation, 2010
Some Recent Success
Some Recent Success
Some Recent Success
“The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”
The Field of AI

- Learning
- Perception
- Language
- Reasoning, Planning
- Motion and Manipulation
- Knowledge Representation

“The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”
The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.

- Learning
  - Perception
  - Language
- Reasoning, Planning
  - Motion and Manipulation
- Knowledge Representation

“The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”
"The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."
Machine Learning: Classification
Machine Learning: Classification

Snozzberry
Machine Learning: Classification

- Snozzberry
- Buzzberry
Machine Learning: Classification

- Snozzberry
- Buzzberry
- Fizzberry
Machine Learning: Classification

- Snozzberry
- Buzzberry
- Fizzberry
- Snozzberry
Machine Learning: Classification

Snozzberry
Buzzberry
Fizzberry
Snozzberry

Training!
Q: What should this be called?
Machine Learning: 
Classification

Q: What should this be called?

A: Buzzberry!

Training!
Problem: Classification

- INPUT: A bunch of labeled training data, a discrete set of possible labels.

- OUTPUT: A classifier! (that best classifies all objects in the space).
Problem: Classification

- **INPUT:** A bunch of labeled training data, a discrete set of possible labels.

- **OUTPUT:** A classifier! (that best classifies all objects in the space).
A Classifier

- Takes an object from the space of interest (pieces of fruit, shapes, images, words, etc.).

- Reports a *label*, from the discrete set of possible labels.

- Could do this according to a simple rule! (If blue, snozzberry!)

- Could be a complicated logical rule.

- **In general: Could be any (Scratch) program!**
Problem: Classification

- **INPUT**: A bunch of labeled training data, a discrete set of possible labels.

- **OUTPUT**: A classifier! (that best classifies all objects in the space).
A Classifier: Fruit Example

- More Red
- More Blue
- Bigger
- Smaller
A Classifier: Fruit Example

More Blue  \rightarrow  More Red

Smaller  \downarrow  Bigger
A Classifier: Fruit Example

Bigger

More Blue  More Red

Smaller
A Classifier: Fruit Example

- More Red
- More Blue
- Bigger
- Smaller

A strawberry is placed in the quadrant that represents a fruit with more red and smaller size.
A Classifier: Fruit Example

More Blue

Bigger

More Red

Smaller
Features

= (4, 5)

More Blue

Bigger

More Red

= (3, -4)

Smaller

Idea: these numbers totally represent each fruit!
Clicker Question!

Q: Which of the following might be reasonable features for classifying a recipe’s cuisines?
Clicker Question!

A list of cooking tools required to make the food

A list of spices in the recipe

The name of the recipe

The country the recipe comes from

Q: Which of the following might be reasonable features for classifying a recipe’s cuisine?
Q: Which of the following might be reasonable features for classifying a recipe’s cuisine?

[A] A list of cooking tools required to make the food

[B] A list of spices in the recipe

[C] The name of the recipe

[D] The country the recipe comes from
Q: Which of the following might be reasonable features for classifying a recipe’s cuisines?

[A] A list of cooking tools required to make the food

[B] A list of spices in the recipe

[C] The name of the recipe

[D] The country the recipe comes from

Features just need to contain some information that bears on the class of the object!
Clicker Question!

Q: Which feature above will be most useful for classifying SPAM vs. HAM emails?
Clicker Question!

[A] Is the email longer than 500 words?

[B] Is .edu in the email address?

[C] Does the email contain a link?

[D] Does the email contain the word “credit card”?

Q: Which feature above will be most useful for classifying SPAM vs. HAM emails?
Clicker Answer!

[A] Is the email longer than 500 words?

[B] Is .edu in the email address?

[C] Does the email contain a link?

[D] Does the email contain the word “credit card”?

Q: Which feature above will be most useful for classifying SPAM vs. HAM emails?
Q: Which feature above will be most useful for classifying SPAM vs. HAM emails?

[A] Is the email longer than 500 words?

[B] Is .edu in the email address?

[C] Does the email contain a link?

[D] Does the email contain the word “credit card”?  

Note: could also imagine the AND of D and C would be really effective as a feature!
A Classifier: Fruit Example

More Blue

More Red

Bigger

Smaller

- = strawberry
- = apple
A Classifier: Fruit Example

More Blue  

More Red

Bigger

Smaller

Classifier

○ = strawberry

● = apple
A Classifier: Fruit Example

- Blue = strawberry
- Red = apple

Classifier
Machine Learning

- **INPUT:** Some labeled training data (some fruits, with their name)

- **OUTPUT:** A classifier.

- **Goal:** Spit out the classifier that will classify the most things correctly.

- **Tools:** What do you do with the training data you receive to inform your classifier?
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Goal: Spit out the classifier that will classify the most things correctly.
Machine Learning

- INPUT: Some labeled *training data* (some fruits, with their name)

- OUTPUT: A classifier.

- **Goal:** Spit out the classifier that will classify the most things *correctly*.

- *Training Data:* finite! Only represents a small space of the thing we’re trying to learn.
Training Data

- = strawberry
- = apple

Note: There are way more than 10 apples in the world
Training Data

- ○ = strawberry
- ● = apple

Note: There may be huge strawberries!
Training Data

- ○ = strawberry
- ● = apple

Note: There may be huge strawberries! And tiny apples.
Training Data

Note: There may be huge strawberries! And tiny apples.
Training Data

There is no line that correctly classifies our training data.

Note: There may be huge strawberries! And tiny apples.
Classification

1. The algorithm receives a bunch of labeled training data.

2. The algorithm tries to use the labels to learn the best classifier it can (rule for distinguishing between classes).

3. We **evaluate** learning algorithms according to how well their classifier is able to classify things more generally. This round is called, “testing” (so the data used to test is called “testing data”).
Testing Data

○ = strawberry
● = apple

More Blue  Bigger  More Red

Smaller
Testing Data

- = strawberry
- = apple
- = testing data
- = training data
Testing Data

Q: How well does this classifier do?
Testing Data

= strawberry

= apple

= testing data

= training data

Q: How well does this classifier do?

A: Super well! Only misses one
Q: How well does this classifier do?

A: Super well! Only misses one
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, **guess the label randomly**.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.

Q1: Halt?
1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.

Q1: Halt?
A: Yep!
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.

Q2: Correct?
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when **given any item seen in our training data, reports exactly that item’s label**. If it gets an item it’s never seen before, guess the label randomly.

Q2: Correct?

A: We return a classifier, but it might be impossible to create a perfect classifier.
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.

Q3: Growth Rate?
Our First Classification Algorithm: Memorize And Guess

1. Memorize every training data-label pair we see.

2. Create a classifier that, when given any item seen in our training data, reports exactly that item’s label. If it gets an item it’s never seen before, guess the label randomly.

Q3: Growth Rate?

A: Have to memorize $N$ things. So let’s say $N$. 
Our First Classification Algorithm: Memorize And Guess

Side note: in learning there’s a different metric that roughly translates to “how many experiences do I need to perform well?”

We’ll revisit this in reinforcement learning

Q3: Growth Rate?

A: Have to memorize $N$ things. So let’s say $N$. 
Our First Classification Algorithm: Memorize And Guess

- So, is this Memorize and Guess thing a good idea?
- Sure! If you’re guaranteed to see just about every possible data point during training.
- Well that’s crazy…
- Conclusion: no. Memorize and Guess is a bad idea.
- Q: Can we do better?
- A: Of course!