Course overview and general classes of problems

Hal Daumé III

CS 726: Machine Learning

1 September 2011

Course Background

What is this course about?

- Finding (and exploiting) patterns in data
- Replacing “human writing code” with “human supplying data”
  - ⇒ System figures out what the person wants based on examples
  - ⇒ Need to abstract from “training” examples to “test” examples
  - ⇒ Most central issue in ML: *generalization*

Course Background

What is this course about?

- Finding (and exploiting) patterns in data
- Replacing "human writing code" with "human supplying data"
  - ⇒ System figures out what the person wants based on examples
  - ⇒ Need to abstract from "training" examples to "test" examples
  - ⇒ Most central issue in ML: *generalization*

Why is machine learning so cool?

- Broad applicability
  - Finance, robotics, vision, machine translation, medicine, etc.
- Close connection between theory and practice
- Open field, lots of room for new work
- `http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9026623`

Course Goals

By the end of the semester, you should be able to:

- Look at a problem and identify if ML is an appropriate solution
- If so, identify what types of algorithms might be applicable
- Apply those algorithms
- Conquer the world

In order to get there, you will need to:

- Do a lot of math (calculus, linear algebra, probability)
- Do a fair amount of programming
- Work hard (this is a 3-credit class)

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything
- I think the projects should be changed and different to the previous semester

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything
- I think the projects should be changed and different to the previous semester
- This course shouldn't be CS 726, it should be MATH 726!

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything
- I think the projects should be changed and different to the previous semester
- This course shouldn't be CS 726, it should be MATH 726!
- All in all, great class, and clear interest in the needs of the students

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything
- I think the projects should be changed and different to the previous semester
- This course shouldn't be CS 726, it should be MATH 726!
- All in all, great class, and clear interest in the needs of the students
- ...and so on ...

Student comments on this course

- The field of machine learning seems too wide for any one-semester long course to completely cover everything
- I think the projects should be changed and different to the previous semester
- This course shouldn't be CS 726, it should be MATH 726!
- All in all, great class, and clear interest in the needs of the students
- ...and so on ...

I try to take your comments seriously!
(but some things won't change...)

Topics Covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world

Topics Covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world
  - Time-series models
  - Structured prediction
  - Semi-supervised learning
  - Large-scale learning

Topics Covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world
  - Time-series models
  - Structured prediction
  - Semi-supervised learning
  - Large-scale learning
- Not a zoo tour!
- Not an introduction to tools!

Topics Covered

- Supervised learning: learning with a teacher
- Unsupervised learning: learning without a teacher
- Complex settings: learning in a complicated world
  - Time-series models
  - Structured prediction
  - Semi-supervised learning
  - Large-scale learning
- Not a zoo tour!
- Not an introduction to tools!
- You will learn how these techniques work and how to implement them.

Syllabus

http://hal3.name/courses/2011F_ML/

On Reading and Responsibilities. . .

Reading: I expect you to do it!

(but most are $\leq$ 12 pages, all are $\leq$ 20)

Online book draft (minus the figures) linked off the web page. (Extra credit for bugs!)

Reading: I expect you to do it!

(but most are $\leq$ 12 pages, all are $\leq$ 20)

Online book draft (minus the figures) linked off the web page. (Extra credit for bugs!)

Class time is for:

- Discussing questions from the reading
  - There are questions in the margins: be prepared to answer them
- Discussing homework assignments
  - Some questions are starred: these will be presented in class
- Me providing an insider's view

Requirements and Grading

Programming projects: 27%

Written homeworks: 18%

Midterm exam: 25%
Final “practical” exam: 25%

Class/Piazza participation: 5%

Requirements and Grading

Programming projects: 27%

- Three total
- Teams of at most three
- May be 48 hours late, at 50% mark down

Written homeworks: 18%

Midterm exam: 25%
Final "practical" exam: 25%

Class/Piazza participation: 5%

Requirements and Grading

Programming projects: 27%

- Three total
- Teams of at most three
- May be 48 hours late, at 50% mark down

Written homeworks: 18%

- One per week, graded 0, 0.5 or 1
- Completed individually
- May not be late at all

Midterm exam: 25%
Final "practical" exam: 25%

Class/Piazza participation: 5%

Requirements and Grading

Programming projects: 27%

- Three total
- Teams of at most three
- May be 48 hours late, at 50% mark down

Written homeworks: 18%

- One per week, graded 0, 0.5 or 1
- Completed individually
- May not be late at all

Midterm exam: 25%

Final "practical" exam: 25%

- Canned or your choice, teams
- Presentations during the final slot

Class/Piazza participation: 5%

Grading, continued...

Grading complaints

- Not allowed after one week

Grading, continued. . .

Grading complaints

- Not allowed after one week

How should you spend your time?

- 3 hours in class
- 2 hours reading
- 2 hours on written assignments
- 2 hours on programming projects

Things that irk me. . . aka: I'm only human!

Things that irk me... aka: I'm only human!

- Questions that have already been answered
  Please read the class mailing list and come to class!

Things that irk me... aka: I'm only human!

- Questions that have already been answered
  Please read the class mailing list and come to class!
- Fire and forget emails
  I'd *much* rather talk to you in person!

Things that irk me. . . aka: I'm only human!

- Questions that have already been answered
  Please read the class mailing list and come to class!

- Fire and forget emails
  I'd *much* rather talk to you in person!

(See HW00 for your opportunity to respond!)

Things you need to do now!

Complete Homework 00

- Due 6 Sep (that's Tuesday!, by beginning of class)
- Submit in `.pdf` format *only* using handin

Complete the first reading

- See syllabus
- Due by class Tuesday (I mean it!)
- Some parts of the web page are password protected!

Sign up to get mails

- Subscribe to the Piazza group.
- But be sure to actually read it!

Read the web page!

Now, on to some real content. . .

(but first, questions?)

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?

- How would you write a program to determine whether a sentence is grammatical or not?

- How would you write a program to distinguish cancerous cells from normal cells?

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?
  - $\Rightarrow$ Provide examples pictures of me and pictures of other people and let a *classifier* learn to distinguish the two.
- How would you write a program to determine whether a sentence is grammatical or not?
- How would you write a program to distinguish cancerous cells from normal cells?

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?
  - ⇒ Provide examples pictures of me and pictures of other people and let a *classifier* learn to distinguish the two.
- How would you write a program to determine whether a sentence is grammatical or not?
  - ⇒ Provide examples of grammatical and *ungrammatical* sentences and let a *classifier* learn to distinguish the two.
- How would you write a program to distinguish cancerous cells from normal cells?

Classification

- How would you write a program to distinguish a picture of me from a picture of someone else?

  ⇒ Provide examples pictures of me and pictures of other people and let a *classifier* learn to distinguish the two.

- How would you write a program to determine whether a sentence is grammatical or not?

  ⇒ Provide examples of grammatical and *ungrammatical* sentences and let a *classifier* learn to distinguish the two.

- How would you write a program to distinguish cancerous cells from normal cells?

  ⇒ Provide examples of cancerous and *normal* cells and let a *classifier* learn to distinguish the two.

Data ("weather" prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Data ("weather" prediction)

Example dataset:

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label
2. Features
3. Feature values

Data ("weather" prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label (aka "label", denoted y)
2. Features
3. Feature values

Data ("weather" prediction)

Example dataset:

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label (aka "label", denoted y)
2. Features (aka "attributes")
3. Feature values

Data (“weather” prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label (aka “label”, denoted y)
2. Features (aka “attributes”)
3. Feature values (aka “attribute values”, denoted x)

Data (“weather” prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label (aka “label”, denoted y)
2. Features (aka “attributes”)
3. Feature values (aka “attribute values”, denoted x)
   $\Rightarrow$ Features can be binary, nomial or continuous

Data (“weather” prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Three principle components:

1. Class label (aka “label”, denoted y)
2. Features (aka “attributes”)
3. Feature values (aka “attribute values”, denoted x)
   - $\Rightarrow$ Features can be binary, nomial or continuous

A *labeled* dataset is a collection of (x, y) pairs

Data ("weather" prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Task:

Class	**Outlook**	**Temperature**	**Windy?**
???	Sunny	Low	No

Predict the class for this "test" example"

Data ("weather" prediction)

Example dataset:

Class	**Outlook**	**Temperature**	**Windy?**
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

Task:

Class	**Outlook**	**Temperature**	**Windy?**
???	Sunny	Low	No

Predict the class for this "test" example"

Requires us to **generalize** from the training data

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

What is a good *representation* for images?

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

What is a good *representation* for images? Pixel values?

Data (face recognition)

Class	Image
Avrim	
Avrim	
Avrim	
Avrim	

Class	Image
Tom	
Tom	
Tom	
Tom	

What is a good *representation* for images? Pixel values? Edges?

Ingredients for classification

Whole idea: Inject *your* knowledge into a learning system

Sources of knowledge:

1. Feature representation
2. Training data: labeled examples
3. Model

Ingredients for classification

Whole idea: Inject *your* knowledge into a learning system

Sources of knowledge:

1. Feature representation
   - Not typically a focus of machine learning
   - Typically seen as “problem specific”
   - *However*, it’s hard to learn on bad representations
2. Training data: labeled examples
3. Model

Ingredients for classification

Whole idea: Inject *your* knowledge into a learning system

Sources of knowledge:

1. Feature representation
   - Not typically a focus of machine learning
   - Typically seen as “problem specific”
   - *However*, it’s hard to learn on bad representations
2. Training data: labeled examples
   - Often expensive to label lots of data
   - Sometimes data is available for “free”
3. Model

Ingredients for classification

Whole idea: Inject *your* knowledge into a learning system

Sources of knowledge:

1. Feature representation
   - Not typically a focus of machine learning
   - Typically seen as "problem specific"
   - *However*, it's hard to learn on bad representations
2. Training data: labeled examples
   - Often expensive to label lots of data
   - Sometimes data is available for "free"
3. Model
   - No single learning algorithm is always good ("no free lunch")
   - Different learning algorithms work with different ways of representing the learned classifier
   - When the data has nothing to say, which model is better
   - Typically requires some control over generalization

More on generalization later. . .

Regression

Regression is like **classification**, except the labels are real values.

Example applications:

Regression

Regression is like **classification**, except the labels are real values.

Example applications:

- Stock value prediction
- Income prediction
- CPU Power consumption

Regression

Regression is like **classification**, except the labels are real values.

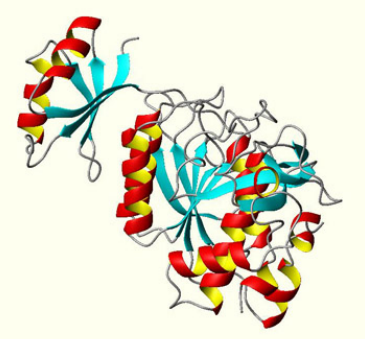
Example applications:

- Stock value prediction
- Income prediction
- CPU Power consumption
- Your grade in CS 726

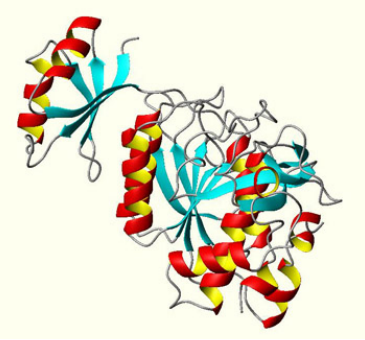
Structured Prediction

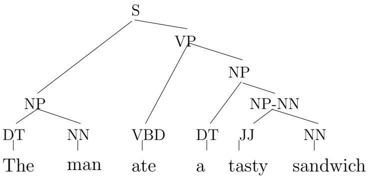
Structured Prediction

Structured Prediction

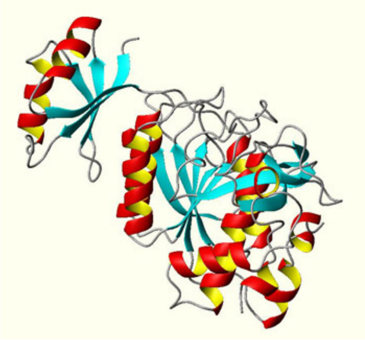


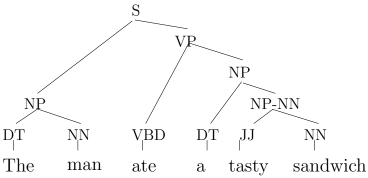
Structured Prediction

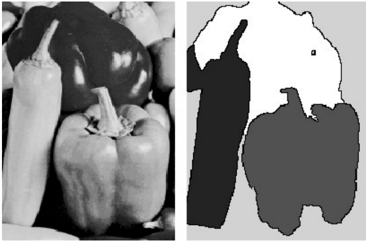




Structured Prediction







Unsupervised learning: Clustering

Two styles of clustering

1. Clustering into distinct components

2. Hierarchical clustering

Unsupervised learning: Clustering

Two styles of clustering

1. Clustering into distinct components
   - 
2. Hierarchical clustering

Unsupervised learning: Clustering

Two styles of clustering

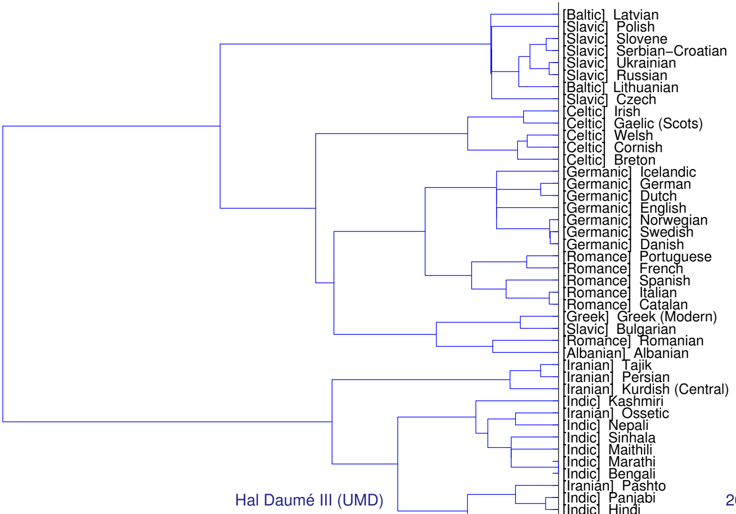
1. Clustering into distinct components
   - 
   - How many clusters are there?

2. Hierarchical clustering

Unsupervised learning: Clustering

Two styles of clustering

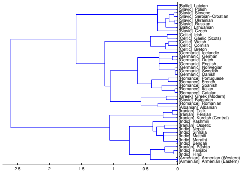
1. Clustering into distinct components
   - 
   - How many clusters are there?
   - What is important? Person? Expression? Lighting?
2. Hierarchical clustering


[Baltic] Latvian
[Slavic] Polish
[Slavic] Slovene
[Slavic] Serbian−Croatian
[Slavic] Ukrainian
[Slavic] Russian
[Baltic] Lithuanian
[Slavic] Czech
[Celtic] Irish
[Celtic] Gaelic (Scots)
[Celtic] Welsh
[Celtic] Cornish
[Celtic] Breton
[Germanic] Icelandic
[Germanic] German
[Germanic] Dutch
[Germanic] English
[Germanic] Norwegian
[Germanic] Swedish
[Germanic] Danish
[Romance] Portuguese
[Romance] French
[Romance] Spanish
[Romance] Italian
[Romance] Catalan
[Greek] Greek (Modern)
[Slavic] Bulgarian
[Romance] Romanian
[Albanian] Albanian
[Iranian] Tajik
[Iranian] Persian
[Iranian] Kurdish (Central)
[Indic] Kashmiri
[Iranian] Ossetic
[Indic] Nepali
[Indic] Sinhala
[Indic] Maithili
[Indic] Marathi
[Indic] Bengali
[Iranian] Pashto
[Indic] Panjabi
[Indic] Hindi

Unsupervised learning: Clustering

Two styles of clustering

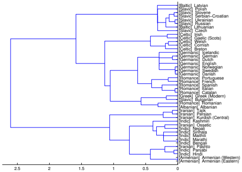
1. Clustering into distinct components
   - 
   - How many clusters are there?
   - What is important? Person? Expression? Lighting?
2. Hierarchical clustering



Unsupervised learning: Clustering

Two styles of clustering

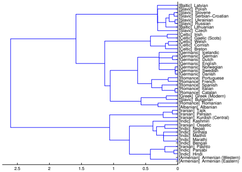
1. Clustering into distinct components
   - 
   - How many clusters are there?
   - What is important? Person? Expression? Lighting?
2. Hierarchical clustering
   - What is important?



Unsupervised learning: Clustering

Two styles of clustering

1. Clustering into distinct components
   - 
   - How many clusters are there?
   - What is important? Person? Expression? Lighting?
2. Hierarchical clustering
   - What is important?
   - How will we use this?



Unsupervised learning: Manifold learning

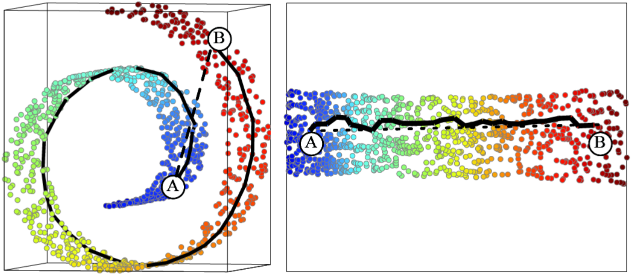
Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped.

Task is to recover the true *geometry* of the underlying data.

Unsupervised learning: Manifold learning

Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped.

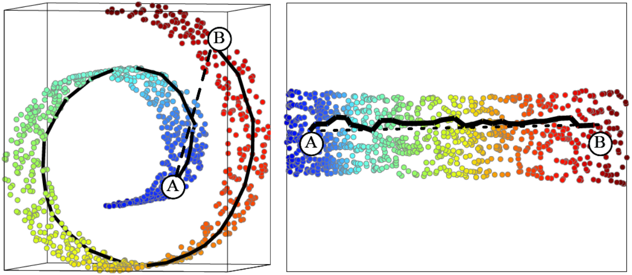
Task is to recover the true *geometry* of the underlying data.



Unsupervised learning: Manifold learning

Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped.

Task is to recover the true *geometry* of the underlying data.

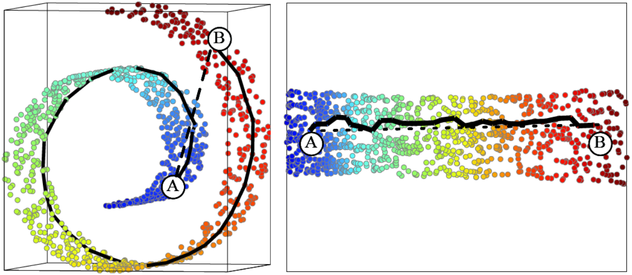


- Usually, replace "two" with "d" and "three" with "D" for $d \ll D$

Unsupervised learning: Manifold learning

Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped.

Task is to recover the true *geometry* of the underlying data.

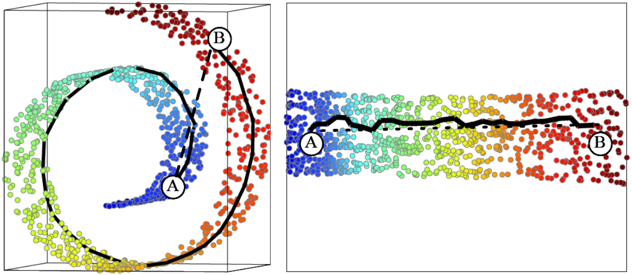


- Usually, replace "two" with "d" and "three" with "D" for $d \ll D$
- Useful for visualization (when $d \in \{2, 3\}$)

Unsupervised learning: Manifold learning

Often data that is really two dimensional is embedded in a higher dimensional space, sometimes warped.

Task is to recover the true *geometry* of the underlying data.



- Usually, replace “two” with “d” and “three” with “D” for $d \ll D$
- Useful for visualization (when $d \in \{2, 3\}$)
- Also useful for finding good representations for input to classifiers

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

Example problems:

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

Example problems:

1. Chess

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

Example problems:

1. Chess
2. Robot control

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

Example problems:

1. Chess
2. Robot control
3. Taxi driving

Reinforcement learning

- Reinforcement learning is the penultimate ML problem
- It is “ML-hard”
- Unlike classification, regression and unsupervised learning, RL does not recieve examples
- Rather, it gathers experience by interacting with the world
- RL problems always include time as a variable

  Example problems:

  1. Chess
  2. Robot control
  3. Taxi driving
- Key trade-off is exploration versus exploitation

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits
- A set of actions that an agent can take in any state

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits
- A set of actions that an agent can take in any state
- The reward the agent gets for reaching some particular state

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits
- A set of actions that an agent can take in any state
- The reward the agent gets for reaching some particular state

The *goal* of the agent is to take a sequence of actions so as to maximize its sum of rewards.

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits
- A set of actions that an agent can take in any state
- The reward the agent gets for reaching some particular state

The *goal* of the agent is to take a sequence of actions so as to maximize its sum of rewards.

The agent's output is a *policy* that maps states to actions.

Reinforcement learning: general setting

A (simple) reinforcement learning problem is defined by:

- A state space that defines the *world* that our *agent* inhabits
- A set of actions that an agent can take in any state
- The reward the agent gets for reaching some particular state

The *goal* of the agent is to take a sequence of actions so as to maximize its sum of rewards.

The agent's output is a *policy* that maps states to actions.

If you want to learn about RL, take AI

Why do we care about math?!

- **Calculus and linear algebra:**
- **Probability:**
- **Statistics:**

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
- **Probability:**
- **Statistics:**

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
  2. Convenient language for high dimensional data analysis
- **Probability:**
- **Statistics:**

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
  2. Convenient language for high dimensional data analysis
- **Probability:**
  1. The study of the outcome of repeated experiments
- **Statistics:**

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
  2. Convenient language for high dimensional data analysis
- **Probability:**
  1. The study of the outcome of repeated experiments
  2. The study of the plausibility of some event
- **Statistics:**

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
  2. Convenient language for high dimensional data analysis
- **Probability:**
  1. The study of the outcome of repeated experiments
  2. The study of the plausibility of some event
- **Statistics:**
  1. The analysis and interpretation of data

Why do we care about math?!

- **Calculus and linear algebra:**
  1. Techniques for finding maxima/minima of functions
  2. Convenient language for high dimensional data analysis
- **Probability:**
  1. The study of the outcome of repeated experiments
  2. The study of the plausibility of some event
- **Statistics:**
  1. The analysis and interpretation of data

Statistics makes heavy use of probability theory.

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

How is machine learning *different* from statistics?

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

How is machine learning *different* from statistics?

1. Stats cares about the model, we care about predictions

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

How is machine learning *different* from statistics?

1. Stats cares about the model, we care about predictions
2. Stats cares about model fit, we care about generalization

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

How is machine learning *different* from statistics?

1. Stats cares about the model, we care about predictions
2. Stats cares about model fit, we care about generalization
3. Stats tries to explain the world, we try to predict the future

Why do we care about probability and statistics?

Recall, statistics is the analysis and interpretation of data.

In machine learning, we attempt to generalize from one “training” data set to general “rules” that can be applied to “test” data.

How is machine learning *different* from statistics?

1. Stats cares about the model, we care about predictions
2. Stats cares about model fit, we care about generalization
3. Stats tries to explain the world, we try to predict the future

It all started with a lady drinking tea. . .

History of ML?

- Initial attempts at object recognition [Rosenblatt, 1958]
- Learning to play checker [Samuel, 1959, 1963]
- Rosenblatt can't learn XOR [Minsky & Pappert, 1969]
- Symbolic learning, spectroscopy [Winston, 1975; Buchanan 1971]
- Backpropagation for neural nets [Werbos, 1974; Rummelhart, 1986]
- PAC model of learning theory [Valiant, 1984]
- Optimization enters machine learning [Bennett & Mangasarian, 1993]
- Kernel methods for non-linearity [Cortes & Vapnik, 1995]
- Machine learning behind day-to-day tasks [2005ish]
- Machine learning takes over the world [2010ish]