Machine Learning

Vaibhav Rastogi

With slides from Doug Downey

Machine Learning

- "The study of computer programs that improve automatically with experience"
  T. Mitchell *Machine Learning*, 1998
- Used heavily in:
  - Bioinformatics, robotics, marketing/advertising, recommendations systems, **information retrieval**, fraud detection, handwriting/speech recognition, etc., etc…

Example Machine Learning Tasks

- How likely is person **x** to default on a loan?
- What is the location of robot **x**?
- Is a given Web page **x** about “baseball?”

More to the point….

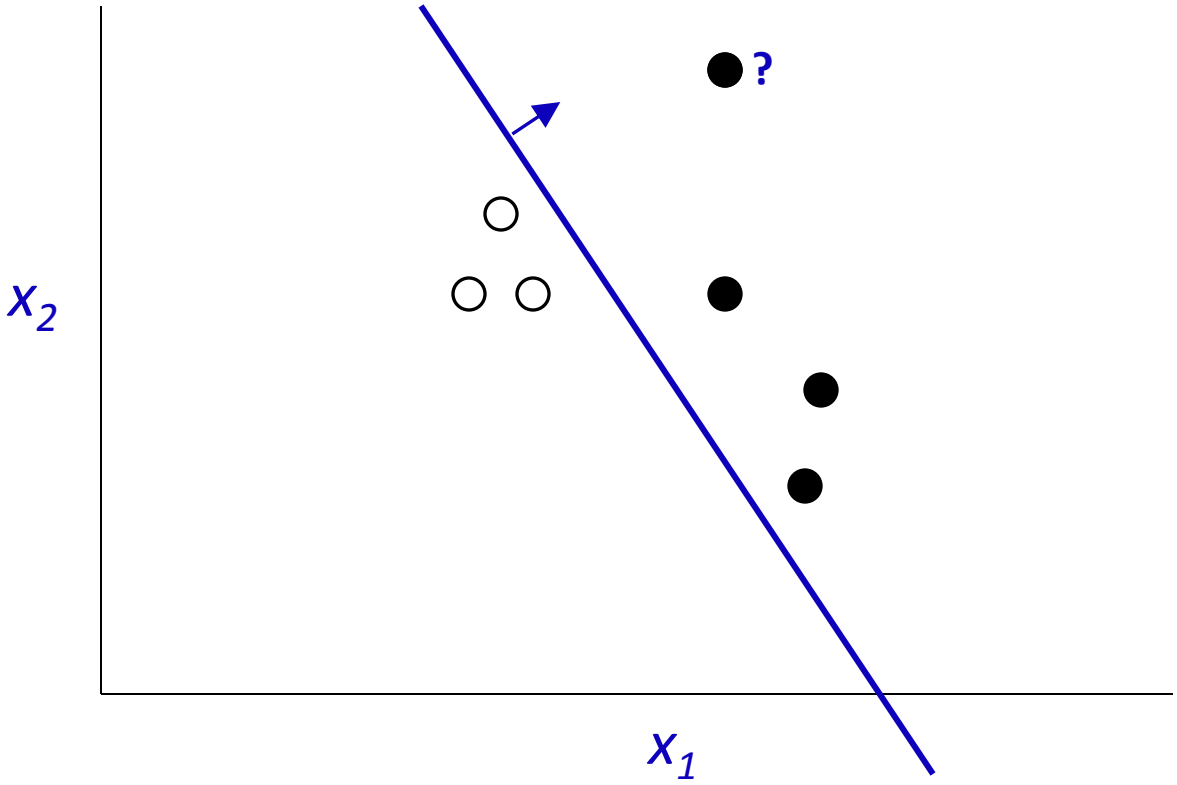
- Is document **x** about music or religion?
- Is blog post **x** positive or negative?
- What's the difficulty of the hike described on page **x**?
- What's the probability that a piece of text **x** occurs?
- Is page **x** referring to the same person as page **y**?
- Is the joke **x** funny?
- Is product **x** better than product **y**?

Learning a function from examples

- **Given:** examples of a function f for various inputs **x**:
  - $\{(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))\}$
- **Goal:** Estimate f
  - Input **x** = $(x_1, \ldots, x_d)$; individual features x_i
  - Output $f(\mathbf{x})$

- Probably the most common machine learning task formulation (though there are others)

Learn function from $\mathbf{x} = (x_1, \ldots, x_d)$ to $f(\mathbf{x}) \in \{0, 1\}$
given labeled examples $(\mathbf{x}, f(\mathbf{x}))$



How to do Machine Learning

1) Pick a feature representation for your task
2) Compile data
3) Choose a machine learning algorithm
4) Train the algorithm
5) Evaluate the results
6) *Probably: go to (1)*

Representation

- In general, inputs and outputs can be
  - Nominal (e.g. Gender)
  - Ordinal (e.g. small, medium, large)
  - Numeric (e.g. Years of Education, probability of credit default, etc.)
- Predicting a nominal output: classification
  - Thus, predicting whether document is about politics or sports is an instance of **Text Classification**
- Predicting a numeric output: regression (typically continuous)

Feature Engineering

- The art of machine learning
  - Features should be predictive and (relatively) independent
- How likely is person **x** to default on a loan?
  - FICO score
  - Income
  - Education Level
  - Assets
  - ~~Social Security Number~~
  - …

How to do Machine Learning

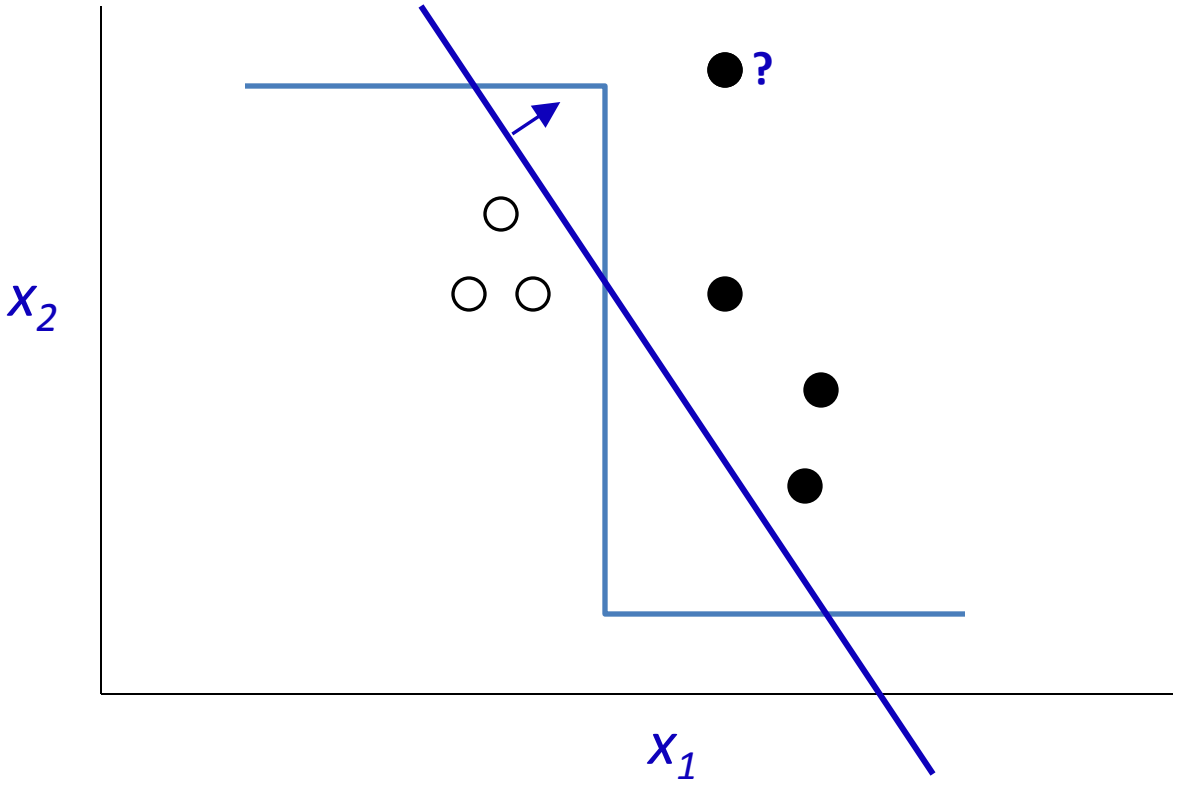
1) Pick a feature representation for your task
2) **Compile data**
3) Choose a machine learning algorithm
4) Train the algorithm
5) Evaluate the results
6) *Probably: go to (1)*

How to do Machine Learning

1) Pick a feature representation for your task
2) Compile data
3) **Choose a machine learning algorithm**
4) Train the algorithm
5) Evaluate the results
6) *Probably: go to (1)*

Which classifier is best?

Learn function from **x** = (x_1, …, x_d) to $y \in \{0, 1\}$
given labeled examples (**x**, y)



Which classifier is best?

- Answer: you don’t know
- Simplest workaround: try several and choose what works

- *In general: select classifier based on domain intuitions and problem structure*

How to do Machine Learning

1) Pick a feature representation for your task
2) Compile data
3) Choose a machine learning algorithm
4) **Train the algorithm**
5) Evaluate the results
6) *Probably: go to (1)*

Train the algorithm

- Use an ML package
  - E.g. Weka (link on course Web page)

How to do Machine Learning

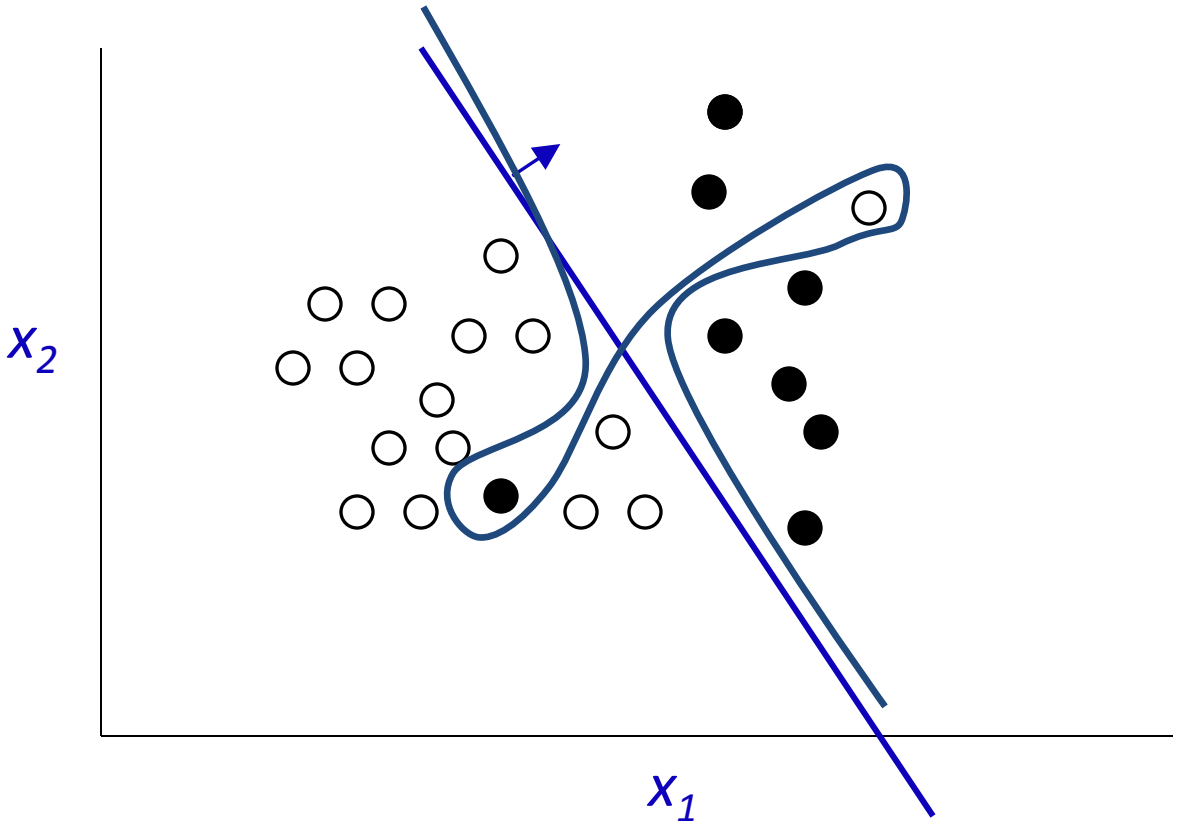
1) Pick a feature representation for your task
2) Compile data
3) Choose a machine learning algorithm
4) Train the algorithm
5) **Evaluate the results**
6) *Probably: go to (1)*

What does it mean for an ML algorithm to perform well?

- Metrics
  - Lots of possibilities
  - Classification: **accuracy**, precision, recall, cost, etc.
    - Accuracy = fraction of examples **x** where algorithm's predicted $f(\mathbf{x})$ matches true classification
  - Regression: mean squared error, etc.

What does it mean for an ML algorithm to perform well?

Learn function from $\mathbf{x} = (x_1, \ldots, x_d)$ to $f(\mathbf{x}) \in \{0, 1\}$
given labeled examples $(\mathbf{x}, f(\mathbf{x}))$



What does it mean for an ML algorithm to perform well?

- We want to know how our algorithm will perform on *new* inputs
  - So, test on a set of examples from disjoint from training (e.g. 80% train, 20% test)

- More general: cross-validation
  - Weka has this built-in
    - Along with significance testing

Machine Learning

- This was brief
- Take EECS 349 if interested