Today

- Administration details
- Why is machine learning so cool?
It is up to you to determine if you have the appropriate background

Tutorials:
- Tuesdays, 2-3, BA 1160

Do I have the appropriate background?
- **Linear algebra**: vector/matrix manipulations, properties
- **Calculus**: partial derivatives
- **Probability**: common distributions; Bayes Rule
- **Statistics**: mean/median/mode; maximum likelihood
- Sheldon Ross: A First Course in Probability

Webpage of the course:
[http://www.cs.toronto.edu/~zemel/Courses/CS2515](http://www.cs.toronto.edu/~zemel/Courses/CS2515)
Textbooks

- Christopher Bishop: "Pattern Recognition and Machine Learning", 2006
- Other Textbooks:
  - Kevin Murphy: "Machine Learning: a Probabilistic Perspective"
  - David Mackay: "Information Theory, Inference, and Learning Algorithms"
Requirements

- Do the readings!

- Assignments:
  - Two assignments, each worth 20%, for a total of 40%
  - Programming: take Matlab/Python code and extend it
  - Derivations: pen(cil)-and-paper

- Project:
  - Due Dec 16th
  - Worth 35% of course mark

- Test:
  - In first hour of last class meeting
  - Worth 25% of course mark
More on Assignments

- **Collaboration** on the assignments is not allowed. Each student is responsible for his/her own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of pseudocode or code or simulation results. Violation of this policy is grounds for a semester grade of F, in accordance with university regulations.

- The **schedule** of assignments is included in the syllabus. Assignments are due at the beginning of class/tutorial on the due date.

- Assignments handed in **late** but before 5 pm of that day will be penalized by 5% (i.e., total points multiplied by 0.95); a late penalty of 10% per day will be assessed thereafter.

- **Extensions** will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the instructor at least one week before the due date.
Resources

- Course on Piazza at piazza.com/utoronto.ca/fall2015/csc2515/home
  - Register to have access at piazza.com/utoronto.ca/fall2015/csc2515
  - Communicate announcements
  - Forum for discussion between students
  - Q/A for instructors/TAs and students: We will monitor as much as possible

- Office hours:
  - Thursday 4-5 Pratt 290D

- Lecture notes, assignments, readings and some announcements will be available on the course webpage
CLASS SCHEDULE

Shown below are the topics for lectures and tutorials (in italics), as are the dates that each assignment will be handed out and is due. All of these are subject to change. The notes from each lecture and tutorial will be available on the class web-site the day of the class meeting.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 17</td>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>Sep 22</td>
<td>Probability for ML &amp; Linear regression</td>
<td></td>
</tr>
<tr>
<td>Sep 24</td>
<td>Basic Methods &amp; Concepts</td>
<td></td>
</tr>
<tr>
<td>Sep 29</td>
<td>Optimization for ML</td>
<td></td>
</tr>
<tr>
<td>Oct 1</td>
<td>Nonparametric methods</td>
<td>Asst 1 Out</td>
</tr>
<tr>
<td>Oct 6</td>
<td>kNN &amp; Decision trees</td>
<td></td>
</tr>
<tr>
<td>Oct 8</td>
<td>Probabilistic Classifiers</td>
<td></td>
</tr>
<tr>
<td>Oct 13</td>
<td>Naive Bayes &amp; Gaussian Bayes classifiers</td>
<td></td>
</tr>
<tr>
<td>Oct 15</td>
<td>Neural Networks</td>
<td></td>
</tr>
<tr>
<td>Oct 20</td>
<td>Deep learning</td>
<td>Asst 1 In</td>
</tr>
<tr>
<td>Oct 22</td>
<td>Clustering</td>
<td></td>
</tr>
<tr>
<td>Oct 27</td>
<td>Mixtures of Gaussians</td>
<td>Asst 2 Out</td>
</tr>
<tr>
<td>Oct 29</td>
<td>Continuous Latent Variable Models</td>
<td>Project Proposals In</td>
</tr>
<tr>
<td>Nov 3</td>
<td>PCA</td>
<td></td>
</tr>
<tr>
<td>Nov 5</td>
<td>Kernel Methods</td>
<td></td>
</tr>
<tr>
<td>Nov 10</td>
<td>SVMs</td>
<td>Asst 2 In</td>
</tr>
<tr>
<td>Nov 12</td>
<td>Structured Prediction Models</td>
<td></td>
</tr>
<tr>
<td>Nov 17</td>
<td>Structured SVMs</td>
<td></td>
</tr>
<tr>
<td>Nov 19</td>
<td>Ensemble Methods</td>
<td></td>
</tr>
<tr>
<td>Nov 24</td>
<td>Boosting &amp; Mixture of experts</td>
<td></td>
</tr>
<tr>
<td>Nov 26</td>
<td>Reinforcement Learning</td>
<td></td>
</tr>
<tr>
<td>Dec 1</td>
<td>Review for Test</td>
<td></td>
</tr>
<tr>
<td>Dec 3</td>
<td>Test; Speech Recognition</td>
<td></td>
</tr>
<tr>
<td>Dec 16</td>
<td>Projects In</td>
<td></td>
</tr>
</tbody>
</table>
What is Machine Learning?

- How can we solve a specific problem?
  - As computer scientists we write a program that encodes a set of rules that are useful to solve the problem
  - In many cases is very difficult to specify those rules, e.g., given a picture determine whether there is a cat in the image

- Learning systems are not directly programmed to solve a problem, instead develop own program based on:
  - Examples of how they should behave
  - From trial-and-error experience trying to solve the problem

- Different than standard CS:
  - Want to implement unknown function, only have access to sample input-output pairs (training examples)

- Learning simply means incorporating information from the training examples into the system
Task that requires machine learning: What makes a 2?
Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.
1. **Classification**: Determine which discrete category the example is
Examples of Classification

What digit is this?
Is this a dog?
what about this one?
Am I going to pass the exam?
Do I have diabetes?
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
Examples of Recognizing patterns
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc.
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
Examples of Recommendation systems

Despicable Me

Villainous Gru hatches a plan to steal the moon from the sky. But he has a tough time staying on task after three orphans land in his care.

Starring: Steve Carell, Jason Segel, Russell Brand
Genres: Children & Family Movies, Movies for ages 5 to 7, Movies for ages 8 to 10
This movie is: Goofy

Steve Carell, Jason Segel, Russell Brand and Kristen Wiig lend their voices to this animated box office hit.
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval**: Find documents or images with similar content
Examples of Information Retrieval

[PDF] CSC 411 MACHINE LEARNING and DATA MINING...
CSC 411. MACHINE LEARNING and DATA MINING. Lectures: Monday, Wednesday 12-1 (section 1), 3-4 (section 2). Lecture Room: MP 134 (section 1); Bahen ...

Professor Richard Zemel - Department of Computer Science
www.cs.toronto.edu/~zemel/
Course Offerings - Research Interests - Students & Post Docs - Contact Info

UofT Machine Learning | Course
learning.cs.toronto.edu/courses
CSC 411, Machine Learning and Data Mining (Raquel Urtasun and Richard Zemel); STA 4513, Statistical models of networks, graphs, and other relational ...

CSC 411: Machine Learning and Data Mining
www.cs.utoronto.ca/~radford/csc411_F06/
CSC 411: Machine Learning and Data Mining (Sept-Dec 2006). Note: The test on December 8 at 3pm will be held in BA B024, not the usual lecture/tutorial room.
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval**: Find documents or images with similar content
5. **Computer vision**: detection, segmentation, depth estimation, optical flow, etc
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval**: Find documents or images with similar content
5. **Computer vision**: detection, segmentation, depth estimation, optical flow, etc
6. **Robotics**: perception, planning, etc
Flying Robots
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval**: Find documents or images with similar content
5. **Computer vision**: detection, segmentation, depth estimation, optical flow, etc
6. **Robotics**: perception, planning, etc
7. **Learning to play games**
Playing Games: Super Mario
Learning algorithms are useful in other tasks

1. **Classification**: Determine which discrete category the example is
2. **Recognizing patterns**: Speech Recognition, facial identity, etc
3. **Recommender Systems**: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval**: Find documents or images with similar content
5. **Computer vision**: detection, segmentation, depth estimation, optical flow, etc
6. **Robotics**: perception, planning, etc
7. **Learning to play games**
8. **Recognizing anomalies**: Unusual sequences of credit card transactions, panic situation at an airport
9. **Spam filtering, fraud detection**: The enemy adapts so we must adapt too
10. **Many more!**
Can you pick out the tufas?
Types of learning task

- **Supervised**: correct output known for each training example
  - Learn to predict output when given an input vector
    - **Classification**: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - **Regression**: real-valued output (predicting market prices, customer rating)

- **Unsupervised learning**
  - Create an internal representation of the input, capturing regularities/structure in data
  - Examples: form clusters; extract features
    - How do we know if a representation is good?

- **Reinforcement learning**
  - Learn action to maximize payoff
    - Not much information in a payoff signal
    - Payoff is often delayed
Machine Learning vs Data Mining

- **Data-mining**: Typically using very simple machine learning techniques on very large databases because computers are too slow to do anything more interesting with ten billion examples.

- Previously used in a negative sense – misguided statistical procedure of looking for all kinds of relationships in the data until finally find one.

- Now lines are blurred: many ML problems involve tons of data.

- But problems with AI flavor (e.g., recognition, robot navigation) still domain of ML.
ML uses statistical theory to build models – core task is inference from a sample.

A lot of ML is rediscovery of things statisticians already knew; often disguised by differences in terminology.

But the emphasis is very different:

- **Good piece of statistics:** Clever proof that relatively simple estimation procedure is asymptotically unbiased.
- **Good piece of ML:** Demo that a complicated algorithm produces impressive results on a specific task.

Can view ML as applying computational techniques to statistical problems. But go beyond typical statistics problems, with different aims (speed vs. accuracy).
Cultural gap (Tibshirani)

MACHINE LEARNING
- weights
- learning
- generalization
- supervised learning
- unsupervised learning
- large grant: $1,000,000
- conference location: Snowbird, French Alps

STATISTICS
- parameters
- fitting
- test set performance
- regression/classification
- density estimation, clustering
- large grant: $50,000
- conference location: Las Vegas in August
Please complete the following survey this week:
https://docs.google.com/forms/d/1O6xRNnKp87GrDM74tkvOMhMIJmwz271TgWdYb6ZitK0/viewform?usp=send_form
Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from previous years
- Process the data
  - Split into training set; test set
  - Determine representation of input features; output
- Choose form of model: linear regression
- Decide how to evaluate the system’s performance: objective function
- Set model parameters to optimize performance
- Evaluate on test set: generalization
Outline

- **Linear regression problem**
  - continuous outputs
  - simple model

- **Introduce key concepts:**
  - loss functions
  - generalization
  - optimization
  - model complexity
  - regularization
Circles are data points (i.e., training examples) that are given to us.

The data points are uniform in $x$, but may be displaced in $y$

$$t(x) = f(x) + \epsilon$$

with $\epsilon$ some noise.

In green is the "true" curve that we don’t know.

Goal: We want to fit a curve to these points.
Simple 1-D regression

Key Questions:
- How do we parametrize the model?
- What loss (objective) function should we use to judge the fit?
- How do we optimize fit to unseen test data (generalization)?
Example: Boston Housing data

- Estimate median house price in a neighborhood based on neighborhood statistics
- Look at first (of 13) attributes: per capita crime rate

![Scatter plot showing relationship between per capita crime rate and median house price.](image)

- Use this to predict house prices in other neighborhoods
- Is this a good input (attribute) to predict house prices?
Represent the Data

- Data is described as pairs \( \mathcal{D} = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(N)}, t^{(N)})\} \)
  - \( x \) is the input feature (per capita crime rate)
  - \( t \) is the target output (median house price)
- Here \( t \) is continuous, so this is a regression problem
- Model outputs \( y \), an estimate of \( t \)
  \[
y(x) = w_0 + w_1 x
\]
- What type of model did we choose?
- Divide the dataset into training and testing examples
  - Use the training examples to construct hypothesis, or function approximator, that maps \( x \) to predicted \( y \)
  - Evaluate hypothesis on test set
A simple model typically does not exactly fit the data – lack of fit can be considered noise.

Sources of noise:

- Imprecision in data attributes (input noise)
- Errors in data targets (mis-labeling)
- Additional attributes not taken into account by data attributes, affect target values (latent variables)
- Model may be too simple to account for data targets
Least-squares Regression

- Define a model
  \[ y(x) = w_0 + w_1 x \]

- Standard loss/cost/objective function measures the squared error between \( y \) and the true value \( t \)
  \[ \ell(w) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2 \]

- The loss for the red hypothesis is the sum of the squared vertical errors.
- How do we obtain the weights \( w = (w_0, w_1) \)?
Optimizing the Objective

- One straightforward method: gradient descent
  - initialize $\mathbf{w}$ (e.g., randomly)
  - repeatedly update $\mathbf{w}$ based on the gradient
    \[
    \mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}}
    \]
  - $\lambda$ is the learning rate
  - For a single training case, this gives the LMS update rule:
    \[
    \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}
    \]
- Note: As error approaches zero, so does the update
Two ways to generalize this for all examples in training set:

1. **Batch updates**: sum or average updates across every example \( n \), then change the parameter values

\[
\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}
\]

2. **Stochastic/online updates**: update the parameters for each training case in turn, according to its own gradients

- Underlying assumption: sample is independent and identically distributed (i.i.d.)
Multi-dimensional Inputs

- One method of extending the model is to consider other input dimensions
  \[ y(x) = w_0 + w_1 x_1 + w_2 x_2 \]
- In the Boston housing example, we can look at the number of rooms

- We can use gradient descent to solve for each coefficient, or use linear algebra – solve system of equations
Imagine now we want to predict the median house price from these multi-dimensional observations.

Each house is a data point \( n \), with observations indexed by \( j \):

\[
x^{(n)} = \left( x_{1}^{(n)}, \cdots, x_{d}^{(n)} \right)
\]

We can incorporate the bias \( w_0 \) into \( w \), by using \( x_0 = 1 \), then

\[
y = w_0 + \sum_{j=1}^{d} w_j x_j = w^T x
\]

We can then solve for \( w = (w_0, w_1, \cdots, w_d) \). How?

What if our linear model is not good? How can we create a more complicated model?
Fitting a Polynomial

- We can create a more complicated model by defining input variables that are combinations of components of $x$
- Example: an $M$-th order polynomial function

$$y(x, w) = w_0 + \sum_{j=1}^{M} w_j x^j$$

where $x^j$ is the $j$-th power of $x$

- We can use the same approach to optimize the values of the weights on each coefficient
- How do we do that?
Which fit is best?

from Bishop Zemel & Urtasun (UofT)

CSC 2515: 01-Introduction
Sep 17, 2015 48 / 50
Increasing the input features this way can complicate the model considerably

Goal: select the appropriate model complexity automatically

Standard approach: regularization

\[ \tilde{\ell}(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2 + \alpha \mathbf{w}^T \mathbf{w} \]

The penalty on the squared weights is known as ridge regression in statistics

Leads to modified update rule

\[ \mathbf{w} \leftarrow \mathbf{w} + 2\lambda \left[ \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)} - \alpha \mathbf{w} \right] \]
1-D regression illustrates key concepts

- Data fits – is linear model best (model selection)?
  - Simple models may not capture all the important variations (signal) in the data: underfit
  - More complex models may overfit the training data (fit not only the signal but also the noise in the data), especially if not enough data to constrain model

- One method of assessing fit: test generalization = model’s ability to predict the held out data

- Optimization is essential: stochastic and batch iterative approaches; analytic when available