Course Information

- Website: [http://classes.cec.wustl.edu/~cse417t/](http://classes.cec.wustl.edu/~cse417t/)

- Piazza (signup with Wash U email)

- Gradescope (signup with M6Z8XD) and SVN

- Textbooks:
Course Information

- Grading:
  - Homework assignments (6-8): 50%
    - Mix of programming and pencil-and-paper problems
    - Worst (second-worst) scores discounted 60% (40%)
    - 5 total late days, no more than 2 usable on any one assignment
  - Collaboration:
    - Feel free to discuss homework with other students
  - **Must write your own solutions**
  - **Must cite all external sources (including other students)**

- Tests (2): 50%
  - **10/4/18**: 6:30 PM – 8:30 PM
  - **12/5/18**: 6:30 PM – 8:30 PM
  - Location TBD
<table>
<thead>
<tr>
<th>First Half of the Course: Foundations</th>
<th>Second Half of the Course: Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory, Proofs, Math, Probability, Boring Stuff, etc...</td>
<td>Random Forests! Support Vector Machines! Neural Networks! Yay!</td>
</tr>
</tbody>
</table>

Overview
## Tentative Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/28</td>
<td>Introduction</td>
</tr>
<tr>
<td>8/30</td>
<td>Generalization</td>
</tr>
<tr>
<td>9/4</td>
<td>Matlab tutorial</td>
</tr>
<tr>
<td>9/6</td>
<td>Hypothesis sets</td>
</tr>
<tr>
<td>9/11</td>
<td>Infinite-dimensional hypothesis sets</td>
</tr>
<tr>
<td>9/13</td>
<td>Bias-variance tradeoff</td>
</tr>
<tr>
<td>9/18</td>
<td>Linear regression</td>
</tr>
<tr>
<td>9/20</td>
<td>Logistic regression</td>
</tr>
<tr>
<td>9/25</td>
<td>Overfitting</td>
</tr>
<tr>
<td>9/27</td>
<td>Regularization</td>
</tr>
<tr>
<td>10/2</td>
<td>Exam review</td>
</tr>
</tbody>
</table>
Machine Learning (Then)
Machine Learning (Now)
The pattern connects what you’re interested in learning about, what you’re interested in predicting to what you know

- There exists a pattern
- The pattern is difficult/impossible to describe
- There is data
- Use data to “learn” the pattern
Let’s say you’re a bank and people apply to you for credit cards or lines of credit; how do you decide who to extend credit to and who to deny?
Formal Setup

Unknown target function
\[ f : X \rightarrow Y \]

Training data
\[ D = \{ (x_1, y_1), \ldots, (x_m, y_m) \} \]

Learning Algorithm \( \mathcal{A} \)

Hypothesis Set \( \mathcal{H} \)

Learned Hypothesis
\[ \mathcal{H} \ni g : X \rightarrow Y \]
Formal Setup

Unknown target function
\[ f : X \rightarrow Y \]

Training data
\[ D = \{(x_i, y_i), ..., (x_n, y_n)\} \]

Learning Algorithm \( \mathcal{A} \)

Hypothesis Set \( \mathcal{H} \)

Learned Hypothesis
\[ \mathcal{H} \ni g \approx f \]
Learning Model

Unknown target function
$f: X \to Y$

Training data
$D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

Learning Algorithm $A$

Hypothesis Set $\mathcal{H}$

Learned Hypothesis
$\mathcal{H} \ni g \approx f$
Example: Inputs, Outputs and Data

- Assumptions:
  - Two continuous inputs - credit score and credit line size
  - One binary output - approve or deny
  - Dataset of \( n \) historical observations

- Formally,
  - Input space - \( X = \mathbb{R}^2 \)
  - Output space - \( Y = \{-1 \text{ (deny), } +1 \text{ (approve)}\} \)
  - Dataset - \( D = \{(x_{11}, x_{12}, y_1), \ldots, (x_{n1}, x_{n2}, y_n)\} = (X, \hat{y}) \)

  where \( X = \begin{bmatrix} x_{11} & x_{12} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix} \in \mathbb{R}^{n \times 2} \) and \( \hat{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n \)
Example: Hypothesis Set

- Perceptron
  - Given some input $\mathbf{x} = (x_1, x_2)$:

$$h(\mathbf{x}) = \begin{cases} +1 & \text{if } \sum_{i=1}^{2} w_i x_i > b \\ -1 & \text{otherwise} \end{cases}$$
Example: Hypothesis Set

- Perceptron
- Given some input $\mathbf{x} = (x_1, x_2)$:

$$h(\mathbf{x}) = \begin{cases} 
  +1 & \text{if } \sum_{i=1}^{2} w_i x_i - b > 0 \\
  -1 & \text{otherwise}
\end{cases}$$
Example: Hypothesis Set

• Perceptron
  • Given some input \( \bar{x} = (x_1, x_2) \):

\[
h(\bar{x}) = \text{sign} \left( \sum_{i=1}^{2} w_i x_i - b \right)
\]
Example: Hypothesis Set

- Perceptron
  - Given some input $\tilde{x} = (x_0 = 1, x_1, x_2)$:

$$h(\tilde{x}) = \text{sign} \left( \sum_{i=0}^{2} w_i x_i \right) = \text{sign}(\tilde{w}^T \tilde{x})$$
Example: Data
Example: Hypothesis

\[ w_0 = -43,000 \]
\[ w_1 = 60 \]
\[ w_2 = 1 \]
Example: Hypothesis

\[ w_0 = -650 \]
\[ w_1 = 1 \]
\[ w_2 = 0 \]
Example: Hypothesis

\[ w_0 = -21200 \]
\[ w_1 = 39 \]
\[ w_2 = -0.5 \]
Example: Learning Algorithm

- Perceptron Learning Algorithm (PLA) finds a linear separator in finite time, if the training data is linearly separable.

- Input: training data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
- Initialize $\mathbf{w}$ to all zeros or (small) random numbers
- While $\exists$ some misclassified training example i.e. $(\mathbf{x}_i, y_i) \in D$ s.t. $h(\mathbf{x}_i) = \text{sign}(\mathbf{w}^T \mathbf{x}_i) \neq y_i$
  - Randomly pick a misclassified training example, $(\mathbf{x}, y)$
  - Update $\mathbf{w}$: $\mathbf{w} = \mathbf{w} + y \mathbf{x}$
- Output: weights $\mathbf{w}$
Perceptron Learning Algorithm (Intuition)

• Suppose \((\tilde{x}, y) \in \mathcal{D}\) is a misclassified training example and \(y = +1\)
  - \(\tilde{w}^T \tilde{x}\) is negative
  - After updating \(\tilde{w}' = \tilde{w} + y\tilde{x}\)
    - \(\tilde{w}'^T \tilde{x} = (\tilde{w} + y\tilde{x})^T \tilde{x} = \tilde{w}^T \tilde{x} + y\tilde{x}^T \tilde{x}\) is less negative than \(\tilde{w}^T \tilde{x}\) because \(y > 0\) and \(\tilde{x}^T \tilde{x} > 0\)

• A similar argument holds if \(y = -1\)
Example: PLA

$\vec{w} = (-4.3, 0.6, 1)$
Example: PLA

\[ \mathbf{w} = (-4.3, 0.6, 1) \]
\[ \mathbf{x} = (1, 6.2, 1.5) \]
\[ y = -1 \]

\[ \mathbf{w} + y\mathbf{x} = (-5.3, -5.6, -0.5) \]
Types of Learning

• Supervised Learning
  • Training data is (input, output)
  • Examples: linear/logistic regression, support vector machines, neural networks
  • Variants: active learning and online learning

• Unsupervised Learning
  • Training data is (input)
Types of Learning
Types of Learning

- **Supervised Learning**
  - Training data is (input, output)
  - Examples: linear/logistic regression, support vector machines, neural networks
  - Variants: active learning and online learning

- **Unsupervised Learning**
  - Training data is (input)
  - Examples: clustering, principal component analysis, outlier detection

- **Reinforcement Learning**
  - Training data is (input, action, score)
  - Examples: Q-learning, temporal difference learning
Types of Learning

Source: https://www.sked.com/322/
Types of Learning

- Supervised Learning (this class!)
  - Training data is (input, output)
  - Examples: linear/logistic regression, support vector machines, neural networks
  - Variants: active learning and online learning

- Unsupervised Learning (CSE 517A)
  - Training data is (input)
  - Examples: clustering, principal component analysis, outlier detection

- Reinforcement Learning (CSE 511A)
  - Training data is (input, action, score)
  - Examples: Q-learning, temporal difference learning