Who are you?
Syllabus

- [www.cs.lafayette.edu/~taylorm/cs414](http://www.cs.lafayette.edu/~taylorm/cs414)

- What are the goals of this class?
Sample Learning Problem

- Learn to play checkers from self-play
- We will develop an approach analogous to that used in the first machine learning system developed by Arthur Samuels at IBM in 1959.
Training Experience

• **Direct experience**: Given sample input and output pairs for a useful target function.
  – Checker boards labeled with the correct move, e.g. extracted from record of expert play

• **Indirect experience**: Given feedback which is *not* direct I/O pairs for a useful target function.
  – Potentially arbitrary sequences of game moves and their final game results.

• **Credit/Blame Assignment Problem**: How to assign credit blame to individual moves given only indirect feedback?
Source of Training Data

- Provided random examples outside of the learner’s control.
  - Negative examples available or only positive?
- Good training examples selected by a “benevolent teacher.”
  - “Near miss” examples
- Learner can query an oracle about class of an unlabeled example in the environment.
- Learner can construct an arbitrary example and query an oracle for its label.
- Learner can design and run experiments directly in the environment without any human guidance.
Training vs. Test Distribution

• Generally assume that the training and test examples are independently drawn from the same overall distribution of data.
  – IID: Independently and identically distributed

• If examples are not independent, requires *collective classification*.

• If test distribution is different, requires *transfer learning*.
Choosing a Target Function

- What function is to be learned and how will it be used by the performance system?
- For checkers, assume we are given a function for generating the legal moves for a given board position and want to decide the best move.
  - Could learn a function:
    \[ \text{ChooseMove}(\text{board, legal-moves}) \rightarrow \text{best-move} \]
  - Or could learn an **evaluation function**, \( V(\text{board}) \rightarrow R \), that gives each board position a score for how favorable it is. \( V \) can be used to pick a move by applying each legal move, scoring the resulting board position, and choosing the move that results in the highest scoring board position.
Ideal Definition of $V(b)$

- If $b$ is a final winning board, then $V(b) = 100$
- If $b$ is a final losing board, then $V(b) = -100$
- If $b$ is a final draw board, then $V(b) = 0$
- Otherwise, then $V(b) = V(b')$, where $b'$ is the highest scoring final board position that is achieved starting from $b$ and playing optimally until the end of the game (assuming the opponent plays optimally as well).
  
  - Can be computed using complete mini-max search of the finite game tree.
Approximating $V(b)$

- Computing $V(b)$ is intractable since it involves searching the complete exponential game tree.
- Therefore, this definition is said to be non-operational.
- An operational definition can be computed in reasonable (polynomial) time.
- Need to learn an operational approximation to the ideal evaluation function.
Representing the Target Function

• Target function can be represented in many ways: lookup table, symbolic rules, numerical function, neural network.

• There is a trade-off between the expressiveness of a representation and the ease of learning.

• The more expressive a representation, the better it will be at approximating an arbitrary function; however, the more examples will be needed to learn an accurate function.
Linear Function for Representing $V(b)$

- In checkers, use a linear approximation of the evaluation function.

$$\tilde{V}(b) = w_0 + w_1 \cdot bp(b) + w_2 \cdot rp(b) + w_3 \cdot bk(b) + w_4 \cdot rk(b) + w_5 \cdot bt(b) + w_6 \cdot rt(b)$$

- $bp(b)$: number of black pieces on board $b$
- $rp(b)$: number of red pieces on board $b$
- $bk(b)$: number of black kings on board $b$
- $rk(b)$: number of red kings on board $b$
- $bt(b)$: number of black pieces threatened (i.e. which can be immediately taken by red on its next turn)
- $rt(b)$: number of red pieces threatened
Obtaining Training Values

• Direct supervision may be available for the target function.
  – $\langle bp=3, rp=0, bk=1, rk=0, bt=0, rt=0 \rangle$, 100>(win for black)

• With indirect feedback, training values can be estimated using *temporal difference learning* (used in *reinforcement learning* where supervision is *delayed reward*).