* Introduction
  + A bit about myself
  + A bit about everyone / expectations
  + Syllabus and organization
* How do we learn?
  + By being told
  + By analogy
  + By induction
* We will focus on Learning by Induction:
  + **Induction is a process that "involves intellectual leaps from the particular to the general"**
* Consider the following examples of induction:
  + Look at these two pictures
    - Pic 1: **Group** these objects in a meaningful way
    - Pic 2: Replace these objects by **rules** that represents them
  + Do not share your answers with anyone yet
* What is required for Induction?
  + A language, *Lp*, to represent the **particular** (i.e., specific instances)
  + A language, *Lg*, to represent the **general** (i.e., generalizations)
  + A matching predicate, *match*(*G,i)*, that is true if *G* is correct about *i*
  + A set, *I*, of particulars of some unknown general *G\**
* What does Inductive Learning do?
  + Observes the particulars in *I*
  + Produces a generalization, *G*, such that:
    - *G* is **consistent**:
      * For all *i* ∈ *I*, *match(G,i)*
    - *G* **generalizes** beyond *I*:
      * For many i ∉ *I*, *match(G,i)*
* Two different contexts
  + Pic 1 is an example of **Unsupervised Learning**
    - There is no predefined label for the instances
  + Pic 2 is an example of **Supervised Learning**
    - There is a predefined label for the instances
* What does this mean in terms of our definition?
  + Unsupervised learning
    - Consistency is (vacuously) guaranteed since you pick *G* and then assign instances to groups based on *G*
    - Generalization accuracy is also guaranteed, for the same reason
    - Need mechanism to choose among possible groupings (i.e., what makes your grouping better than mine?)
      * External metrics: rely on some subset of labeled data (generalization can now be measured)
      * Internal metrics: objective and subjective quality measures over groupings
  + Supervised learning
    - Must try to achieve consistency
      * Why is that desirable?
      * What may make it difficult/impossible?
        + Noise (i.e., mislabeled instances)
        + Limitation of *Lg* (i.e., cannot represent a consistent *G*)
    - Generalization accuracy can be measured directly
    - Need mechanism to choose among possible generalizations (i.e., what makes one generalization “better” than another?)
* Returning to our examples:
  + What features can/did you use to build your groupings and generalizations?
    - Color
    - Shape
    - Size
    - Other?
  + Example 1
    - What groupings did you come up with?
    - How good is your grouping?
  + Example 2
    - What generalizations did you come up with?
    - Is it consistent?
    - Why did you choose the one you did?
    - There are obviously several that are consistent:
      * If red or green then class 1 otherwise class 2
      * If less than 3 edges then class 1 otherwise class 2
    - How well do you think your generalization will do beyond the observed instances?
    - Suppose that I now tell you that class 2 is the set of complex polygons (i.e., objects with 4 sides or more). Would that change your preference? Why?
* Taking this last idea a bit further
  + Is it always possible to achieve consistency?
    - No. Recall our discussion above:
      * Data could be noisy
      * Language may not allow it
  + What can/should you do in the presence of noise?
    - Relax the consistency requirement
    - May actually prove useful in other situations (see below)
  + Assume no noise
    - What is an easy way to achieve consistency?
      * Memorize *I*
    - How do you then generalize to other instances?
      * Guess (no other choice)
    - How useful is that?
      * Rote learning (or no “real” learning at all)
    - In CS/programming terms, what does it mean to memorize?
      * *G*=*I*
      * In other words, there is no limitation to the language: it can represent any set of instances
    - Can you ever meaningfully generalize then if you insist on such level of consistency?
      * No! (guessing is not “meaningful” generalization)
    - How do we get out of this and effectively learn?
      * Introduce some kind of bias
      * **A bias is any basis for choosing one generalization over another, other than strict consistency with the observed training instances**
    - What types of bias can we use?
      * Language (i.e., can’t do *G*=*I*)
        + Example: Boolean conjunctions (0,1,\*)
      * Search procedure, such as:
        + Trade consistency for generalization
        + Favor simplicity and generality, specialize only when needed (e.g., “French people are obnoxious,” “Birds fly”)
        + Use domain knowledge (e.g., “double bonds rarely break,” “many infections induce fever”)
        + Consider intended use (e.g., differential cost of alert/no alert in ICU monitoring)
        + Etc.
* Another way to look at bias:
  + Weak/no bias
    - Observed instances are memorized
    - Learner **overfits**
  + Strong/extreme bias
    - Observed instances are mostly missed
    - Learner **underfits**
  + Trade off between bias and variance
    - If there is **no bias**, the outcome of the learner is highly dependent on the training data, and thus there is **much variance** among the models induced from different sets of observations
    - If there is a **strong bias**, the outcome of the learner is much less dependent on the training data, and thus there is **little variance** among induced models
  + This idea has been formalized in what has become known as the **bias-variance decomposition** of error
  + Example: fitting an arbitrary polynomial vs fitting a straight line to sine wave-like observations
* Our first lesson
  + The power of a learning system follows directly from its biases
    - Absence of bias = rote learning
    - Too much bias = result has little to do with the observations
  + **Make biases and their use as explicit as observations and their use**
  + Most of the supervised learning algorithms we will study have some language bias, as well as some procedural bias. It will be important to understand for each one of them what their search space is (i.e., language bias) and what their search bias is (i.e., procedural bias)