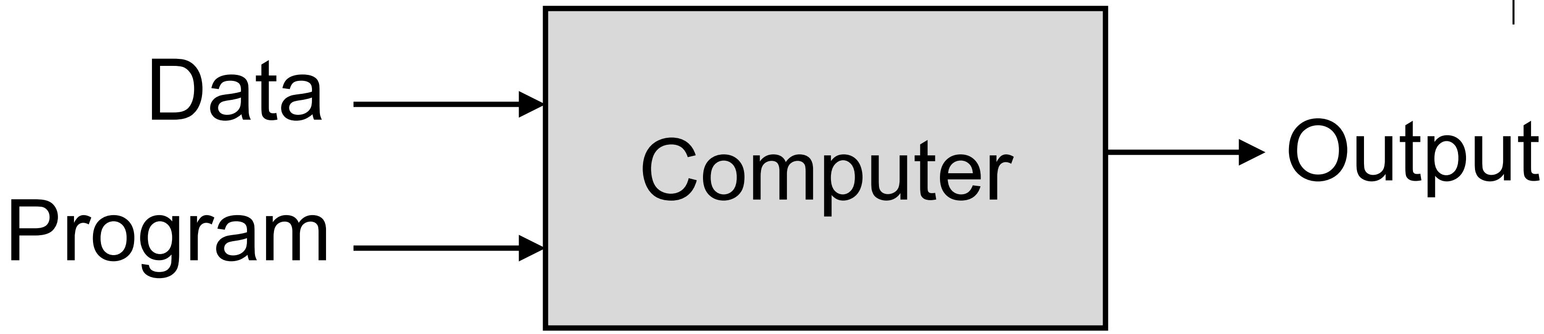
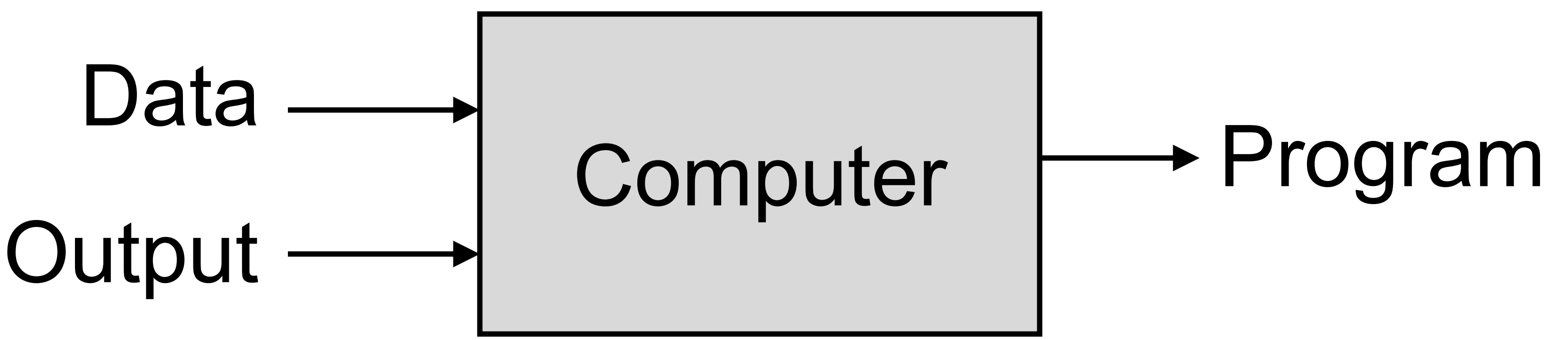
Twelve Key Ideas
In Machine Learning

Slides courtesy of Pedro Domingos

Traditional Programming



Machine Learning



Example: Classification

- **Classifier**
  - **Input:** Vector of discrete/numeric values (features)
  - **Output:** Class
  - **Example:** Spam filter
- **Learner**
  - **Input:** Training set of *(input, output)* examples
  - **Output:** Classifier
  - **Test:** Predictions on new examples

1. Learning = Representation + Evaluation + Optimization

- Thousands of learning algorithms
- Combinations of just three elements

Representation	Evaluation	Optimization
Instances	Accuracy	Greedy search
Hyperplanes	Precision/Recall	Branch & bound
Decision trees	Squared error	Gradient descent
Sets of rules	Likelihood	Quasi-Newton
Neural networks	Posterior prob.	Linear progr.
Graphical models	Margin	Quadratic progr.
Etc.	Etc.	Etc.

2. It's Generalization that Counts

- Test examples never seen before
- Training examples can just be memorized
- Set data aside to test
- Don't tune parameters on test data
- Use cross-validation
- No access to optimization goal
- Local optimum may be fine

3. Data Alone Is Not Enough

- Classes of unseen examples are arbitrary
- So learner must make assumptions
- “No free lunch” theorems
- Luckily, real world is not random
- Induction is knowledge lever

4. Overfitting Has Many Faces

- Overfitting = Hallucinating patterns
  = Chosen classifier not best on test
- The biggest problem in machine learning
- Bias and variance
- Less powerful learners can be better
- Solutions
  - Cross-validation
  - Regularization

5. Intuition Fails In High Dimensions

- Curse of dimensionality
- Sparseness worsens exponentially with number of features
- Irrelevant features ruin similarity
- In high dimensions all examples look alike
- 3D intuitions do not apply in high dimensions
- Blessing of non-uniformity

6. Theoretical Guarantees Are Not What They Seem

- Bounds on number of examples needed to ensure good generalization
- Extremely loose
- Low training error ≠> Low test error
- Asymptotic guarantees may be misleading
- Theory is useful for algorithm design, not evaluation

7. Feature Engineering Is the Key

- Most effort in ML projects is constructing features
- Black art: Intuition, creativity required
- ML is iterative process

8. More Data Beats A Cleverer Algorithm

- Easiest way to improve: More data
- Then: Data is bottleneck
- Now: Scalability is bottleneck
- ML algorithms more similar than they appear
- Clever algorithms require more effort but can pay off in the end
- Biggest bottleneck is human time

9. Learn Many Models, Not Just One

- Three stages of machine learning
  1. Try variations of one algorithm, chose one
  2. Try variations of many algorithms, choose one
  3. Combine many algorithms, variations
- Ensemble techniques
  - Bagging
  - Boosting
  - Stacking
  - Etc.

10. Simplicity Does Not Imply Accuracy

- Occam’s razor
- Common misconception:
  Simpler classifiers are more accurate
- Contradicts “no free lunch” theorems
- Counterexamples: ensembles, SVMs, etc.
- Can make preferred hypotheses shorter

11. Representable Does Not Imply Learnable

- Standard claim: “My language can represent/approximate any function”
- No excuse for ignoring others
- Causes of non-learnability
  - Not enough data
  - Not enough components
  - Not enough search
- Some representations exponentially more compact than others

12. Correlation Does Not Imply Causation

- Predictive models are guides to action
- Often interpreted causally
- Observational vs. experimental data
- Correlation → Further investigation

To Learn More

- **Article:**
  P. Domingos, "A Few Useful Things to Know About Machine Learning," *Communications of the ACM*, October 2012
  (Free version on my Web page)