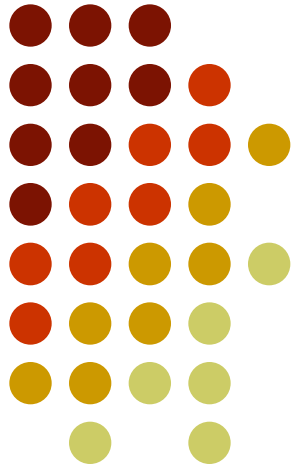
Supervised Learning (Big Data Analytics)

Vibhav Gogate

Department of Computer Science

The University of Texas at Dallas

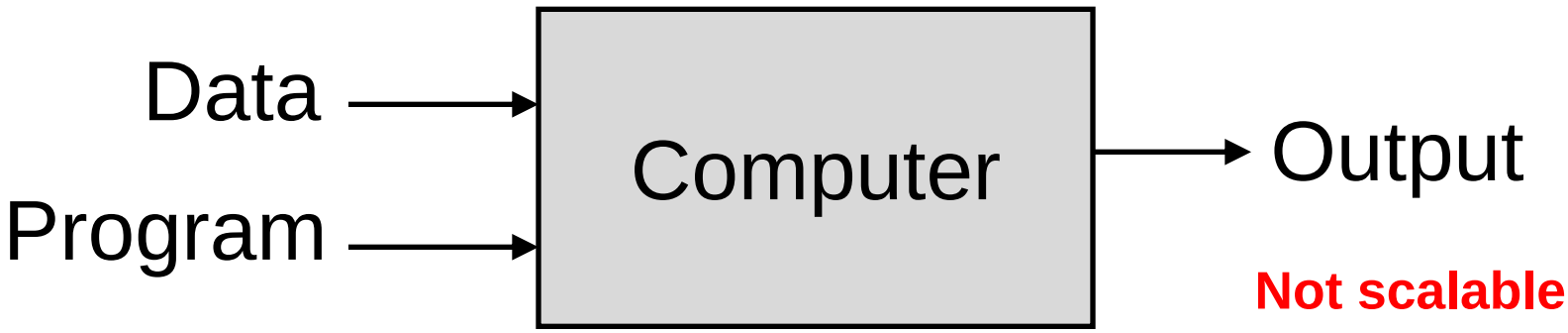


Practical advice

Goal of Big Data Analytics

- Uncover patterns in Data. Can be used to:
  - Gaining competitive advantage if you are a marketer
  - Making a lot of money if you are working in the stock market
  - Winning presidential Elections and so on.
- Analytics = Machine learning
- Practical advice on “how to use machine learning the right way.”

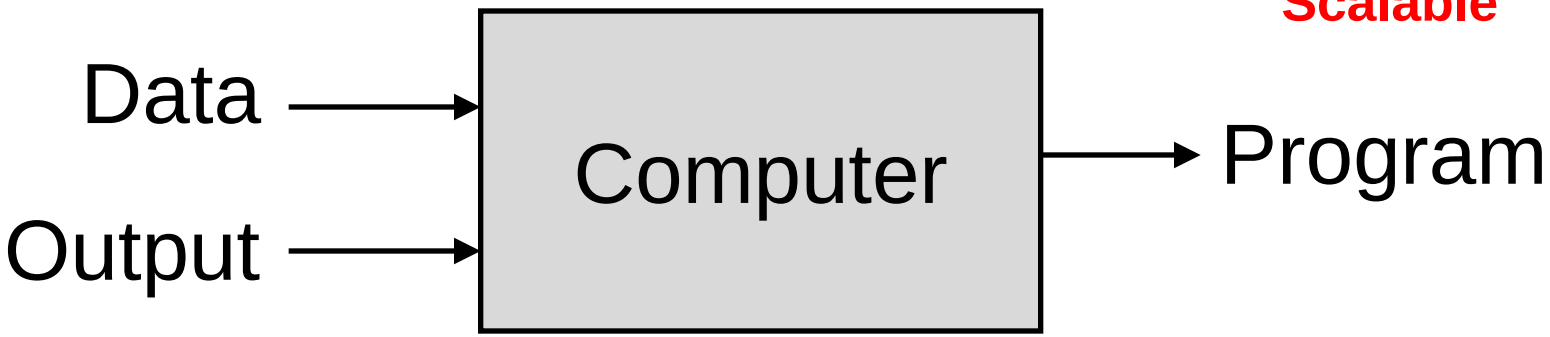
Traditional Programming



Not scalable

Machine Learning (Analytics)

Scalable



Not Magic: More like Gardening. Farmers combine seeds with nutrients to grow crops. Learners combine knowledge with data to grow programs

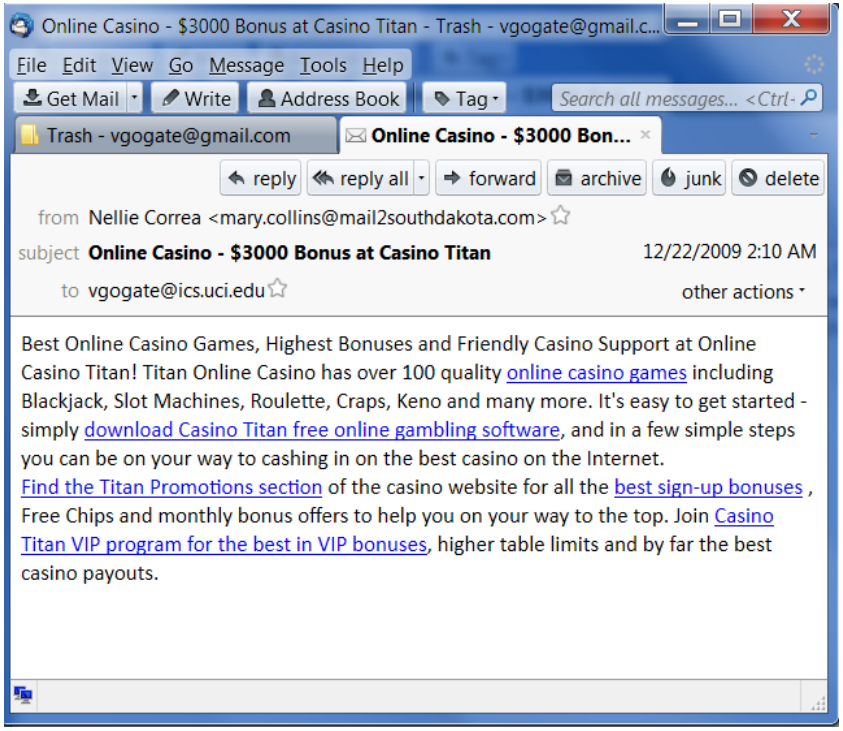
Supervised Learning

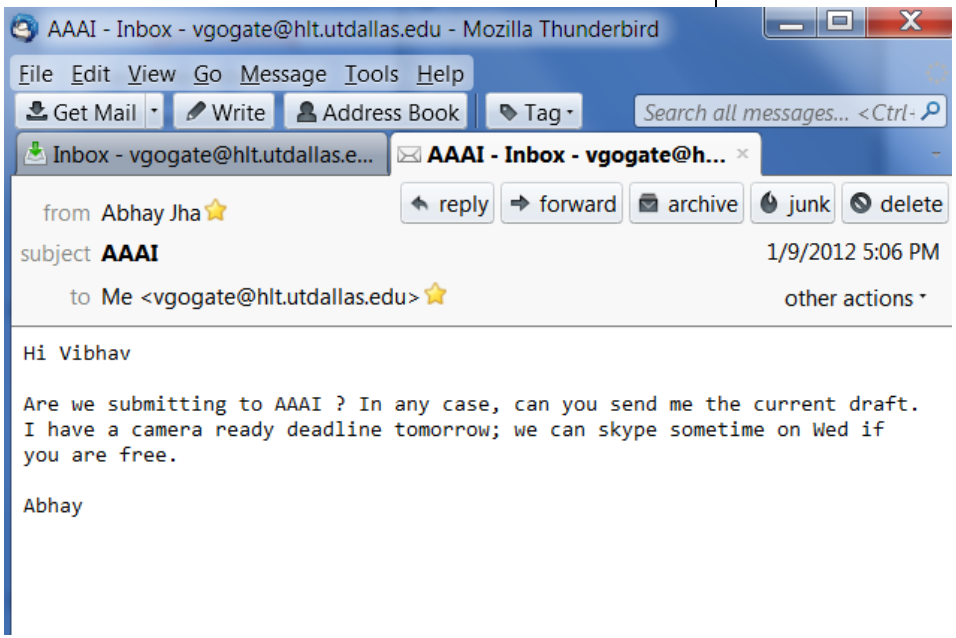
- **Given:** Training examples $(\boldsymbol{x}, f(\boldsymbol{x}))$ for some unknown function f.
- **Find:** A good approximation to f.
  - Classification problem: $f(\boldsymbol{x})$ is an (small) integer
  - Regression problem: $f(\boldsymbol{x})$ is a real number
- **Example Applications:** Credit Card approval; Spam Filtering; Disease Diagnosis; Automatically tagging images with location; etc.

Example: Credit Card approval

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant:
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to classify applications into two categories, approved and not approved.

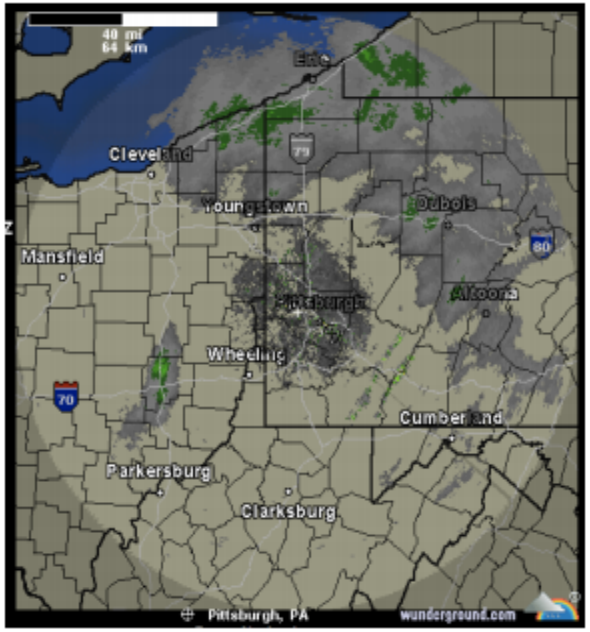
Classification Example: Spam Filtering





Classify as “Spam” or “Not Spam”

Classification Example: Weather Prediction



Classify as “Rainy”, “Cloudy”, “Sunny”

Regression example: Predicting Gold/Stock prices

Good ML can make you rich (but there is still some risk involved).

Given historical data on Gold prices, predict tomorrow's price!

Supervised Learning: Some Terminology

- Training Data
- Training Example: Example of the form $\left(\boldsymbol{x}, f(\boldsymbol{x})\right)$
- Classifier: A discrete-valued function or an algorithm that outputs a discrete-valued function
- Classes: The number of distinct values that $f(\boldsymbol{x})$ can take.

Training Data

Training Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Two Classes: {Yes,No}

Steps in Supervised Learning

1. Determine the representation for “x,f(x)” and determine what “x” to use

   Feature Engineering

2. Gather a training set (not all data is kosher)

   Data Cleaning

3. Select a suitable evaluation method
4. Find a suitable learning algorithm among a plethora of available choices

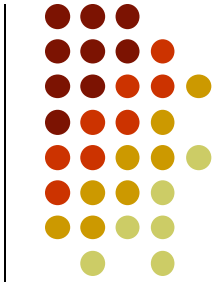
Feature Engineering is the Key

- Most effort in ML projects is constructing features
- Black art: Intuition, creativity required
  - Understand properties of the task at hand
  - How the features interact with or limit the algorithm you are using.
- ML is an iterative process
  - Try different types of features, experiment with each and then decide which feature set/algorithm combination to use

What features will you use?

- Examples
  - Spam Filtering
  - Mapping images to names
- Feature Combination
  - Linear models cannot handle some dependencies between features (e.g. XOR)
  - Feature combinations might work better.
  - Quick growth of the number of features.

Evaluation



- Accuracy
  - Fraction of the examples that are correctly classified by the learner
- Precision, Recall and F-score (Next slide)
- Squared error (Regression problems)
- Likelihood
- Posterior probability
- Cost / Utility
- Etc.

Precision, Recall and F-1 score

	Actual=True	Actual=False
Predicted=True	***tp*** (correct result)	***fp*** (unexpected result)
Predicted=False	***fn*** (missing result)	***tn*** (correct absence of result)

- Precision (P) = $\frac{tp}{tp+fp}$; Recall (R)= $\frac{tp}{tp+fn}$
- F1-score = $2\frac{P\times R}{P+R}$
  - Harmonic mean of precision and recall

What algorithms (Classifiers/learners) to use

- Naïve Bayes
- Logistic Regression
- Linear SVMs
- Decision Trees
- Neural Networks
- Support Vector Machines
- K-nearest neighbors (Non-parametric)
- Bagging (Meta); Boosting (Meta)

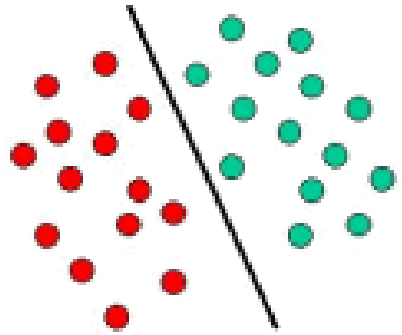
Linear classifiers

Non-linear

Weka Software

Classifiers: Bias versus Variance

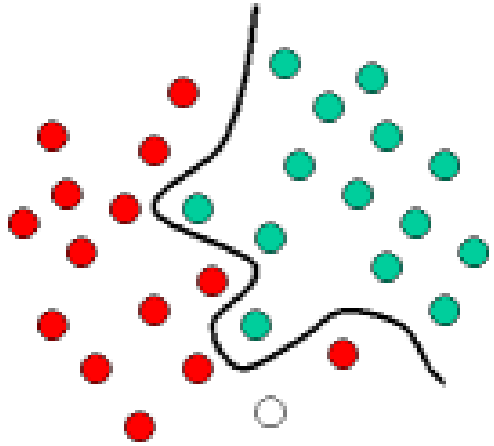
High Bias, low variance



Linear Classifier
- Naïve Bayes
- Logistic Regression
- Perceptron

Simpler

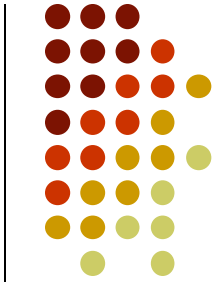
Low Bias, High variance



Non-Linear Classifier
- Support vector machine (Kernels)
- Neural networks
- Decision Trees

Complex

Learning = Representation + Evaluation + Optimization

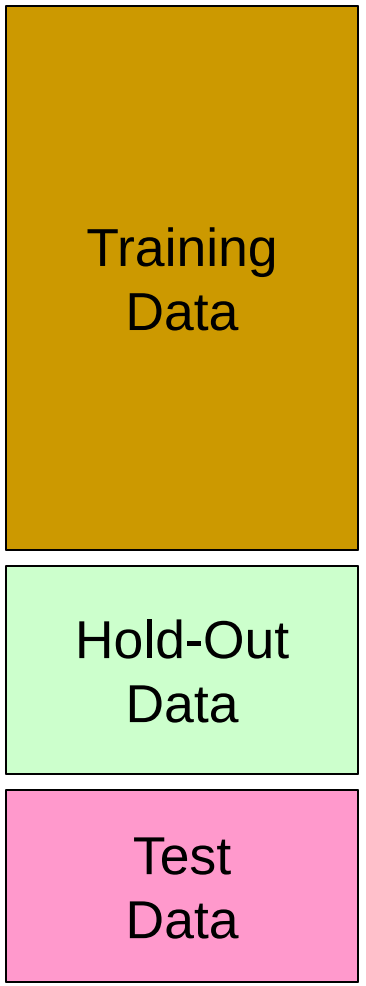


- Thousands of learning algorithms
- Combinations of just three elements

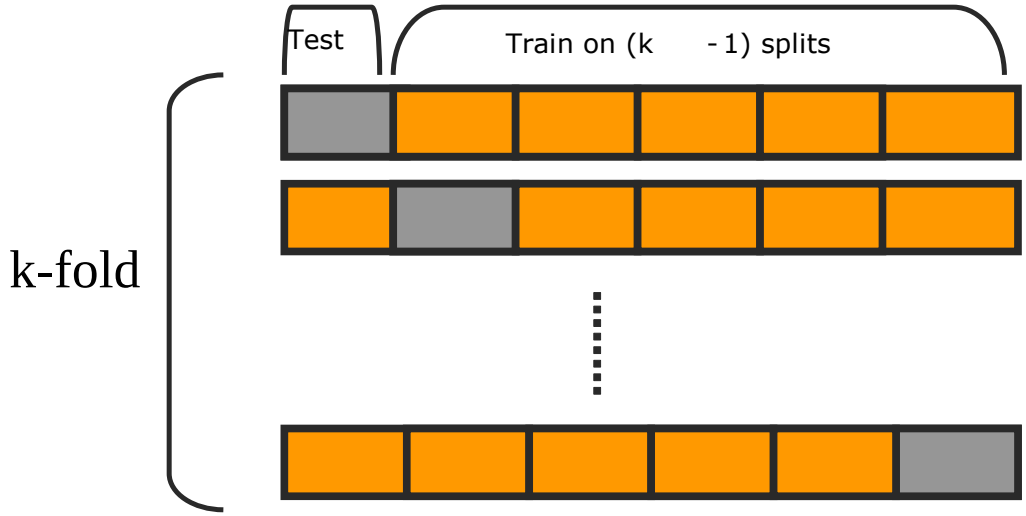
Representation	Evaluation	Optimization
Instances	Accuracy	Greedy search
Hyperplanes	Precision/Recall	Branch & bound
Decision trees	Squared error	Gradient descent
Sets of rules	Likelihood	Quasi-Newton
Neural networks	Posterior prob.	Linear progr.
Graphical models	Margin	Quadratic progr.
Etc.	Etc.	Etc.

It's Generalization that Counts

- Divide data into training, test and hold-out or validation set
- Algorithm must work on test examples never seen before
  - Training examples can just be memorized
- Don't tune parameters on test data
- Use cross-validation



K-Fold Cross Validation



- Choose a suitable K (usually 10)

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

Overfitting Has Many Faces

- **Classifier A is better than B on the training set but the reverse is true on the test set!!**
- The biggest problem in machine learning
- Bias and variance (Simple vs. Complex)
  - Can learn a simpler linear function vs. can learn any function
- Less powerful learners can be better
- Solutions: Cross-validation; Regularization

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

More Data Beats a Cleverer Algorithm

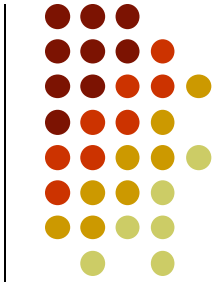
- Easiest way to improve: More data
- Then
  - Data is bottleneck
- Now:
  - Scalability is bottleneck
- ML algorithms more similar than they appear
- Clever algorithms require more effort but can pay off in the end
- Biggest bottleneck is human time

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.

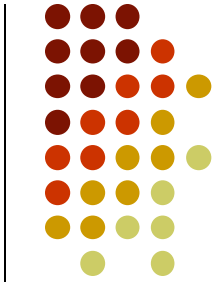
Representable Does Not Imply Learnable

- 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



ADVANCED TOPICS

Supervised Learning and its Generalizations



- Supervised Learning
  - Desired output is simple. (e.g., purchase an item or not; the person has the disease or not; etc.)
- Structured Prediction: is a Generalization
  - Desired output is complex.

Structured Prediction: Examples

- Parsing: given an input sequence, build a tree whose leaves are the elements in the sequence and whose structure obeys some grammar.
- Collective classification: given a graph defined by a set of vertices and edges, produce a labeling of the vertices.
  - Labeling web pages given link information

Models and Algorithms for Structured Prediction

- Probabilistic Graphical Models
  - Compact representation of joint distribution
  - Principled way of dealing with uncertainty
  - Take advantage of conditional independence
- Markov logic and statistical relational models
  - Model both relational structure and uncertainty
    - One example related with another example
- Considerable machine learning expertise required here! (not yet a blackbox)