CSC2515 Spring 2014

Introduction to Machine Learning

Lecture 1: Introduction

All lecture slides will be available as .pdf on the course website:
http://www.cs.toronto.edu/~urtasun/courses/CSC2515/CSC2515_Winter15.html

Many of the figures are provided by Chris Bishop
from his textbook: "Pattern Recognition and Machine Learning"

Admin Details

- Permanent tutorial time/place:
  - Thursdays 2-3, Haultain 401

- Do I have the appropriate background?
  - Linear algebra: vector/matrix manipulations, properties
  - Calculus: partial derivatives
  - Probability: common distributions; Bayes Rule
  - Statistics: mean/median/mode; maximum likelihood

  - Sheldon Ross: A First Course in Probability

- Related Courses

Textbooks

- Christopher Bishop:
  - ”Pattern Recognition and Machine Learning”, 2006.

- Other recommended texts

  - Kevin Murphy: Machine Learning: a Probabilistic Perspective

  - David Mackay: Information Theory, Inference, and Learning Algorithms

Requirements

- Do the readings!

- Assignments
  - Two assignments, worth 10% each
  - Programming: take Matlab/Python code and extend it
  - Derivations: pen(cil)-and-paper

- Test
  - Two hour exam on last day of class, check that understand main concepts in course
  - Worth 35% of course mark

- Project
  - Proposal due Jan 26
  - Presentations: Week March 23 (date might change)
  - Write-up due April 3rd (date might change)
  - Worth 45% of course mark

What is Machine Learning?

- Learning systems are not directly programmed to solve a problem, instead develop own program based on:
  - Examples of how they should behave
  - From trial-and-error experience trying to solve the problem

- Different than standard CS: want to implement unknown function, only have access to sample input-output pairs (training examples)

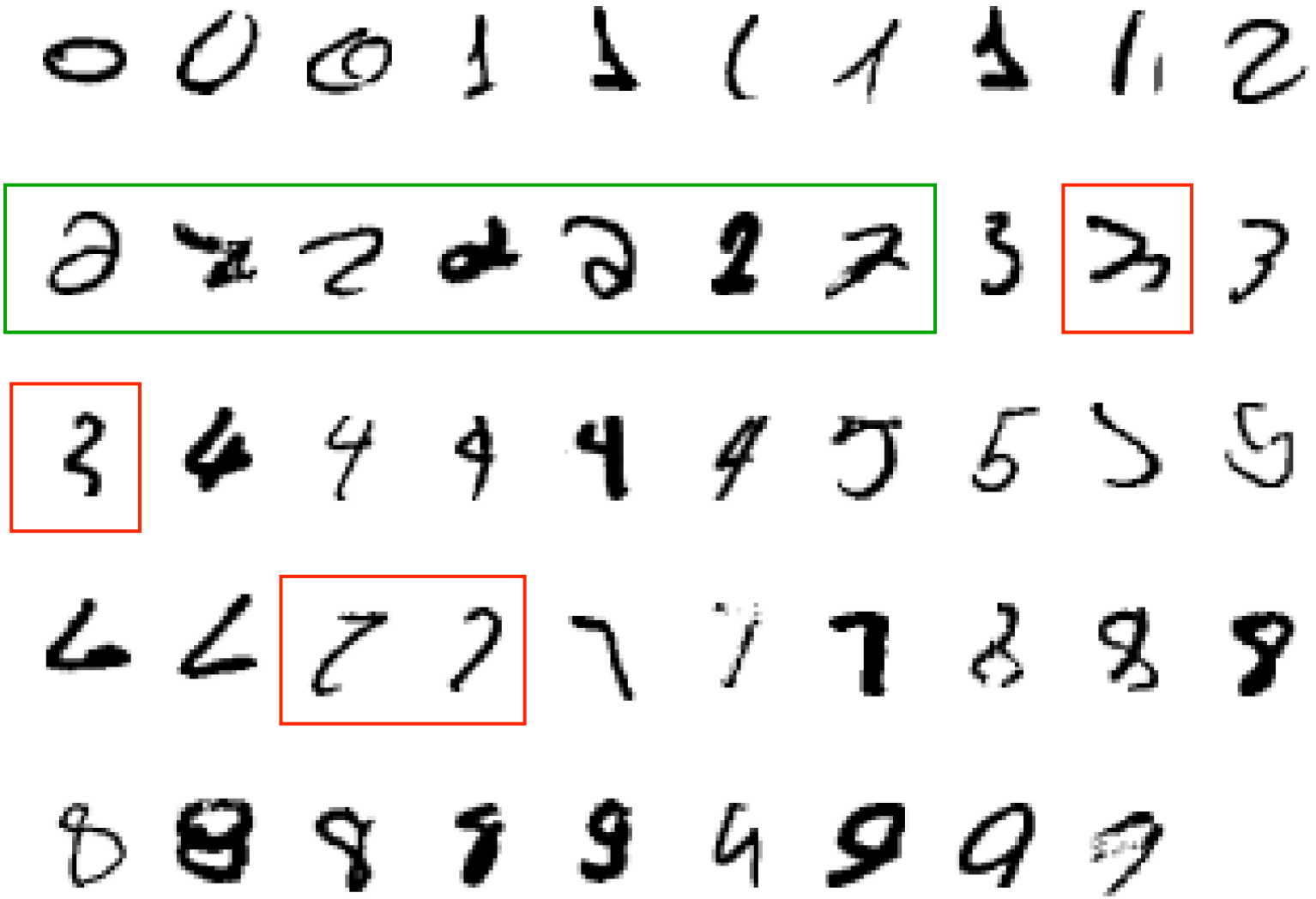
- Learning simply means incorporating information from the training examples into the system

Why Study Learning?

- Develop enhanced computer systems
  - Automatically adapt to user, customize
  - Often difficult to acquire necessary knowledge
- Improve understanding of human, biological learning
  - Computational analysis provides concrete theory, predictions
  - Explosion of methods to analyze brain activity during learning
- Timing is good
  - Ever growing amounts of data available
  - Cheap and powerful computers
  - Suite of algorithms, theory already developed

A classic example of a task that requires machine learning:

What makes a 2?



Why use learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.

Two classic examples of tasks that are best solved by using a learning algorithm

Date: Mon, 6 Sep 2027 05:08:33 -0400
From: Essence <Jonathan@wupperverband.de>
To: dcsprofs@cs.toronto.edu
Subject: Emerging Growth stock Opportunity

Big news expected.
This stock will explode.
Do not wait until it is too late.
Investment Times Alert Issues: (STRONG BUY

Learning algorithms are useful in other tasks

- Recognizing patterns:
  - Facial identities, expressions
  - Handwritten or spoken words
- Digital images and videos:
  - Locating, tracking, and identifying objects
  - Driving a car
- Recognizing anomalies:
  - Unusual sequences of credit card transactions
- Spam filtering, fraud detection:
  - The enemy adapts so we must adapt too
- Recommendation systems:
  - Noisy data, commercial pay-off (Amazon, Netflix).
- Information retrieval:
  - Find documents or images with similar content

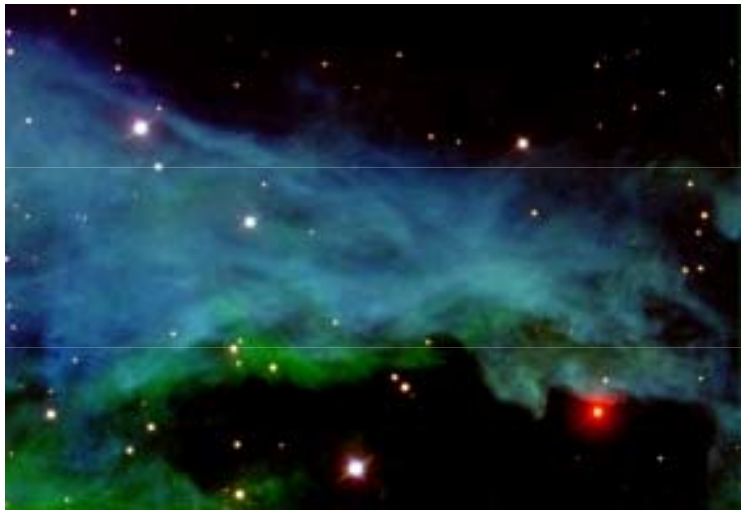
Data Explosion: Text

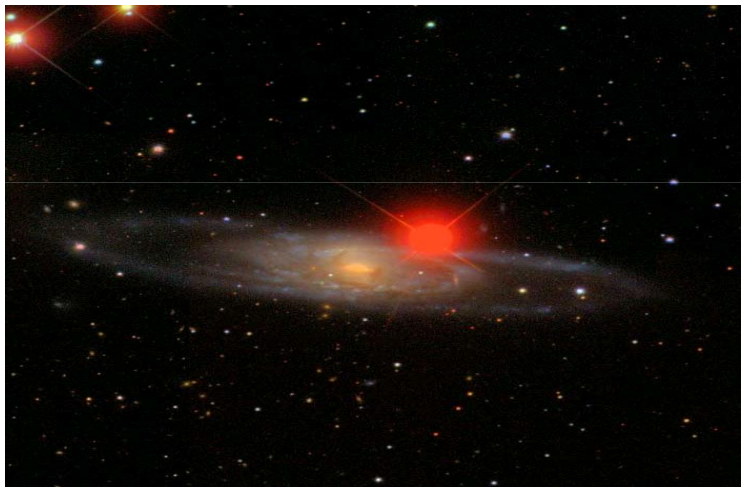
- "Large" text dataset
  - 1,000,000 words in 1967
  - 1,000,000,000,000 words in 2006

- Successful Applications
  - Speech recognition
  - Machine translation

  - Lots of labeled data
  - Memorization is useful

Really Big Data





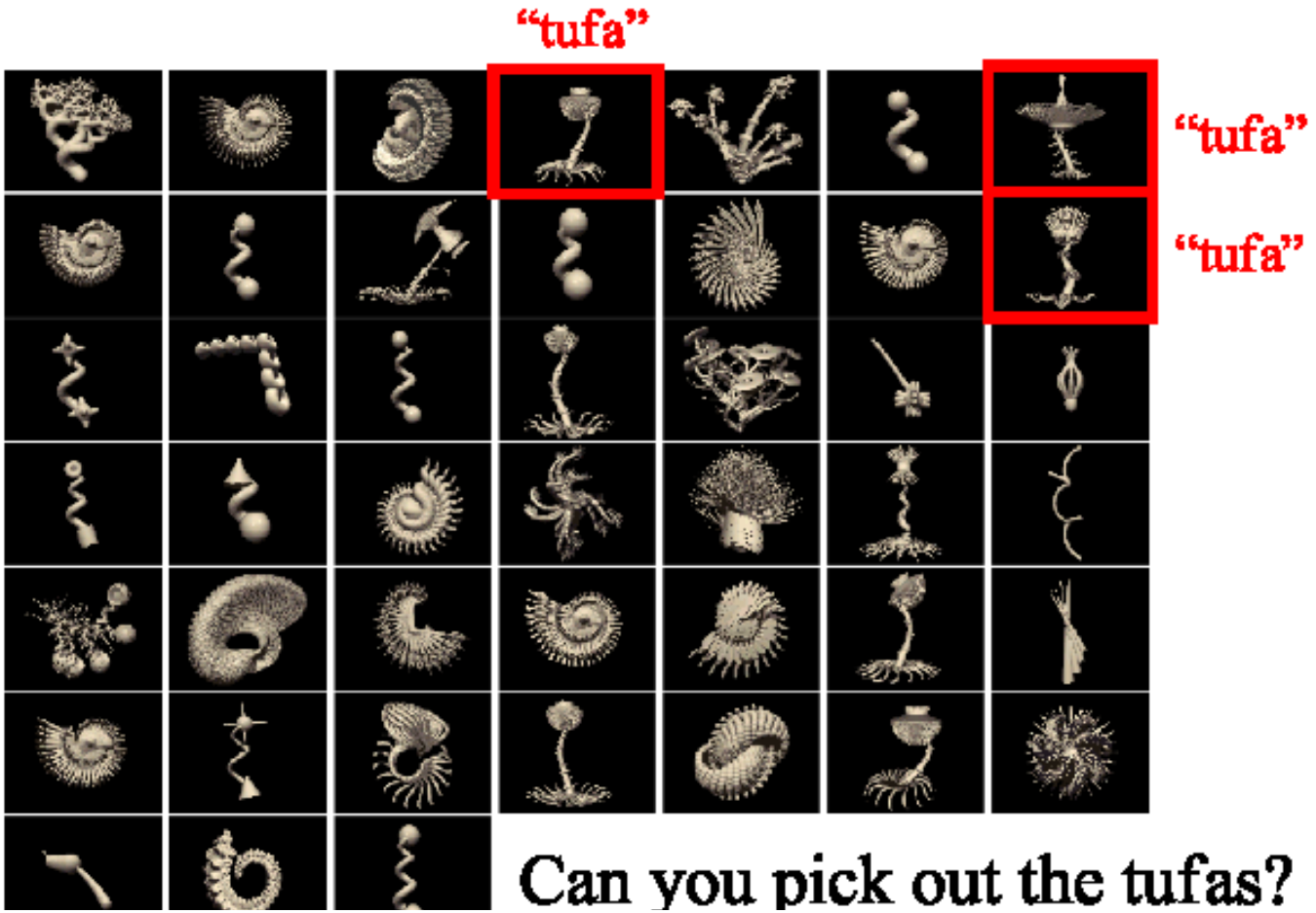
"When the **Sloan Digital Sky Survey** started work in 2000, its telescope in New Mexico collected more data in its first few weeks than had been amassed in the entire history of astronomy.

Now, a decade later, its archive contains a whopping **140 terabytes** of information.

A successor, the **Large Synoptic Survey Telescope**, due to come on stream in Chile in 2016, will acquire that quantity of data every five days."

[*The Economist* February 2010]

Human learning



Josh Tenenbaum

Types of learning task

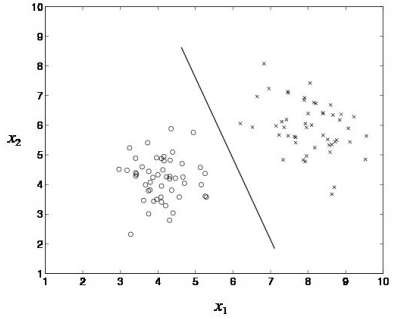
- Supervised: correct output known for each training example
  - Learn to predict output when given an input vector
    - Classification: 1-of-N output (speech recognition, object recognition, medical diagnosis)
    - Regression: real-valued output (predicting market prices, customer rating)

- Unsupervised learning
  - Create an internal representation of the input, capturing regularities/structure in data
  - Examples: form clusters; extract features
    - How do we know if a representation is good?

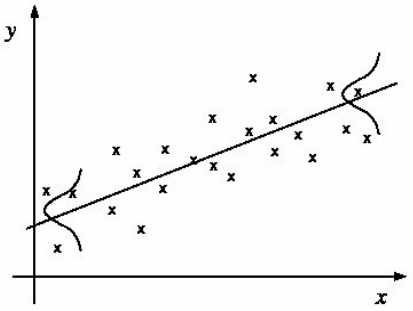
- Reinforcement learning
  - Learn action to maximize payoff
    - Not much information in a payoff signal
    - Payoff is often delayed
    - Important area not covered here, many applications: games, SmartHouse

Supervised Learning

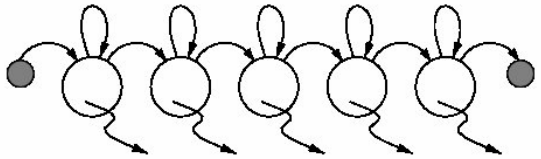
- Classification
  - Outputs are categorical (1-of-N)
  - Inputs are anything
  - Goal: select correct class for new inputs
  - Ex: speech, object recognition, medical diagnosis



- Regression
  - Outputs are continuous
  - Inputs are anything (typically continuous)
  - Goal: predict outputs accurately for new inputs
  - Ex: predicting market prices, customer rating of movie



- Temporal Prediction
  - Goal: perform classification/regression on new input sequences values at future time points
  - Given input values and corresponding class labels/outputs at some previous time points



Unsupervised Learning

- Clustering:
  - Inputs are vector or categorical
  - Goal: group data cases into a finite number of clusters so that within each cluster all cases have very similar inputs
- Compression
  - Inputs are typically vector
  - Goal: deliver an encoder and decoder such that size of encoder output is much smaller than original input, but composition of encoder followed by decode very similar to original input
- Outlier detection
  - Inputs are anything
  - Goal: select highly unusual cases from new and given data

Machine Learning & Data Mining

- Data-mining: Typically using very simple machine learning techniques on very large databases because computers are too slow to do anything more interesting with ten billion examples

- Previously used in a negative sense – misguided statistical procedure of looking for all kinds of relationships in the data until finally find one

- Now lines are blurred: many ML problems involve tons of data

- But problems with AI flavor (e.g., recognition, robot navigation) still domain of ML

Machine Learning & Statistics

- ML uses statistical theory to build models – core task is inference from a sample
- A lot of ML is rediscovery of things statisticians already knew; often disguised by differences in terminology:
- But the emphasis is very different:
  - Good piece of statistics: Clever proof that relatively simple estimation procedure is asymptotically unbiased.
  - Good piece of ML: Demo that a complicated algorithm produces impressive results on a specific task.
- Can view ML as applying computational techniques to statistical problems. But go beyond typical statistics problems, with different aims (speed vs. accuracy).

Cultural gap (Tibshirani)

Machine Learning----------Statistics

- network, graphs
- weights
- learning
- generalization
- supervised learning

- unsupervised learning.

- large grant: $1,000,000
- conference location: Snowbird, French Alps

- model
- parameters
- fitting
- test set performance
- regression/classification

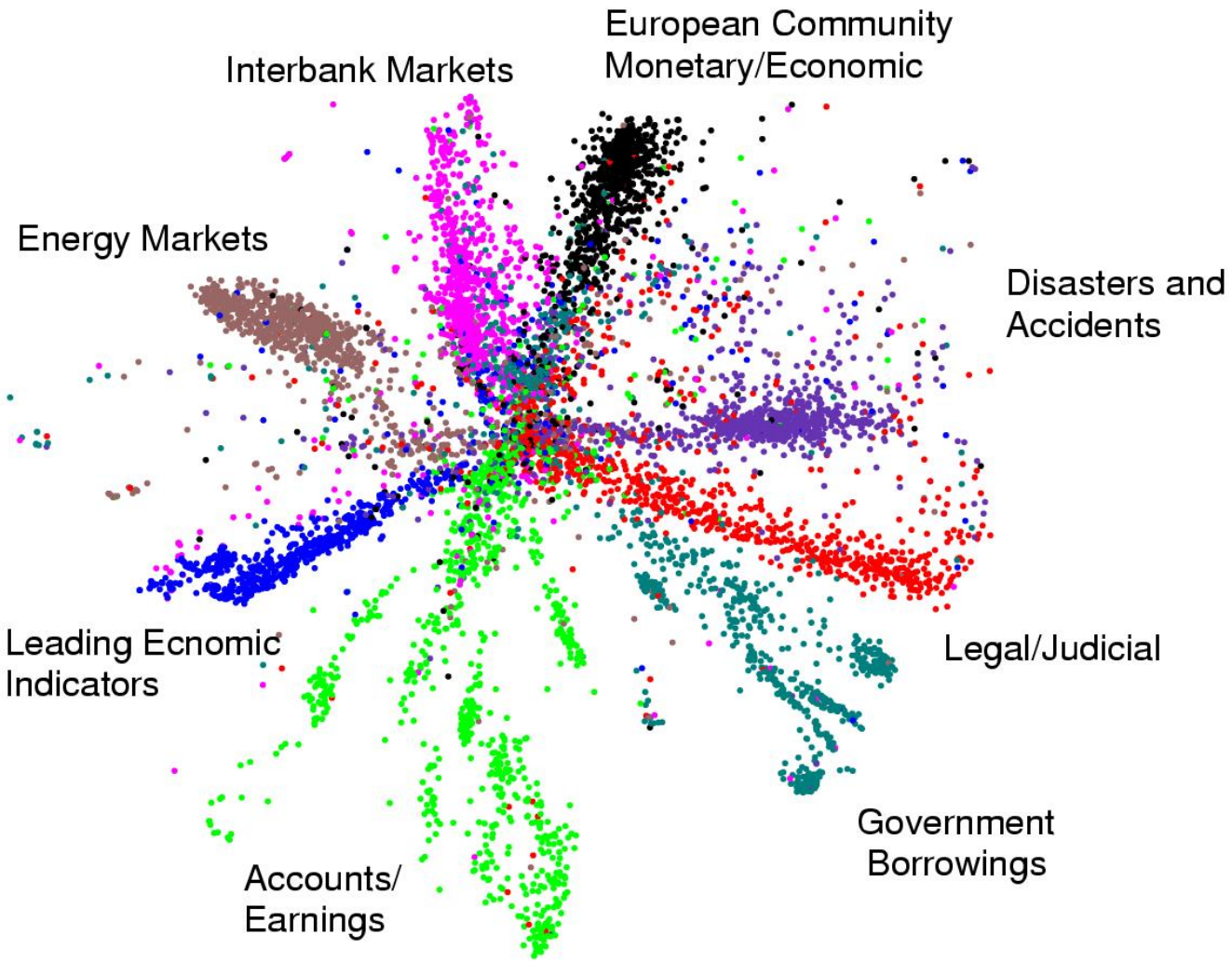
- density estimation, clustering
- large grant: $50,000
- conference location: Las Vegas in August

Representing the structure of a set of documents using Latent Semantic Analysis (a form of PCA)

Each document is converted to a vector of word counts. This vector is then mapped to two coordinates and displayed as a colored dot. The colors represent the hand-labeled classes.

When the documents are laid out in 2-D, the classes are not used. So we can judge how good the algorithm is by seeing if the classes are separated.

Representing the structure of a set of documents using a neural network



Using Variables to Represent the World

- We use mathematical variables to encode everything we know about the task: inputs, outputs and internal states.
- Variables may be discrete/categorical; continuous/vector
  - Discrete quantities take on one of a fixed set of values
    e.g., {0,1}, {email,spam}, {sunny,overcast,raining}
  - Continuous quantities take on real values
    e.g.. 1.6632, [3.3,-1.8,120.4]
- Generally have repeated measurements of same quantities
- Conventions
  - *i,j,…* indexes components/variables/dimensions
  - *n,m,…* indexes cases/records
  - $x_i^{(n)}$: value of the i^{th} input variable on the n^{th} case
  - $y_j^{(m)}$: value of the j^{th} output variable on the m^{th} case
  - $\boldsymbol{x}^{(n)}$: vector of inputs for the n^{th} case
  - $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}\, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ is all the inputs

Initial Case Study

- What grade will I get in this course?
- Data: entry survey and marks from previous years
- Process the data
  - Split into training set; test set
  - Determine representation of input features; output
- Choose form of model: linear regression
- Decide how to evaluate the system's performance: objective function
- Set model parameters to optimize performance
- Evaluate on test set: generalization

Hypothesis Space

- Now have a representation for inputs and outputs
- How to represent a supervised learning machine?
- One way to think about a supervised learning machine is as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.
  - If the data is noise-free, each training example rules out a region of hypothesis space.
  - If the data is noisy, each training example scales the posterior probability of each point in the hypothesis space in proportion to how likely the training example is given that hypothesis.
- The art of supervised machine learning is in:
  - Deciding how to represent the inputs and outputs
  - Selecting a hypothesis space that is powerful enough to represent the relationship between inputs and outputs but simple enough to be searched.

Searching a hypothesis space

- The obvious method is to first formulate a loss function and then adjust the parameters to minimize the loss function.
  - This allows the optimization to be separated from the objective function that is being optimized.
- Bayesians do not search for a single set of parameter values that do well on the loss function.
  - They start with a prior distribution over parameter values and use the training data to compute a posterior distribution over the whole hypothesis space.

Some Loss Functions

- Squared difference between actual and target real-valued outputs

- Number of classification errors
  - Problematic for optimization because the derivative is not smooth

- Negative log probability assigned to the correct answer.
  - This is usually the right function to use.
  - In some cases it is the same as squared error (regression with Gaussian output noise)
  - In other cases it is very different (classification with discrete classes needs cross-entropy error)