Herbert Wertheim College of Engineering
UNIVERSITY *of* FLORIDA

Introduction to Machine Learning

Lecture 00

POWERING THE NEW ENGINEER TO TRANSFORM THE FUTURE

Instruction Team:

Dr. Alina Zare
Associate Professor in Electrical and Computer Engineering
Director of the Machine Learning and Sensing Lab
University of Florida

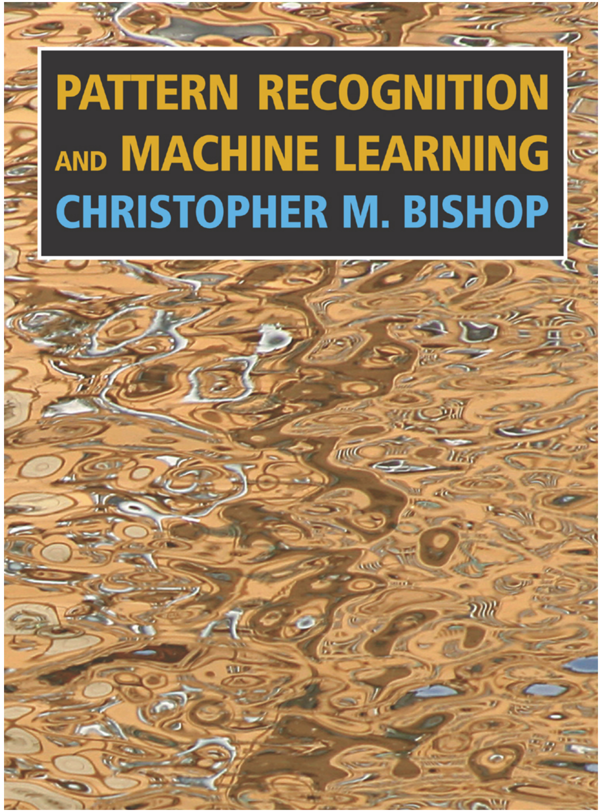
Connor McCurley
Teaching Assistant
Member of the Machine Learning and Sensing Lab

Xiaolei Guo
Teaching Assistant
Member of the Machine Learning and Sensing Lab

Daniel Wells
Teaching Assistant

Textbook

Title: Pattern Recognition and Machine Learning
Author: Christopher M. Bishop
Publisher: Springer
Year: 2006



- ***Course Overview***: Introduction to **machine learning** and its role in variety of real-world problems in areas such as remote sensing and image processing.

- ***Course Overview***: Introduction to **machine learning** and its role in variety of real-world problems in areas such as remote sensing and adaptive filtering.
- So, What is *machine learning?*
- *Can a machine or computer learn?*
- *Can a machine or computer be intelligent?*

- ***Course Overview***: Introduction to **machine learning** and its role in variety of real-world problems in areas such as remote sensing and adaptive filtering.
- So, What is *machine learning?*
- *Can a machine or computer learn?*
- *Can a machine or computer be intelligent?*
- One definition of ***Machine Learning:*** Area of study to develop methods for computers to make (intelligent?) decisions without being explicitly programmed.

Many Sub-areas in Machine Learning

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning
- Multiple Instance Learning
- Active Learning
- Neural Networks & Deep Learning
- Transfer Learning
- Structured Learning
- Associative Learning
- ….

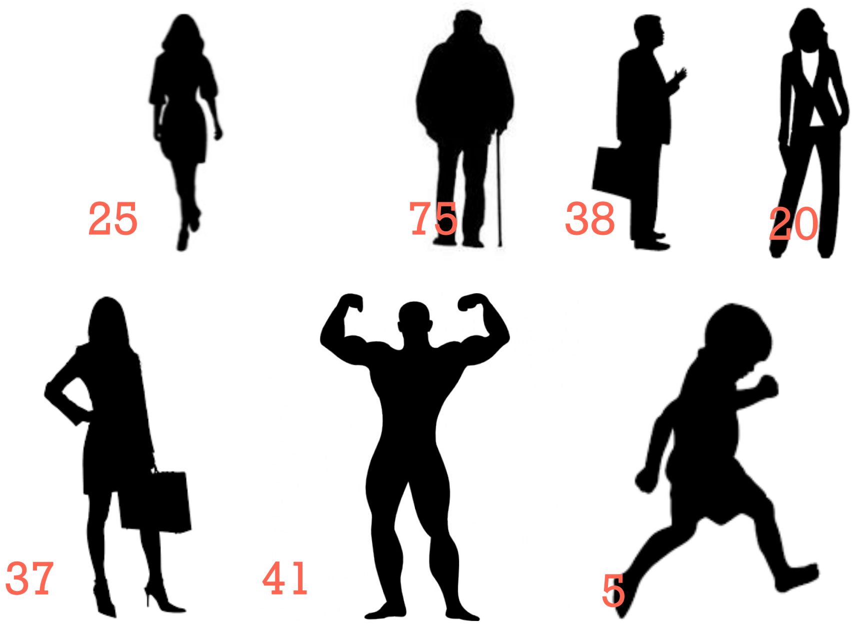
Supervised Learning

Learning mapping from input data to desired output values given labeled training data

0: Macaw 1: Conure

Supervised Learning

Learning mapping from input data to desired output values given labeled training data



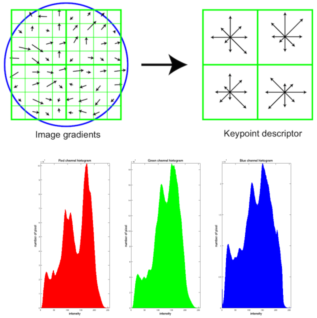
Supervised Learning

The Usual Flow (but not always)

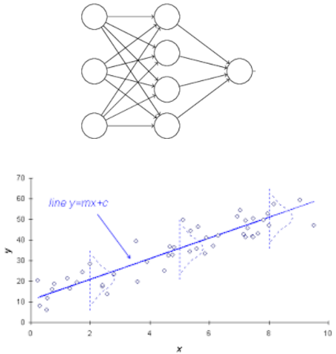
Training:

Collect Labeled Training Data

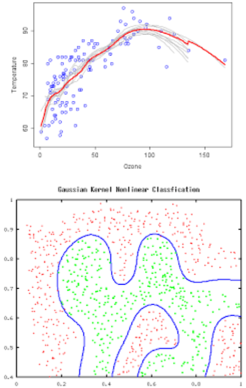
Extract Features



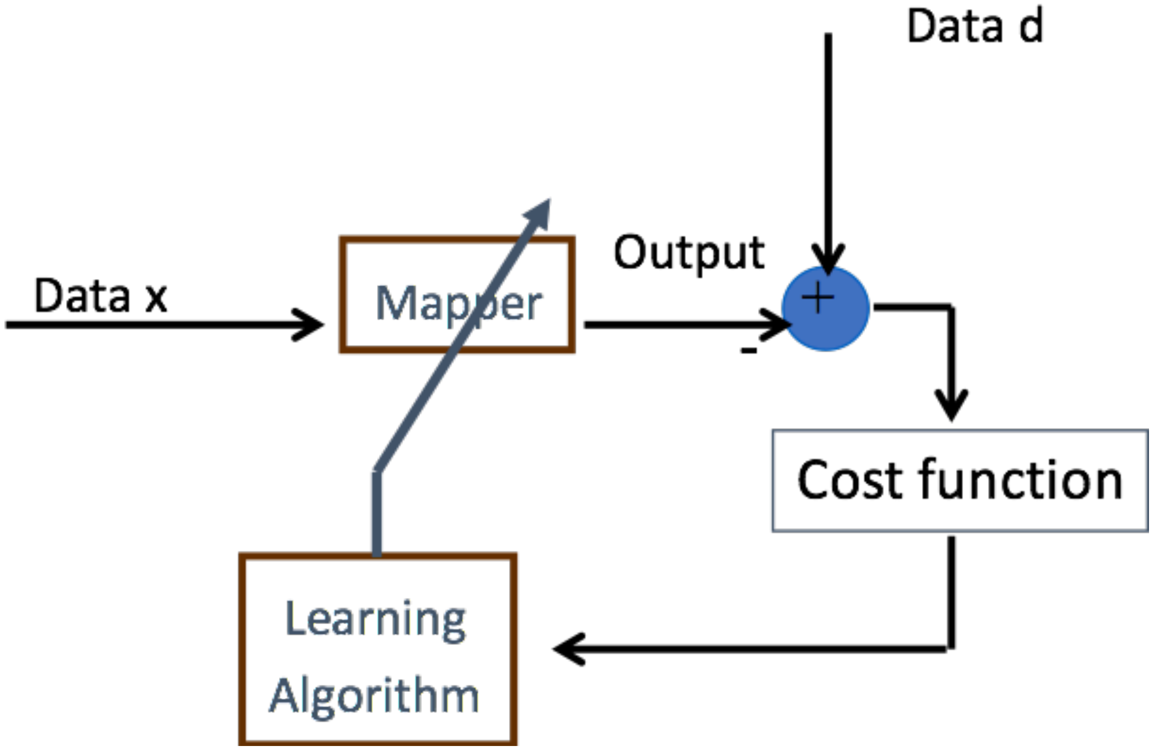
Select a Model



Fit the Model



Block Diagram of a Learning System



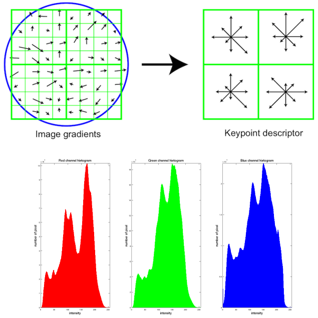
Supervised Learning

The Usual Flow (but not always)

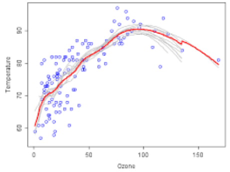
Testing:

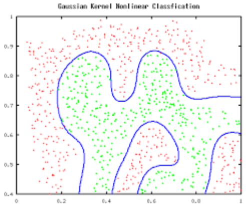
Given Unlabeled Test Data

Extract (the same) Features



Run It Through Your Trained Model





(Subset of) Challenges

How do you know if you have *representative* training data?

How do you know if you extracted *good* features?

How do you know if you selected the *right* model?

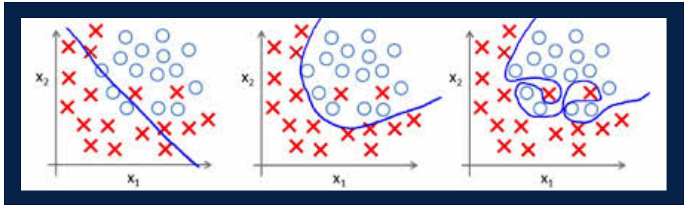
How do you know if you trained the model *well*?

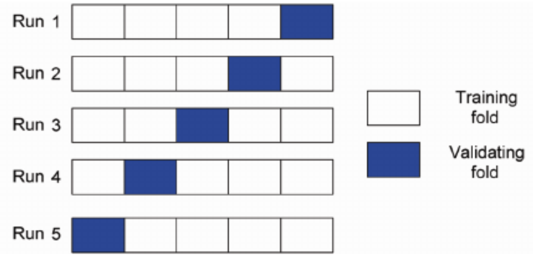
Gets Loads and Loads of Data

Partition (thoughtfully) into Training, Validation, & Testing Data

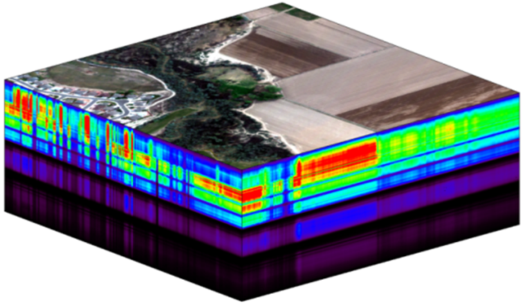
Conduct Cross-Validation

Carefully Select Evaluation Metrics





Obtaining Labeled Training Data is often hard, expensive, and sometimes infeasible…



From NEON
neonscience.org

Unsupervised Learning

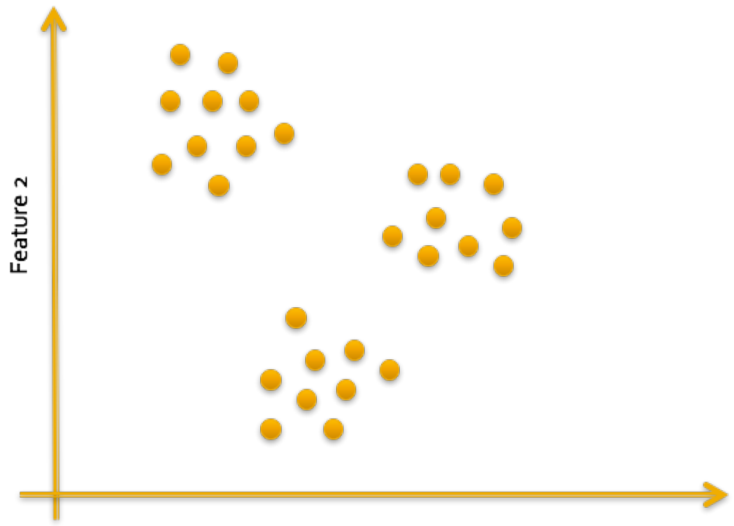
Learning structure from data *without any labels*

Unsupervised Learning

Learning structure from data *without any labels*

Unsupervised Learning

Learning structure from data *without any labels*

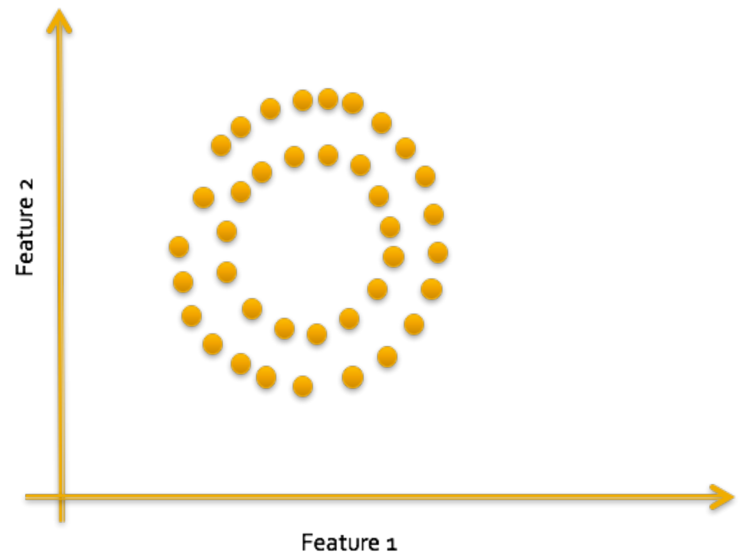


Unsupervised Learning

Learning structure from data *without any labels*

Unsupervised Learning

Learning structure from data *without any labels*



Other Sub-areas of Machine Learning

Semi-supervised Learning - some training data labeled, some not, use all during training

Reinforcement Learning - reinforcement based on action in an environment so to maximize/minimize a reward/penalty

Active Learning - obtaining labels online from a user/oracle in an *intelligent* fashion

Transfer Learning - having labels on a related problem and transferring it to the task of interest

Multiple Instance Learning – have only imprecise labels for training data

Manifold Learning – non-linear dimensionality reduction of embedded data while preserving characteristic properties

Before Next Class:

- Read Chapter 1 (Introduction) and Appendix B (Linear Algebra) in the Textbook
- Prepare to be able to run and use Python 3 Jupyter Notebooks in class. Easiest method is to install Anaconda. See: http://jupyter.readthedocs.io/en/latest/install.html
- Complete Homework 0 (including the git and python tutorials)

Supplemental Slides

Machine Learning terminology

There is some common terminology that we will be using throughout:

Feature, attribute, characteristic: A measurement made for an object (e.g., length, width, height, color, mass, shape, texture, and frequency).

Observation, sample, input data, example inputs: A collection of features that succinctly describe an object (e.g., a human face, handwriting sample, speech pattern). The features are often combined to form a numerical or symbolic feature vector for each observation.

Class, category, group, label: A set of related observations that all share the same label vector. Labels are usually numerical. They can sometimes be symbolic.

Mapping, Model: A transformation by a mathematical function from one domain to another (e.g., a transformation from feature vectors to class label vectors).

Manifold: manifolds are topological spaces that can take many shapes. They “locally” preserve Euclidean space properties but can be equipped with more structure than a locally Euclidean topology

Machine Learning

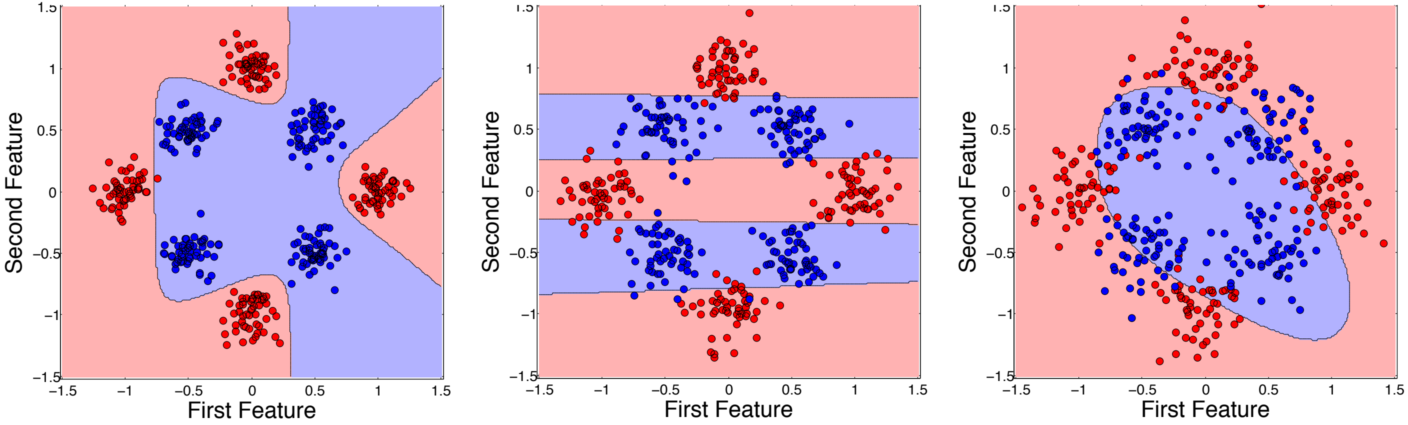
Machine learning can be defined as: "the field of study that gives computers the ability to learn without being explicitly programmed." In essence, machine learning explores the study and construction of methods that can learn from and make predictions about data. Such methods work by building mappings/models from example inputs (usually vector-based) and sometimes example outputs (usually vector-based) to make data-driven predictions or decisions.

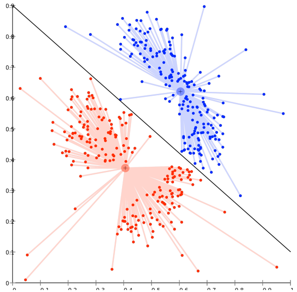
Machine learning can be applied in situations where it is very challenging to manually enumerate all possible rules (e.g., face detection, speech recognition, object classification). It differs from classical artificial intelligence, which is predominantly rule- and logic-based (e.g., decision trees).

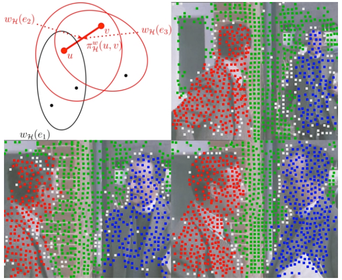
Supervised learning

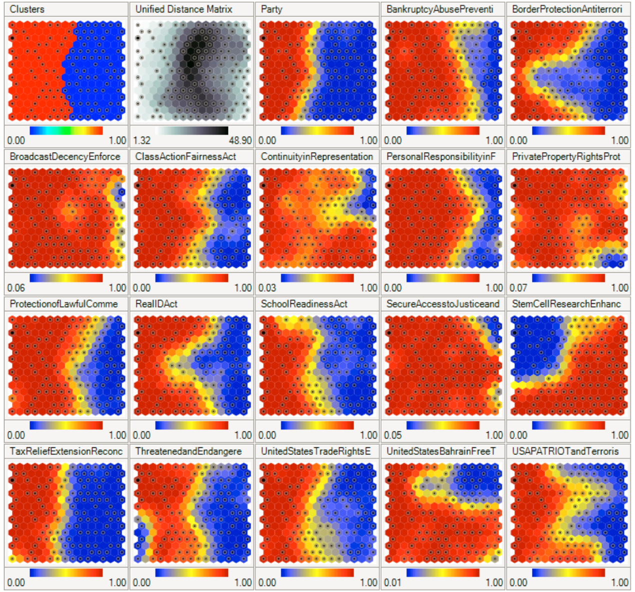
Many types of applications are instances of supervised learning. Supervised learning involves inferring an input-output mapping from labeled training observations (usually vectors) and desired responses (usually vectors).

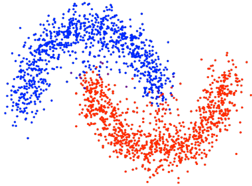
The most common type of supervised learning is classification: want to discriminate between observations of different classes/categories. We will cover both linear (e.g., linear discriminant analysis) and non-linear classification methods (e.g. neural networks)

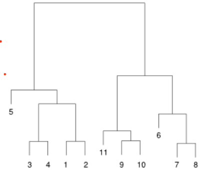


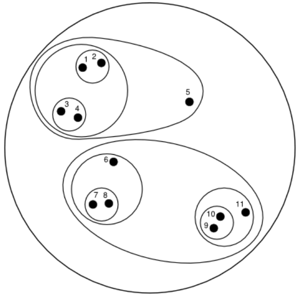






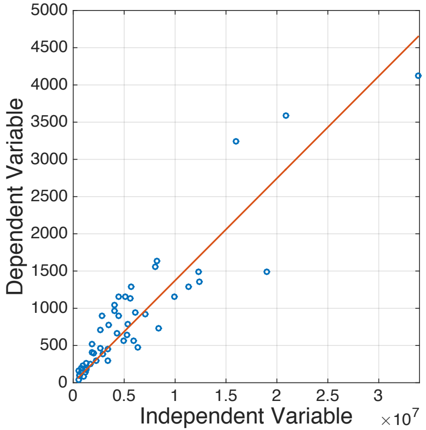


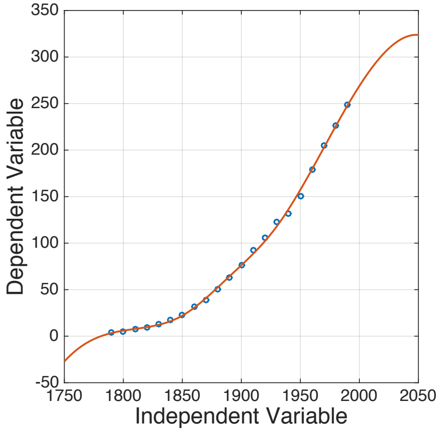




Sometimes the observations are not labeled by class. We would like to understand something about the structure of the observations using prior specifications of what we think the structure should look like (e.g., spheres or curves). This is referred to as unsupervised learning.

We will cover a variety of clustering methods.



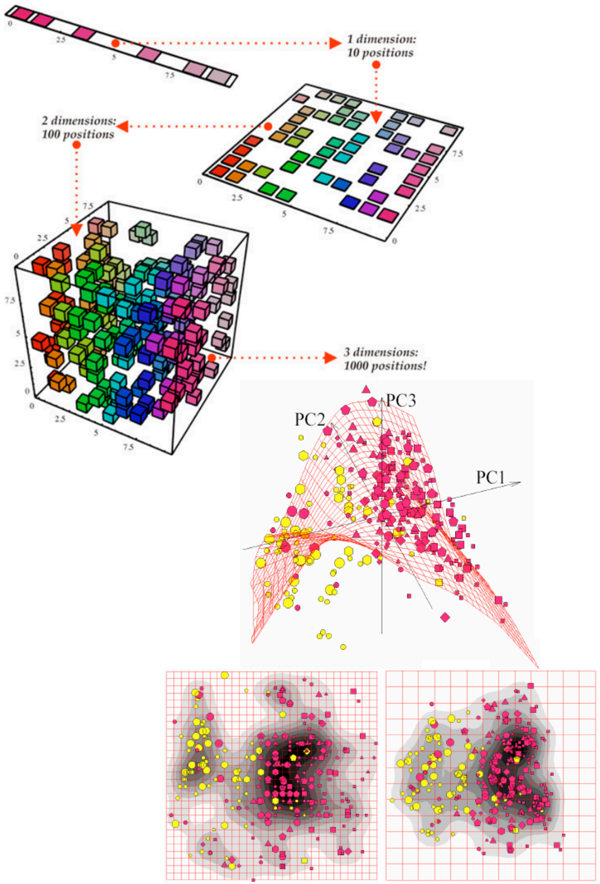


For some problems, we will be interested in understanding the relationship between a series of vector observations (independent variables) and the corresponding responses (dependent variable). These variables could be non-dimensional, temporal, spatial, or even spatio-temporal. We would like to obtain a linear or non-linear model that provides a good mapping between the two types of variables. This process is known as regression.

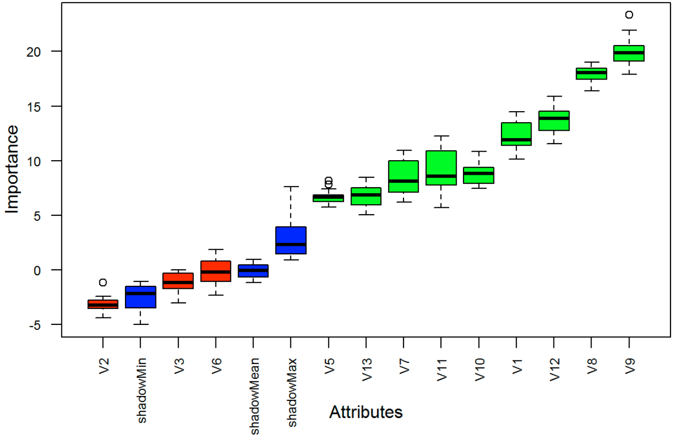
We will cover linear and non-linear, kernel-based regression approaches.

Sometimes the dimensionality of the observations is very high. This means that we might need a large number of model parameters to accurately map the data. We might also have high model training times. A very large number of samples may be needed to overcome this curse of dimensionality.

We can use dimensionality reduction techniques to remove redundant features from the observations, which partly solves many of these issues. Such techniques map the original observations to a lower-dimensional space, in a lossy manner, such that the important attributes of the observations are usually retained. This sometimes retains much of the original observation distributions.



An important first step to machine learning is feature generation. We have to decide what features are worth measuring to describe an object (application dependent). We then have to determine how to reliably, repeatedly, and efficiently measure those features.



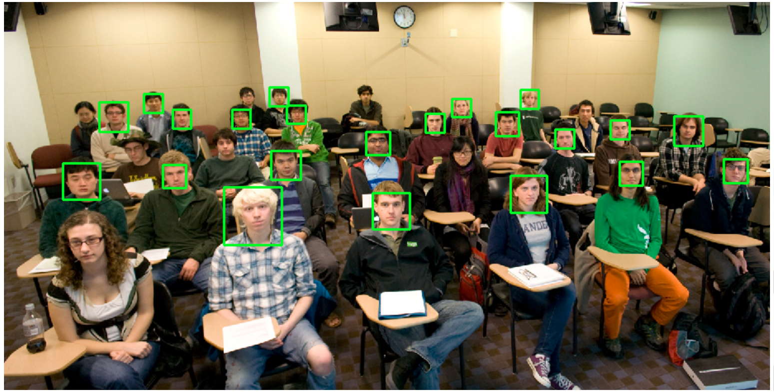
Once we have a set of features, we need a way to evaluate their ability to describe the objects and hence produce good models. We want to select as few features as possible to reduce training time, data set sizes, and poor generalization performance. This can be done using feature selection techniques.

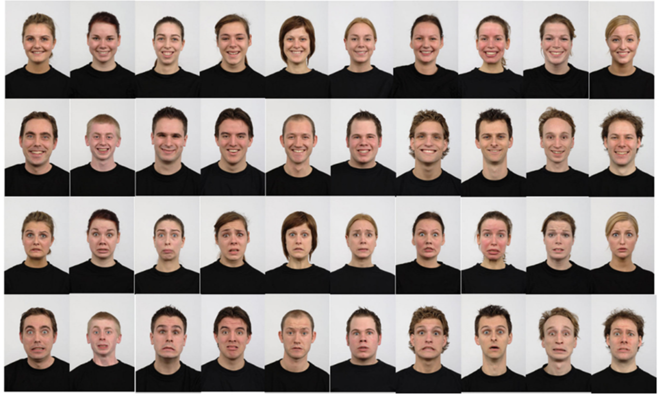
Machine Learning examples

Goal: Want to develop a system that can determine the hand-written numerical digit from images.

Data (MNIST) : Pre-segmented 28x28 pixel images of a single digit. 70k images are provided from different people.

Process: Unwrap each image so that it is a vector of size 781x1. Use a set of training data (e.g., 40-60k images of different digits written by different people) to learn a mapping from that space to the set of integers. Use a set of testing data (e.g., 10-20k images, which are not in the training set) to evaluate the mapping quality.





Goal: Want to develop a system that can detect all faces in images. Want to do this for different viewpoints, expressions, poses, and lighting conditions.

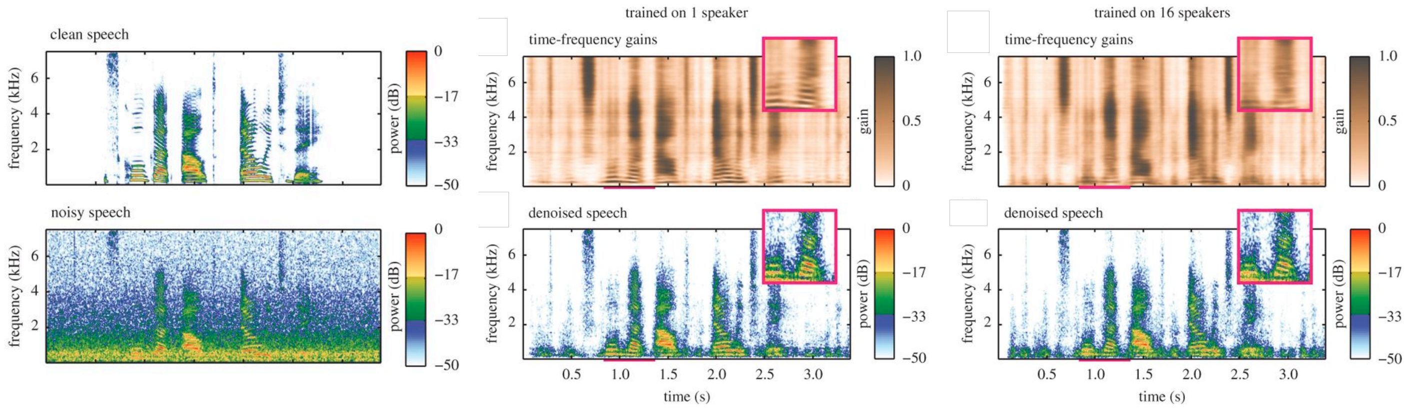
Data (FDDB): Pre-segmented, variable-sized images of a single face under different conditions. 5,171 images are provided.

Process: Unwrap each image so that it is a column. Apply a transformation to reduce the vector dimensionality before learning. Use a set of training data (e.g., 3,500 images) to learn a mapping. Use a set of testing data (e.g., 1,671 images, which are not in the training set) to evaluate the mapping.

Goal: Want to develop a system that can determine if a query fingerprint matches an existing fingerprint in a database. Want to handle partial prints and different print orientations.

Data (CASIAv5) : Pre-segmented 328x356 pixel images of a single digit. 20k images are provided from 500 people.

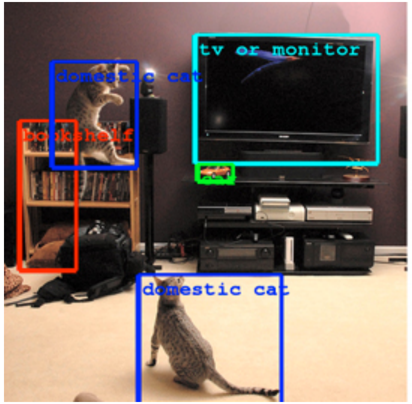
Process: Threshold the grayscale values in each image. Look for “interesting” patterns in the fingerprint, called minutiae points. Find a linear or non-linear transformation between sets of minutiae points for pairs of prints. Determine how many minutiae points match, within some tolerance. Evaluate this system on the entire dataset (no training is needed).

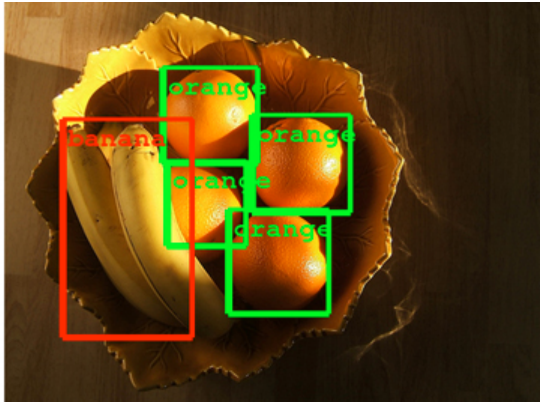


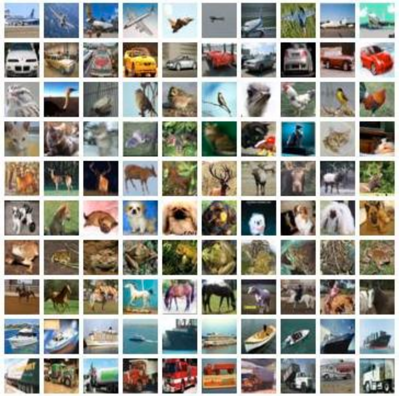
Goal: Want to develop a system that can learn a person's speech patterns and filter out ambient background noise using a single microphone (many noise cancelling systems need more than one microphone to work well).

Data: Temporally pre-segmented speech segments of varying length under different noise conditions.

Process: Learn a set of linear or non-linear filter coefficients that best transform a person's noisy speech pattern to a corresponding noise-free version. Alternatively, learn a mapping for noise conditions that can be universally applied.







Goal: Want to develop a system that can recognize objects in images. Want to do this for different camera viewpoints and lighting conditions.

Data (CIFAR-10): Pre-segmented 32x32 pixel images of objects under different conditions. 60k images are provided for 10 object classes.

Process: Unwrap each image so that it is a vector of size 1024x1. Apply a transformation to reduce the vector dimensionality before learning. Use a set of training data (e.g., 30-45k images) to learn a mapping. Use a set of testing data (e.g., 15-30k images, which are not in the training set) to evaluate the mapping.