Intro to Machine Learning

Recitation 1: Intro, KNN, K-Means
Administrative Stuff

- Name: Yossi Adi and Felix Kreuk
- Number of assignments: 6
  - Programming (python) and theoretical exercises
  - Checked with the “Submit” system.
- % assignments, % test
- Office hours: ??
- Piazza: ?
- Project ?
Machine Learning is Everywhere

INTERNET & CLOUD
Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation

MEDICINE & BIOLOGY
Cancer Cell Detection
Diabetic Grading
Drug Discovery

MEDIA & ENTERTAINMENT
Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENCES
Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES
Pedestrian Detection
Lane Tracking
Recognize Traffic Sign
Machine Learning

Herbert Alexander Simon:

“Learning is any process by which a system improves performance from experience.”

“Machine Learning is concerned with computer programs that automatically improve their performance through experience. “
Why use Machine Learning?

● Develop systems that can automatically adapt themselves
  ○ Personalised systems (facebook feed, google search)
● Discover new knowledge from large databases (cluster users)
● Market basket analysis (e.g. diapers and beer)
● Mimic human abilities in mundane tasks
  ○ Recognising handwritten characters
  ○ Speech recognition
  ○ Image annotation
● Develop systems that are too difficult to construct manually (if & else)
Why Now?

- Flood of data
- Computational power
- Research Boost (academic and industry)
What is Learning?

Learning = Improving with experience at some task

Improve over task - T

With respect to performance measure - P

Based on experience - E
Example - image classification

{cat, dog, …, car}

*Adapted from Stanford cs231n course presentations.
Example - image classification

*Adapted from Stanford cs231n course presentations.*
Challenges: viewpoint

All pixels change when the camera moves!

*Adapted from Stanford cs231n course presentations.
Challenges: illumination

*Adapted from Stanford cs231n course presentations.*
Challenges: Deformation

*Adapted from Stanford cs231n course presentations.
Challenges: Occlusion

*Adapted from Stanford cs231n course presentations.*
Challenges: Background Clutter

*Adapted from Stanford cs231n course presentations.
Challenges: Intraclass variation

*Adapted from Stanford cs231n course presentations.
Learning Schemes

- Supervised Learning
  - Examples are labeled
- Unsupervised
  - No labels
- Semi-Supervised
  - Partially labeled
- Reinforcement Learning
  - Reward
Outputs

- Classification: one over K classes
  - Object classification, text classification, etc.
- Regression: predict real valued vector
  - Stock values, height prediction, etc.
- Structured: predict complex outputs with structure
  - Parsing trees, ASR, translation, etc.
Separators and Generalisation

Which one would you choose?
Separators and Generalisation
Separators and Generalisation

Linear case - two mistakes

Non Linear case - zero mistakes
Terminology

- Target function: \( t : X \rightarrow Y \)
  - In classification: \( Y = \{1, 2, \cdots, k\} \)
  - In regression: \( Y = \mathbb{R}^n \)
- Hypothesis: A proposed function \( h \), an approximation of \( f \).  
- Hypothesis space: The space of all hypotheses that can, in principle, be output by the learning algorithm.
- Model: A function \( f \). The output of our learning algorithm.
- Loss: an evaluation metric: \( \ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+ \)
Terminology: Empirical Risk Minimization

Goal: to minimize:

$$\mathbb{E}_{(x,y) \sim \rho} [\ell(y, \hat{y}(x))]$$

Rho is unknown, so we use a set of examples:

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$$

We would like to minimize:

$$\frac{1}{m} \sum_{i=1}^{m} \ell(y_i, \hat{y}_w(x_i))$$
Terminology

- Training set
  - Used to train the classifier
- Validation set
  - Used to tune hyper-parameters
- Test set
  - Not seen during training, used to evaluate our classifier
- Epoch: one “pass” over the training set
Training Process

- Training loop:
  - Modify predictor function according to the training set
  - Evaluate the loss on the validation set
- Evaluate the loss on test set
- Output predictor function (which performs “well” on the test set according to the loss function)
Terminology: overfitting and underfitting

![Graph showing overfitting and underfitting](image)
$K$ Nearest Neighbours
**K Nearest Neighbours (KNN)**

- Our first machine learning algorithm
- One of the simplest algorithms
- Algorithm:
  - Given a test example $x$
  - Define a metric $d$:
    - Non-negative
    - Identity
    - Symmetry
    - Triangle inequality
  - Find the $k$ closest examples to $x$ according to $d$
  - Predict the majority class
$K$ Nearest Neighbours ($K$NN) - Properties

- Supervised
- Lazy
- Need to specify $K$
- Non-Linear
- Different metrics will output different boundaries
- New examples directly change the classifier
- Every dimension contributes equally
- Sensitive to outliers
$K$ Nearest Neighbours (KNN) - Demo

- Demo
$K$ Means
K Means

- The k-means algorithm is an iterative method for clustering a set of N points into K clusters.

- Algorithm:
  - Define a metric $d$
  - Initialise K centroids
  - Repeat until convergence:
    - Assign each point to the closest centroid according to $d$
    - Update each centroid to be the mean of the points in its group.
**K Means - Properties**

- Unsupervised
- Need to specify K
- Different centroids and metrics will output different boundaries
- Assumptions: spherical, same size and density
- Sensitive to outliers
$K$ Means - Demo

- Demo
Questions?