INTRODUCTION TO STATISTICAL MACHINE LEARNING

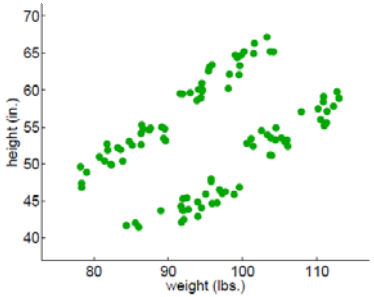
Xiaojin Zhu jerryzhu@cs.wisc.edu

Outline

- Representing "things"
  - Feature vector
  - Training sample
- Unsupervised learning
  - Clustering
- Supervised learning
  - Classification
  - Regression

Little green men

- The weight and height of 100 little green men



- What can you learn from this data?

Representing "things" in Machine Learning

- An instance x represents a specific object ("thing")
- x often represented by a D-dimensional feature vector $x = (x_1, \ldots, x_D) \in R^D$
- Each dimension is called a feature. Continuous or discrete.
- x is a dot in the D-dimensional feature space
- Abstraction of object. Ignores any other aspects (two men having the same weight, height will be identical)

Feature Representation Example

- Text document
  - Vocabulary of size D (~100,000): "aardvark … zulu"
- "bag of word": counts of each vocabulary entry
  - To marry my true love ➔ (3531:1 13788:1 19676:1)
  - I wish that I find my soulmate this year ➔ (3819:1 13448:1 19450:1 20514:1)
- Often remove stopwords: the, of, at, in, …
- Special "out-of-vocabulary" (OOV) entry catches all unknown words

More Feature Representations

- Image
  - Color histogram
- Software
  - Execution profile: the number of times each line is executed
- Bank account
  - Credit rating, balance, #deposits in last day, week, month, year, #withdrawals …
- You and me
  - Medical test1, test2, test3, …

Training Sample

- A *training sample is a collection of instances* $\mathbf{x}_1, \ldots, \mathbf{x}_n$*, which is the input to the learning process*
- $\mathbf{x}_i = (x_{i1}, \ldots, x_{iD})$
- Assume these instances are sampled independently from an unknown (population) distribution, $P(x)$
- We denote this by $\mathbf{x}_i \overset{\text{i.i.d.}}{\sim} P(x)$, where i.i.d. stands for independent and identically distributed

Training Sample

- A training sample is the "experience" given to a learning algorithm
- What the algorithm can learn from it varies
- We introduce two basic learning paradigms:
  - *unsupervised learning*
  - *supervised learning*

Unsupervised Learning

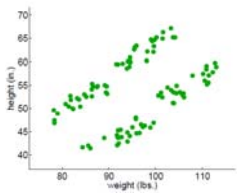
No teacher

Unsupervised Learning

- Training sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$, that's it
- No teacher providing supervision as to how individual instances should be handled
- Common tasks:
  - clustering, separate the *n* instances into groups
  - novelty detection, find instances that are very different from the rest
  - dimensionality reduction, represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training samples

Clustering

- Group training sample *into k* clusters, such that instances in the same cluster are similar, and instances in different clusters are dissimilar
- How many clusters do you see?
- Many clustering algorithms



Hierarchical Agglomerative Clustering

Input: a training sample $\{\mathbf{x}_i\}_{i=1}^n$*; a distance function* $d()$.
1. Initially, place each instance in its own cluster (called a singleton cluster).
2. ***while*** *(number of clusters* > 1*)* ***do:***
3. Find the closest cluster pair A, B*, i.e., they minimize* $d(A, B)$.
4. Merge A, B *to form a new cluster.*
Output: *a binary tree showing how clusters are gradually merged from singletons to a root cluster, which contains the whole training sample.*

- Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{s=1}^{D} (x_{is} - x_{js})^2}$.
- What about the distance between two clusters?
  - Single linkage $d(A, B) = \min_{\mathbf{x} \in A, \mathbf{x}' \in B} d(\mathbf{x}, \mathbf{x}')$.
  - Complete linkage: replace min with max
- Demo

Supervised Learning

Teacher shows labels

Label

- Little green men:
  - Predict gender (M, F) from weight, height?
  - Predict adult, juvenile from weight, height?
- A label y is the desired prediction on an instance $\mathbf{x}$
- Discrete label: classes
  - M, F; A, J: often encode as 0,1 or -1,1
  - Multiple classes: 1, 2, 3, …, C. No class order implied.
- Continuous label: e.g., blood pressure

Supervised Learning

- A labeled training sample is a collection of instances $(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_n, y_n)$
- Assume $(\mathbf{x}_i, y_i) \overset{\text{i.i.d.}}{\sim} P(x, y)$. Again, $P(x, y)$ is unknown
- Supervised learning learns a function $f: X \rightarrow Y$ in some function family F, such that $f(\mathbf{x})$ predicts the true label y on future data $\mathbf{x}$, where $(\mathbf{x}, y) \overset{\text{i.i.d.}}{\sim} P(x, y)$
  - Classification: if y discrete
  - Regression: if y continuous

Evaluation

- Training set error
  - 0-1 loss for classification: $\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) \neq y_i)$,
  - squared loss for regression $\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2$
  - overfitting
- Test set error: use a separate test set
- True error of f: $\mathbb{E}_{(\mathbf{x},y)\sim P}[c(\mathbf{x}, y, f(\mathbf{x}))]$, where c() is an appropriate loss function
- Goal of supervised learning is to find

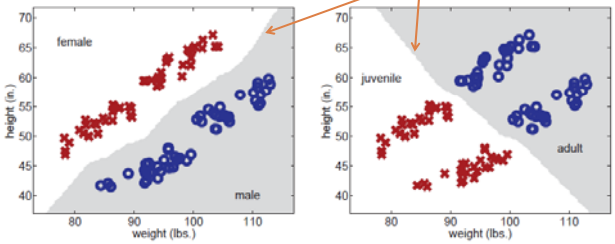
$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \mathbb{E}_{(\mathbf{x},y)\sim P}[c(\mathbf{x}, y, f(\mathbf{x}))]$$

k-Nearest-Neighbor (kNN)

Input: Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$*; distance function* $d()$*;*
number of neighbors k*; test instance* $\mathbf{x}^*$

1. *Find the* k *training instances* $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ *closest to* $\mathbf{x}^*$ *under distance* $d()$.
2. *Output* y^* *as the majority class of* $y_{i_1}, \ldots, y_{i_k}$*. Break ties randomly.*

- 1NN for little green men:



(a) classification by gender (b) classification by age

kNN

- What if we want regression?
  - Instead of majority vote, take average of neighbors' y
- How to pick *k*?
  - Split data into training and tuning sets
  - Classify tuning set with different *k*
  - Pick *k* that produces least tuning-set error

Summary

- Feature representation
- Unsupervised learning / Clustering
  - Hierarchical Agglomerative Clustering
    - Single linkage
    - Complete linkage
- Supervised learning / Classification
  - k-nearest-neighbor
  - decision trees
  - neural networks