Analytics and Visualization of Big Data

Fadel M. Megahed

Lecture 24: Machine Learning and Perceptron

AUBURN UNIVERSITY
SAMUEL GINN
COLLEGE OF ENGINEERING

Department of Industrial and Systems Engineering

Spring 13

Motivation Behind Machine Learning

- **Goal: Classify Data that will be seen in the future**

- Components of ML Algorithms:
  - Training Set
  - Classifier
  - Test Set

- **Unsupervised vs. Supervised Learning**

Source: http://youtu.be/e0WKJLovaZg?t=27s

Outline for Today's Topics – Chapter 12

The Machine Learning Model

- Training Sets
- Some Illustrative Examples
- Approaches to Machine Learning
- Machine Learning Architecture

Perceptrons

- Training a Perceptron with Zero Threshold
- Multiclass Perceptrons
- Problems with Perceptrons
- Parallel Implementation

Chapter link: http://i.stanford.edu/~ullman/pub/ch12.pdf

The Machine Learning Model – The Training Set

- The data to which ML algorithm is applied is called a **training set**.

- A training set consists of a set of pairs (x, y), where
  - x is a vector of values, often called a feature vector.
  - y is the label, the classification value for x.

- **The objective of the ML process is to discover a function $y = f(x)$ that best predicts the value of y associated with unseen values of x.**

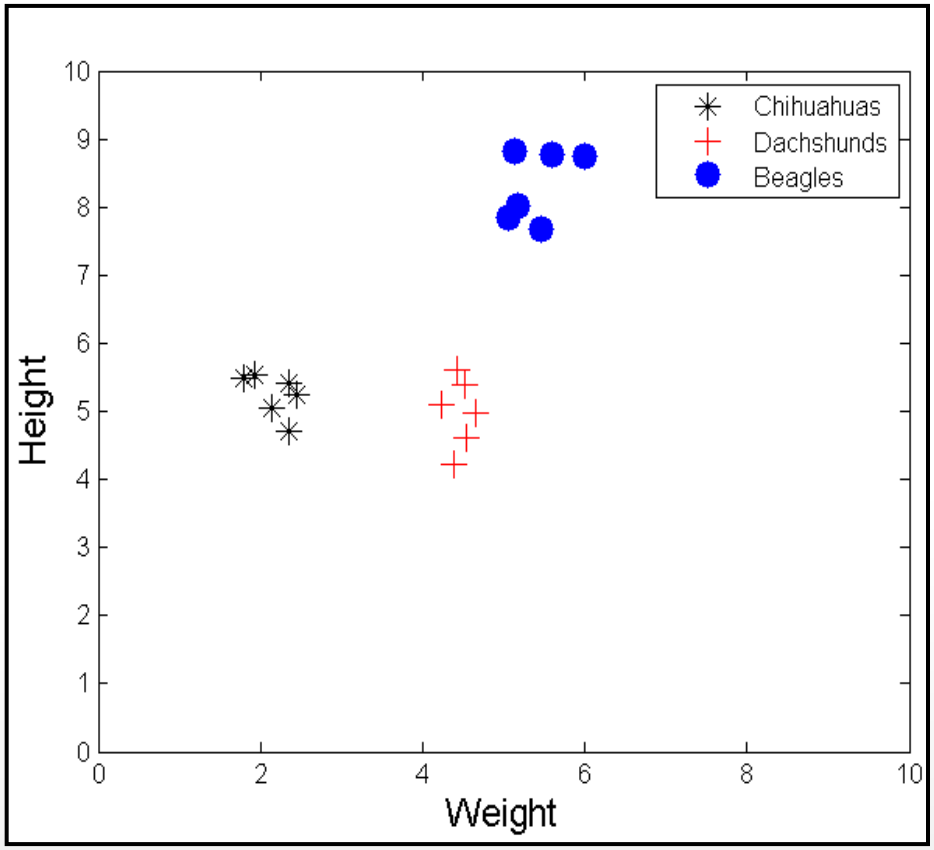
The Machine Learning Model – The Training Set (Cont.)

y can be of several types:

Real Number	→	
Boolean Value	→	Binary Classification
Member of Some Finite Set	→	Multiclass Classification
Member of Infinite Set	→	Ex: Predicting sentences

Exercise

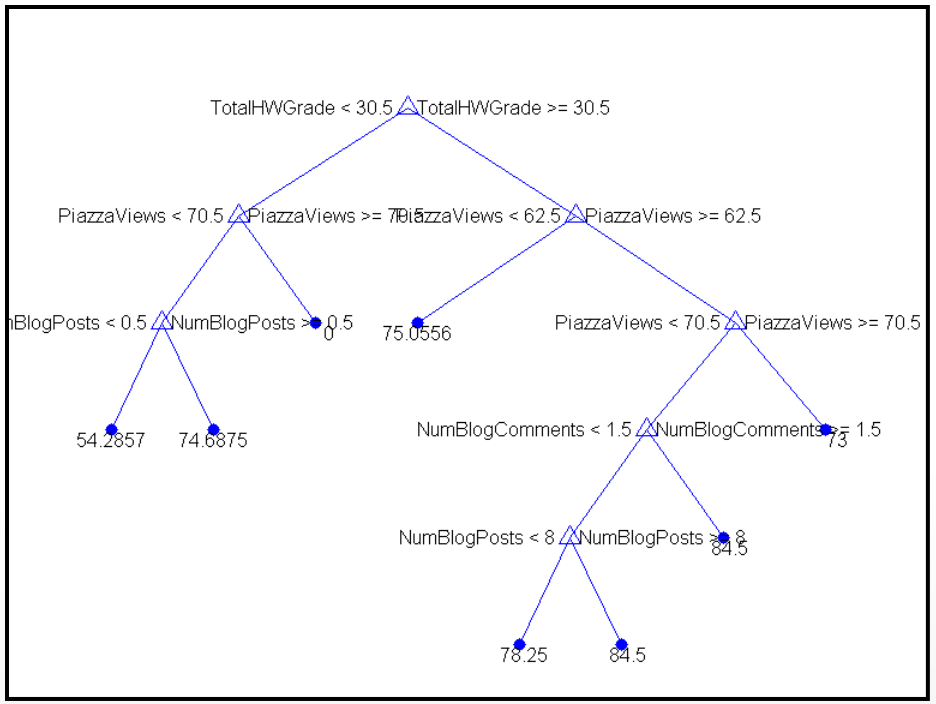
- Based on this figure, answer the following questions:
  - What *type* of y are we discussing here?
  - Develop an algorithm that would use the label and the x data to predict future dog labels based on their weight and height
  - For a [6 inch, 2.5 pounds] what is the likely dog type?
  - How is this different from clustering?



Decision Trees

- The form of f is a tree, and each node of the tree has a function of x that determines to which child or children the search must proceed.

- Decision trees are suitable for binary and multiclass classification.



Approaches to Machine Learning (Cont.)

Perceptrons:

- Threshold functions applied to the components of the vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$.

- A weight w_i is associated with the ith component, for each $i = 1, 2, \ldots, n$, and there is a threshold θ.

- The output is +1 if: $\sum_{i=1}^{n} w_i x_i \geq \theta$

  and the output is −1 otherwise.

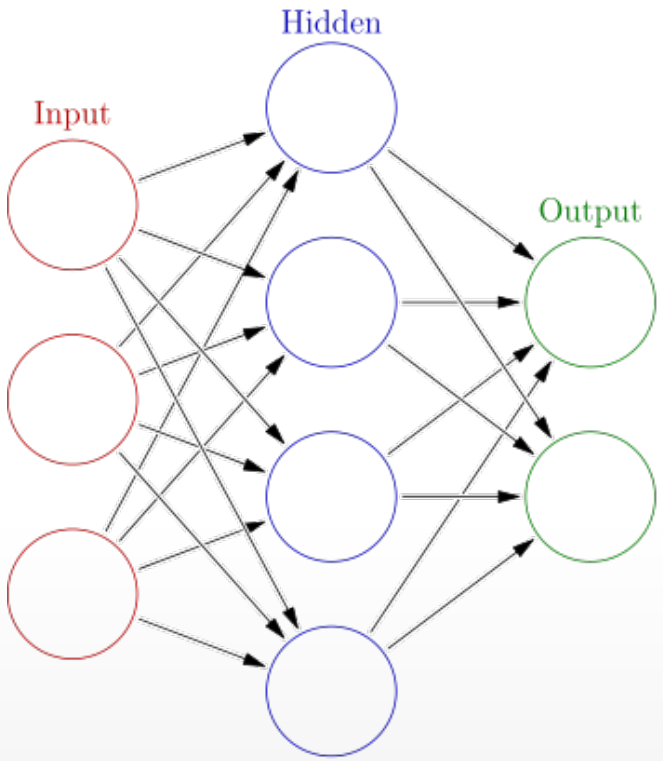
- A perceptron is suitable for binary classification, even when the number of features is very large.

Approaches to Machine Learning (Cont.)

Neural Networks

- Acyclic networks of perceptrons, with the outputs of some perceptrons used as inputs to others.

- These are suitable for binary or multiclass classification ☺

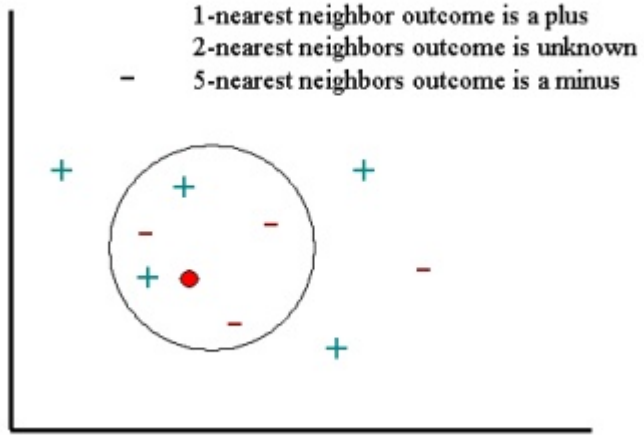


Dr. A. Smith is teaching a whole class on them next semester

Approaches to Machine Learning (Cont.)

Instance-based learning: (k-nearest-neighbor)

- Uses the entire training set to represent the function *f*.
- Appropriate for any kind of classification
  - We will concentrate on the case where *y* and the components of *x* are real numbers.



Source: http://www.statsoft.com/textbook/k-nearest-neighbors/

Approaches to Machine Learning (Cont.)

- SVM:
  - An advance over algorithms traditionally used to select weights and thresholds
  - It tends to be more accurate

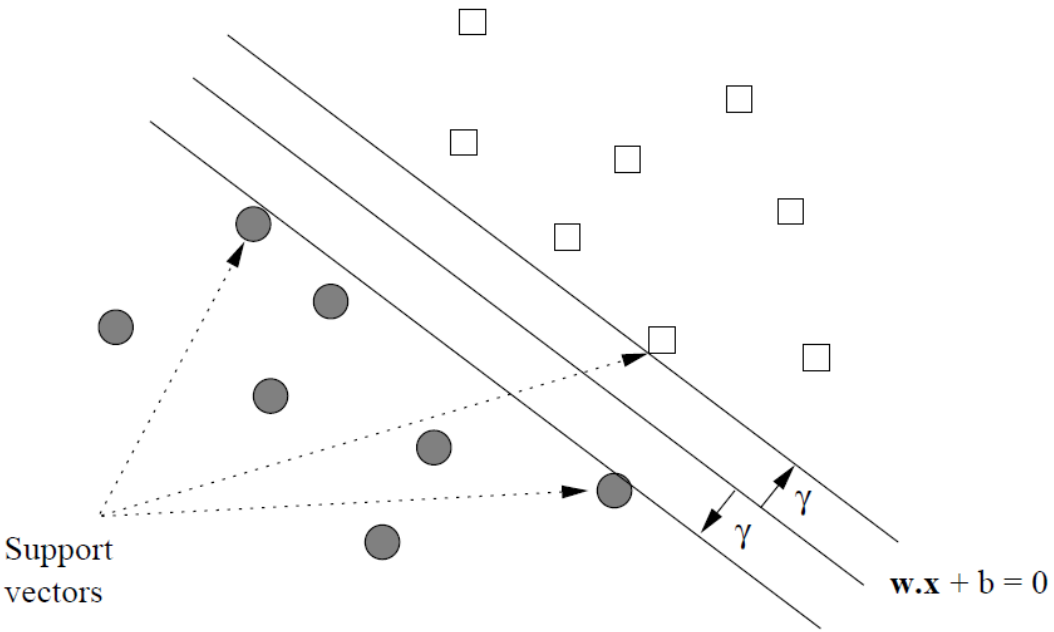


Figure 12.13: An SVM selects the hyperplane with the greatest possible margin γ between the hyperplane and the training points

Outline for Today's Topics – Chapter 12

The Machine Learning Model

- Training Sets
- Some Illustrative Examples
- Approaches to Machine Learning
- Machine Learning Architecture

Perceptrons

- Training a Perceptron with Zero Threshold
- Multiclass Perceptrons
- Problems with Perceptrons
- Parallel Implementation

Chapter link: http://i.stanford.edu/~ullman/pub/ch12.pdf

Perceptrons

- A perceptron is a linear binary classifier.

- Each perceptron has a threshold θ. The output of the perceptron is: +1 if w.x > θ, and is −1 if w.x < θ.
  - **The special case where w.x = θ will be regarded as "wrong".**

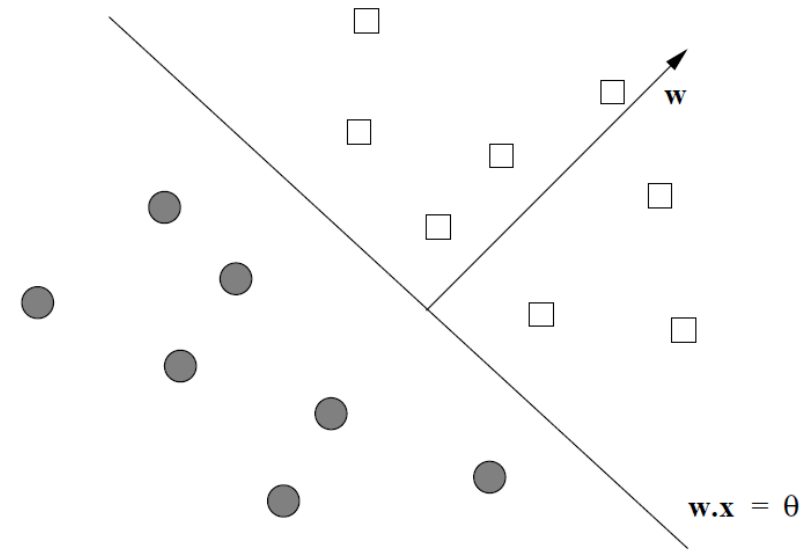


Figure 12.3: A perceptron divides a space by a hyperplane into two half-spaces

Training the Perceptron

- **Goal:** Find a weight vector **w** and a threshold θ to separate the two labels

- Steps: (Assuming that θ=0)
  - Initialize the vector **w** to all 0's.
  - Pick a learning-rate parameter *c*
    - A small, positive real number.
  - Consider each training example t = (x, y) in turn
    - Let y′ = w.x
  - If y & y′ have = sign, do nothing;
  - Else replace **w** by **w** + cy**x**.
    - Adjust **w** in **x**'s direction

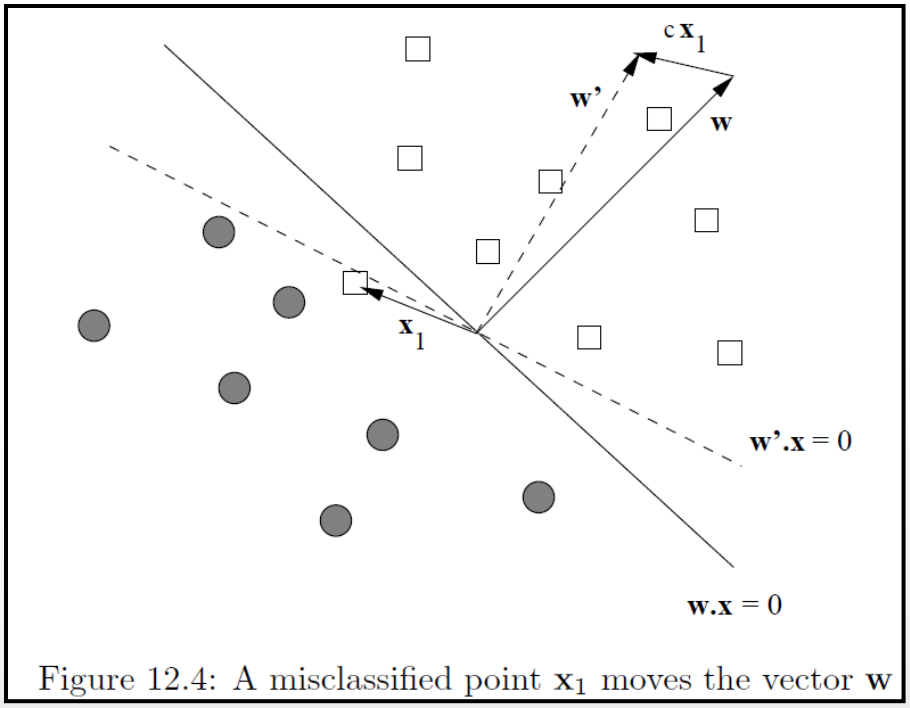


Figure 12.4: A misclassified point $\mathbf{x}_1$ moves the vector **w**

Example – Spam Detection Using Perceptron

- Let us consider training a perceptron to recognize spam email. The training set consists of pairs (x, y) where x is a vector of 0's and 1's, with each component x_i corresponding to the presence or absence of a particular word in the email.

- The value of y is +1 if the email is known to be spam and −1 if it is known not to be spam.

- Using a simplified example of five words and six examples, train the perceptron.

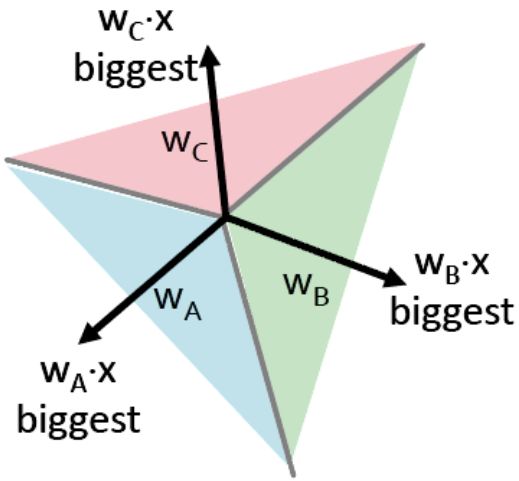
Multiclass Perceptron

- **If more than 2 classes:**
  - Weight vector w_c for each class
    - Train one class vs. the rest
      - Example: 3-way classification y = {A, B, C}
      - Train 3 classifies: w_A: A vs. B,C; w_B: B vs. A,C; w_C: C vs. A,B
  - **Calculate activation for each class**

$$f(x,c) = \Sigma_i \; w_{c,i} \; x_i \; = \; w_c \cdot x$$

  - **Highest activation wins:**

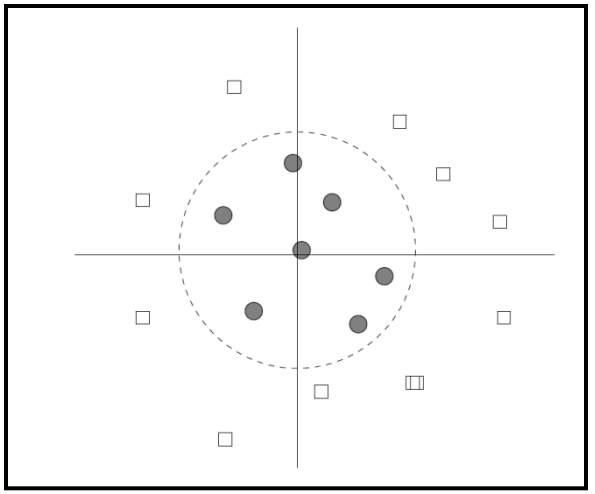
$$c = \arg\max_c f(x,c)$$

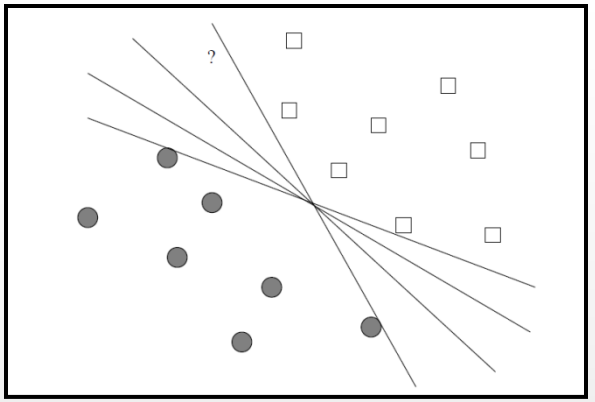


Issues with Perceptrons

- **Overfitting**
- **Regularization**

  If the data is not separable weights dance around
- **Mediocre generalization**
  - Finds a "barely" separating solution





HW 05 – Route 1 (10 Points)

- Research at least **three** articles that highlight the use of **data analytics in Web advertising**. **Write a summary (minimum of one paragraph each) on the articles you have found**.

- **Requirements:**
  - Researched articles must have a minimum of 400 words
  - You must provide a link to the researched articles (Wikipedia should not be used)
  - The total length of the submission should not exceed 500 words
  - Submit to PDF with your name and ID to fmegahed@auburn.edu

HW 05 – Route 11 (20 Points)

- Read Section 12.2.7 in the book and develop a MapReduce code that can exploit parallel implementation of Perceptrons on AWS

- **Requirements:**
  - Python fundamentals available at: http://www.codecademy.com/tracks/python
  - Validate your model with Xinyu
  - All questions should be addressed to Xinyu
  - Group work up to 4 team members is allowed