

## **CS544: Classification Algorithms**

**February 2, 2012**

Zornitsa Kozareva  
USC/ISI  
Marina del Rey, CA  
kozareva@isi.edu  
www.isi.edu/~kozareva

### Today

- Named Entity Recognition
- Multi-class classification
  - Decision trees
  - $k$  Nearest Neighbor
- Binary classification
  - Perceptron

## Named Entity Recognition

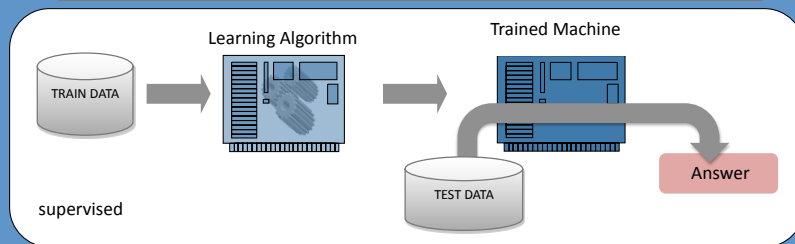
**Adam Smith** works for **IBM**, **London** since **February 2010**.

- Identify mentions in text and classify them into a predefined set of categories of interest:
  - Person: **Adam Smith**
  - Organizations: **IBM**
  - Locations: **London**
  - Date: **February 2010**

2

United States presidential election of 2008, scheduled for Tuesday November 4, 2008, will be the 56th consecutive quadrennial United States presidential election and will select the President and the Vice President of the United States. The Republican Party has chosen John McCain, the senior United States Senator from Arizona as its nominee; the Democratic Party has chosen Barack Obama, the junior United States Senator from Illinois, as its nominee.

United,0,1,0,0,1,ed,ted,Un,Uni,1,0,1,1,1,null,null,null,States,presidential,election,1,1,1,0,0



United\_B States\_I presidential\_O election\_O of\_O 2008\_O ,\_O scheduled\_O for\_O Tuesday\_O November\_O 4\_O ,\_O 2008\_O ,\_O will\_O be\_O the\_O 56<sup>th</sup>\_O consecutive\_O quadrennial\_O United\_B States\_I presidential\_O election\_O and\_O will\_O select\_O the\_O President\_B and\_O the\_O Vice\_B President\_I of\_I the\_I United\_I States\_I. The\_O Republican\_B Party\_I has\_O chosen\_O John\_B McCain\_I ,\_O the\_O senior\_O United\_B

3

## Types of Machine Learning

- Supervised Learning
  - labeled training examples with correct responses (targets) are provided
  - based on the training set, the algorithm **generalizes** to respond correctly to all possible inputs
- (Some) Methods:
  - Hidden Markov Models, k-Nearest Neighbors, Decision Trees, AdaBoost, SVM
- NLP Tasks:
  - Named Entity recognition, POS tagging, Parsing

4

## Types of Machine Learning

- Unsupervised Learning
  - correct responses (targets) are not provided
  - the algorithm identifies similarities between the inputs based on something in common
- Method:
  - Clustering
- NLP Tasks:
  - Named Entity Disambiguation, Text Categorization

5

## Types of Machine Learning

- Semi-Supervised Learning
  - small percentage of labeled examples with correct responses are provided, the rest are unlabeled
  - label the unlabeled examples using the labeled ones, add the newly labeled data to the training data set
- Method:
  - Co-training, self-training, active learning
- NLP Tasks:
  - Named Entity Recognition, POS-tagging, Parsing

6

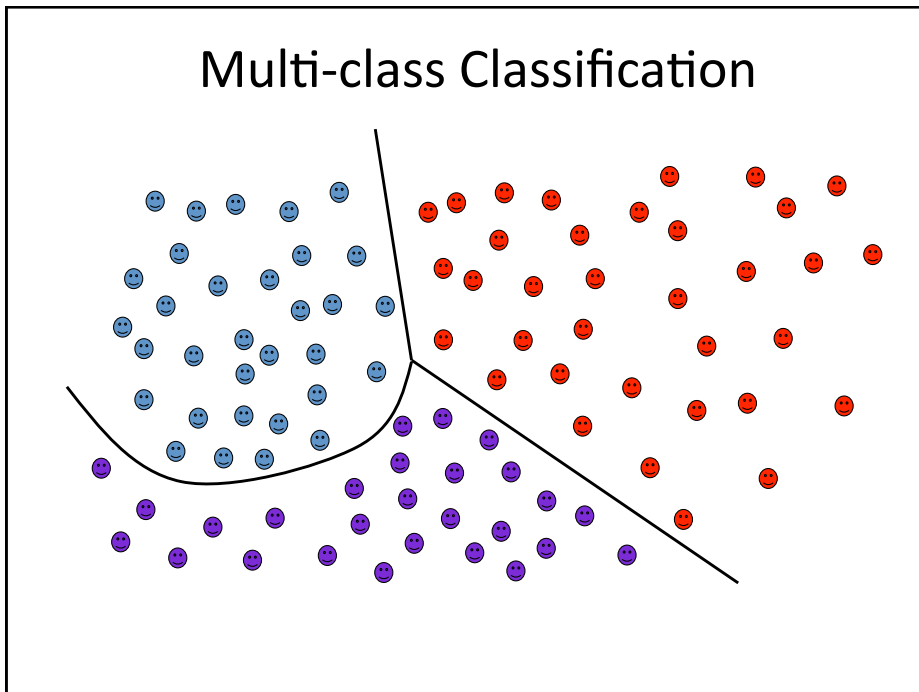
## Multi-Class Classification (Example)

- Named Entity Recognition ✓
  - person, organization, location, miscellaneous name
- Text Categorization by Topic
  - economy, sport, entertainment
- Weather Forecast
  - sunny, foggy, snowy, rainy
- Author Identification

## Multi-Class Classification

- **Given:** some data items that belong to one of  $N$  possible classes
- **Task:** train a classifier to predict the class for a new data item
- Geometrically: hard

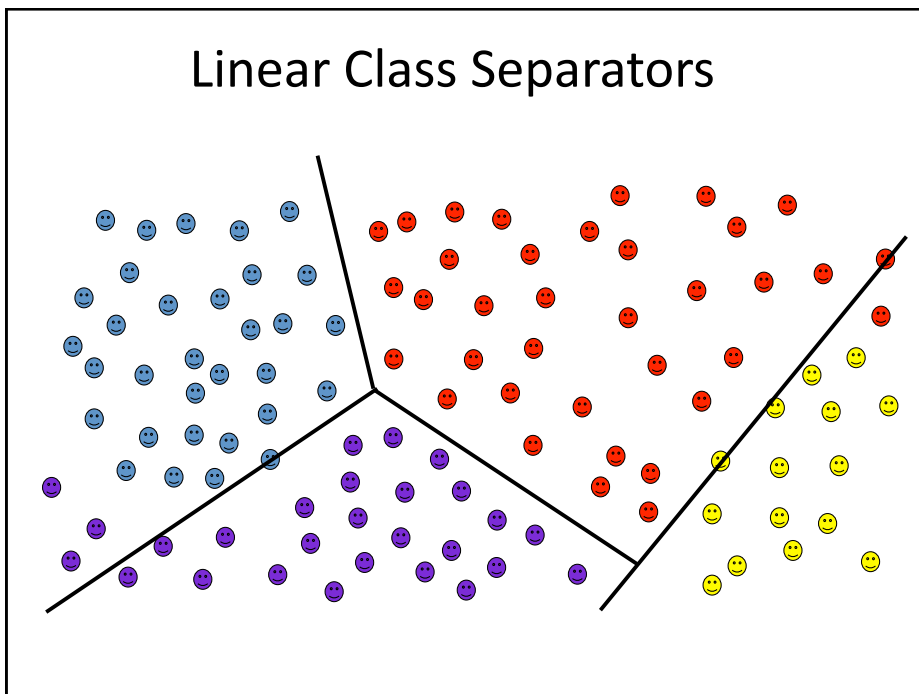
## Multi-class Classification



## (Some) Multi-class Classification Algorithms

- Linear
  - Decision trees
  - Naïve Bayes
- Non Linear
  - K-nearest neighbors
  - Neural Networks

## Linear Class Separators



## Things Students Enjoy Doing

- ✓ *going to pub*
- ✓ *watching TV*
- ✓ *going to a party*
- ✓ *Studying*

Build an algorithm that will let you decide what to do each evening without having to think about it every night?

- If you have an assignment due next day, you need to study
- If you feel lazy, the you don't like going to the pub
- If there is no party, you cannot go to it

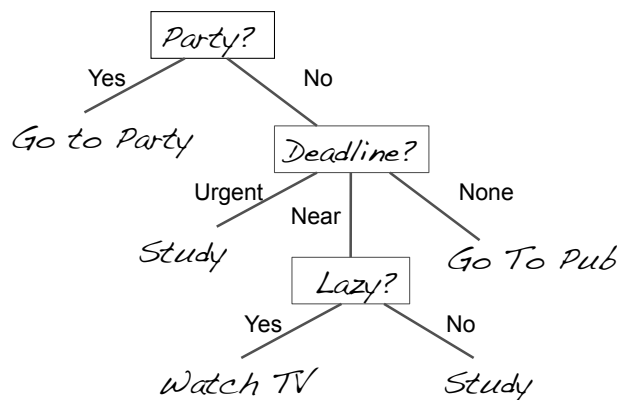
12

## Decision Trees

- The classifier has a tree structure, where each node is either:
  - a leaf indicating the value of the target attribute (class) of examples
  - a decision specifying some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test
- An instance  $x_p$  is classified by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance

13

## Decision Tree on How to Spend the Evening



14

## Constructing Decision Trees

- Build a tree in a greedy manner starting at the root
- Choose the most informative feature at each step by computing the entropy  $H(p) = -\sum_i p_i \log_2 p_i$
- Estimate how much the entropy of the whole training set would decrease if a particular feature is chosen for the next classification step

$$\text{Gain}(S, F) = \text{Entropy}(S) - \sum_{f \in \text{values}(F)} \frac{|S_f|}{|S|} \text{Entropy}(S_f)$$

*set of examples* (pointing to S)  
*possible feature* (pointing to F)  
*# of members of S that have value f for feature F* (pointing to |S<sub>f</sub>|)

15

## Walkthrough Example

Set of Examples (S)	Feature (f1)	Feature (f2)	Feature (f3)	Outcome
s1	0	1	0	True
s2	0	1	0	False
s3	0	0	1	False
s4	1	0	0	False

$$\begin{aligned}
 Entropy(S) &= -p_{true} \log_2 p_{true} - p_{false} \log_2 p_{false} \\
 &= -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} \\
 &= 0.5 + 0.311 = 0.811
 \end{aligned}$$

16

## Walkthrough Example

Set of Examples (S)	Feature (f1)	Feature (f2)	Feature (f3)	Outcome
s1	0	1	0	True
s2	0	1	0	False
s3	0	0	1	False
s4	1	0	0	False

$$\frac{|S_{f_1}|}{|S|} Entropy(S_{f_1}) = \frac{1}{4} \times \left( -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} \right) = 0$$

$$\frac{|S_{f_2}|}{|S|} Entropy(S_{f_2}) = \frac{2}{4} \times \left( -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} \right) = \frac{1}{2}$$

$$\frac{|S_{f_3}|}{|S|} Entropy(S_{f_3}) = \frac{1}{4} \times \left( -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} \right) = 0$$

*Set of examples*

$$Gain(S, F) = Entropy(S) - \sum_{f \in \text{values}(F)} \frac{|S_f|}{|S|} Entropy(S_f)$$

*# of members of S that have value f for feature F*

*possible feature*

## Walkthrough Example

Set of Examples (S)	Feature (f1)	Feature (f2)	Feature (f3)	Outcome
s1	0	1	0	True
s2	0	1	0	False
s3	0	0	1	False
s4	1	0	0	False

$$Gain(S, F) = 0.811 - (0 + 0.5 + 0) = 0.311$$

$$Gain(S, F) = Entropy(S) - \sum_{f \in \text{values}(F)} \frac{|S_f|}{|S|} Entropy(S_f)$$

*set of examples* (pointing to S)  
*possible feature* (pointing to F)  
*# of members of S that have value f for feature F* (pointing to |S<sub>f</sub>|)  
*18* (small number at the bottom right of the equation)

## Another Classification Example

- List everything that you have done for the past few days to get a decent dataset

Deadline?	Is there a party?	Lazy?	Activity
Urgent	Yes	Yes	Party
Urgent	No	Yes	Study
Near	Yes	Yes	Party
None	Yes	No	Party
None	No	Yes	Pub
None	Yes	No	Party
Near	No	No	Study
Near	No	Yes	TV
Near	Yes	Yes	Party
Urgent	No	No	Study

19

## Decision Trees

### Pros

- + generate understandable rules
- + provide a clear indication of which features are most important for classification

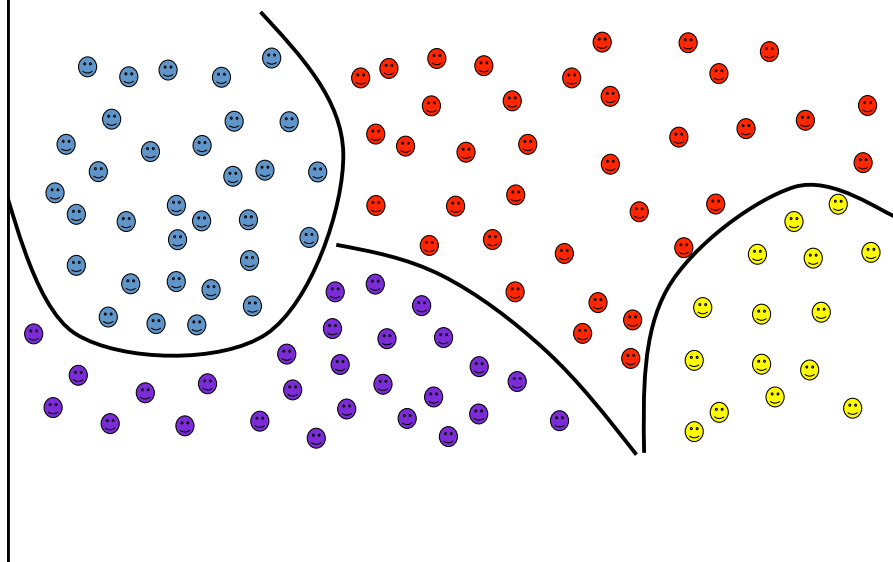
### Cons

- error prone in multi-class classification and small number of training examples
- computationally expensive to train (need to compare all possible splits; and also because of pruning)

*$O(N \log N)$  tree construction  
 $O(\log N)$  to return particular leaf*

20

## Non Linear (ex: $k$ Nearest Neighbor)

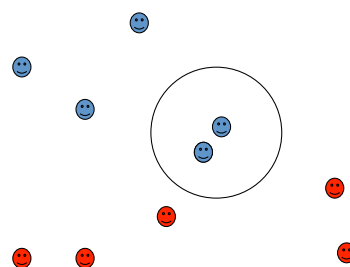


## $k$ Nearest Neighbor

- Classification rule:
  - to classify a new object, find the object in the training set that is most similar
  - then assign the class of this neighbor to the new object
- $k$  Nearest Neighbor:
  - consult  $k$  nearest neighbors
  - decision based on majority category of the neighbor

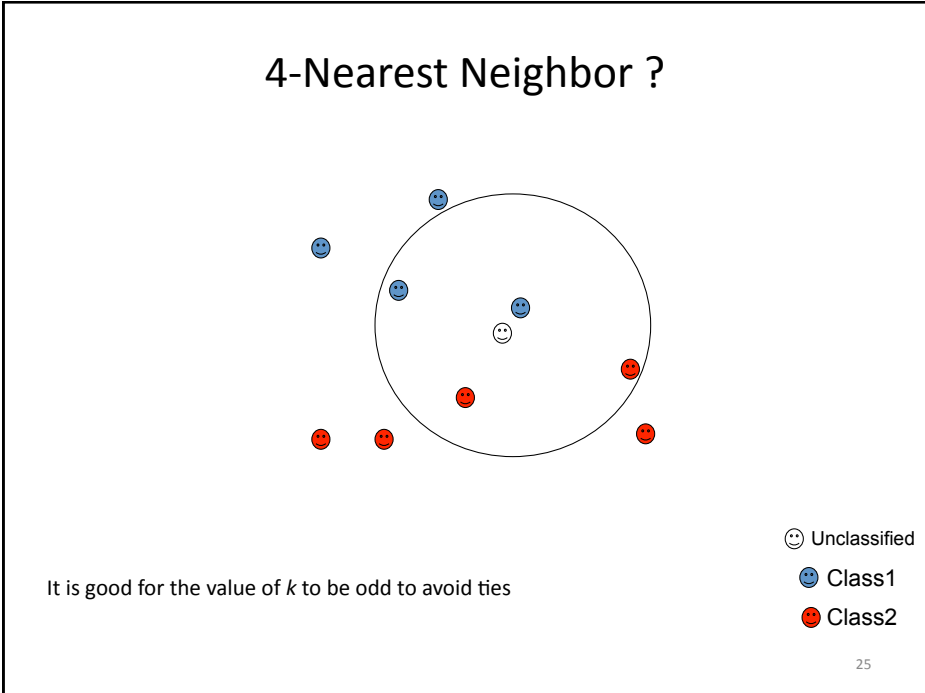
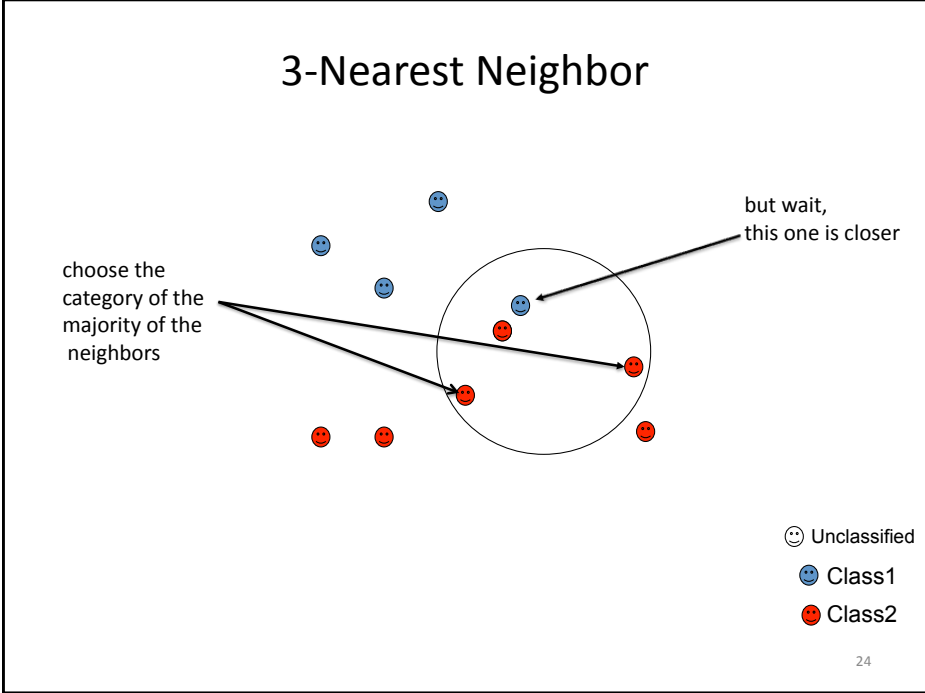
22

## 1-Nearest Neighbor



☺ Unclassified  
😊 Class1  
☹ Class2

23



## $k$ Nearest Neighbor Algorithm

- Learning is just storing the representations of the training examples.
- Testing instance  $x_p$ :
  - compute similarity between  $x_p$  and all training examples
  - take vote among  $x_p$   $k$  nearest neighbours
  - assign  $x_p$  with the category of the most similar example in  $T$

26

## Similarity Computation

- Nearest neighbor method uses similarity (or distance) metric.
- Given two objects  $x$  and  $y$  both with  $n$  values

$$x = (x_1, x_2, \dots, x_n)$$

$$y = (y_1, y_2, \dots, y_n)$$

calculate the Euclidean distance as

$$d(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$$

27

## An Example

	isPersonName	isCapitalized	isLiving	teachesCS544
Zornitsa Kozareva	1	1	1	yes
USC	0	1	0	no
eduard hovy	1	0	1	yes
and	0	0	0	no

$$d(\text{ZornitsaKozareva}, \text{USC}) = \sqrt[3]{(1^2 + 0 + 1^2)} = 1.41$$

*What does  
the score  
mean?*

$$d(\text{ZornitsaKozareva}, \text{eduardhovy}) = \sqrt[3]{(0 + 1^2 + 0)} = 1$$

$$d(\text{ZornitsaKozareva}, \text{and}) = \sqrt[3]{(1 + 1 + 1)} = 1.73$$

28

## k Nearest Neighbours

### Pros

- + robust
- + simple
- + training is very fast (storing examples)



### Cons

- depends on similarity measure & k-NNs
- easily fooled by irrelevant attributes
- computationally expensive



29

## Next Couple of Lectures

- Perceptron
- Putting Machine Learning into practice - NER
- Types of Features and Feature Generation
- Semi-Supervised Algorithms
- Introduction to Weka
- Homework #2