CS 461: Machine Learning
Lecture 2

Dr. Kiri Wagstaff
kiri.wagstaff@calstatela.edu

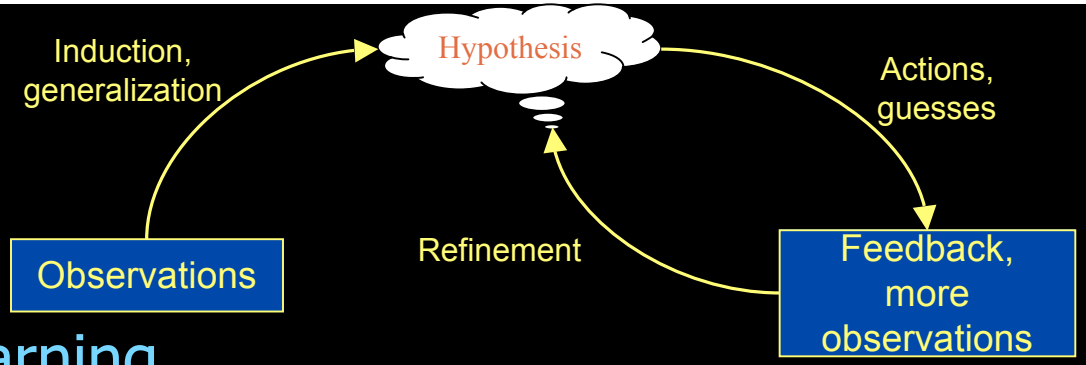
Introductions

- Share with us:
  - Your name
  - Grad or undergrad?
  - What inspired you to sign up for this class? Anything specifically that you want to learn from it?

Today's Topics

- Review
- Homework 1
- Supervised Learning Issues
- Decision Trees
- Evaluation
- Weka

Review



- Machine Learning
  - Computers learn from their past experience
- Inductive Learning
  - Generalize to new data
- Supervised Learning
  - Training data: <*x*, *g(x)*> pairs
  - Known label or output value for training data
  - Classification and regression
- Instance-Based Learning
  - 1-Nearest Neighbor
  - k-Nearest Neighbors

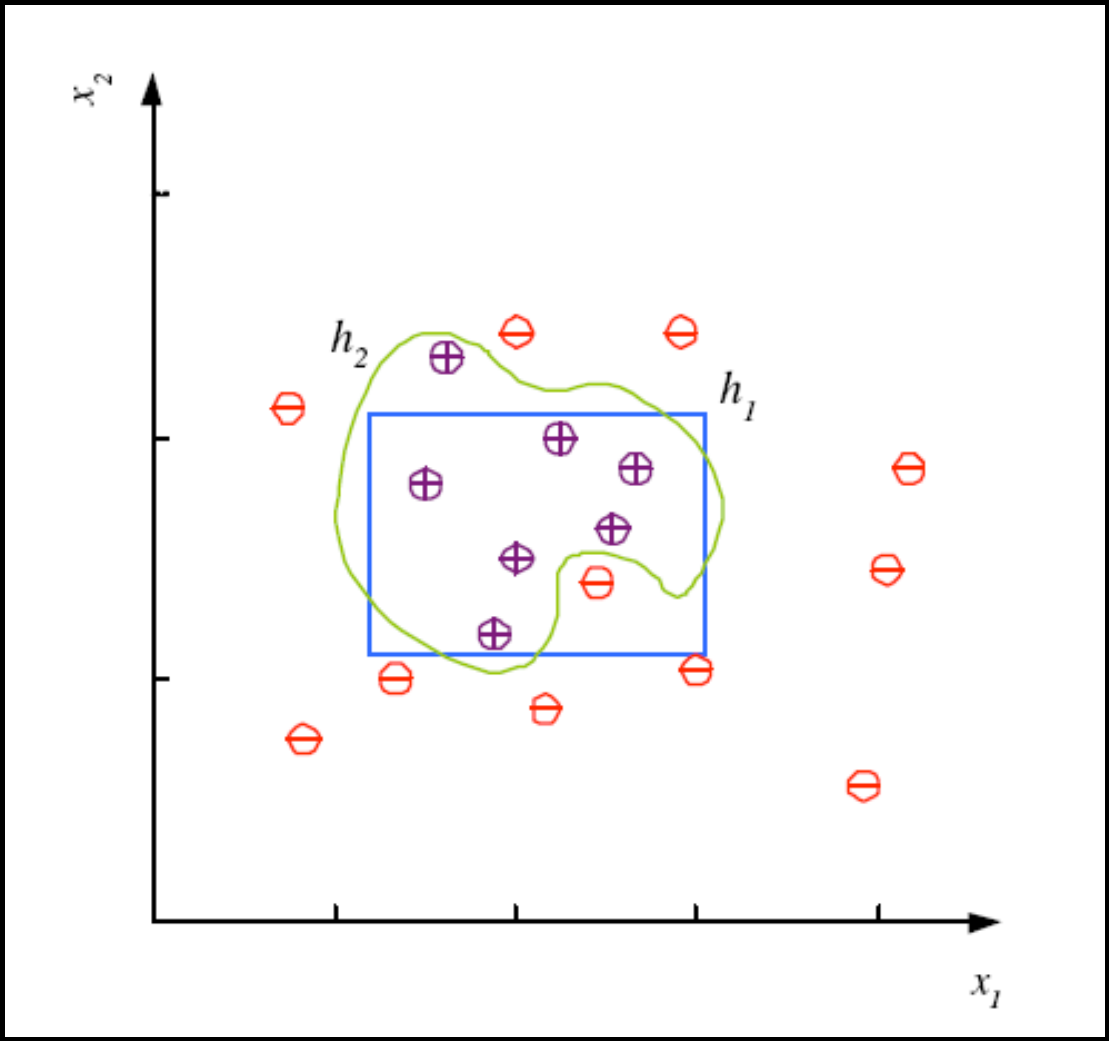
Homework 1

- Solution/Discussion

Issues in Supervised Learning

1. Representation: which features to use?
2. Model Selection: complexity, noise, bias

When don't you want zero error?



Model Selection and Complexity

- Rule of thumb: prefer simplest model that has "good enough" performance

"Make everything as simple as possible, but not simpler."
-- Albert Einstein

Another reason to prefer simple models…

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Decision Trees

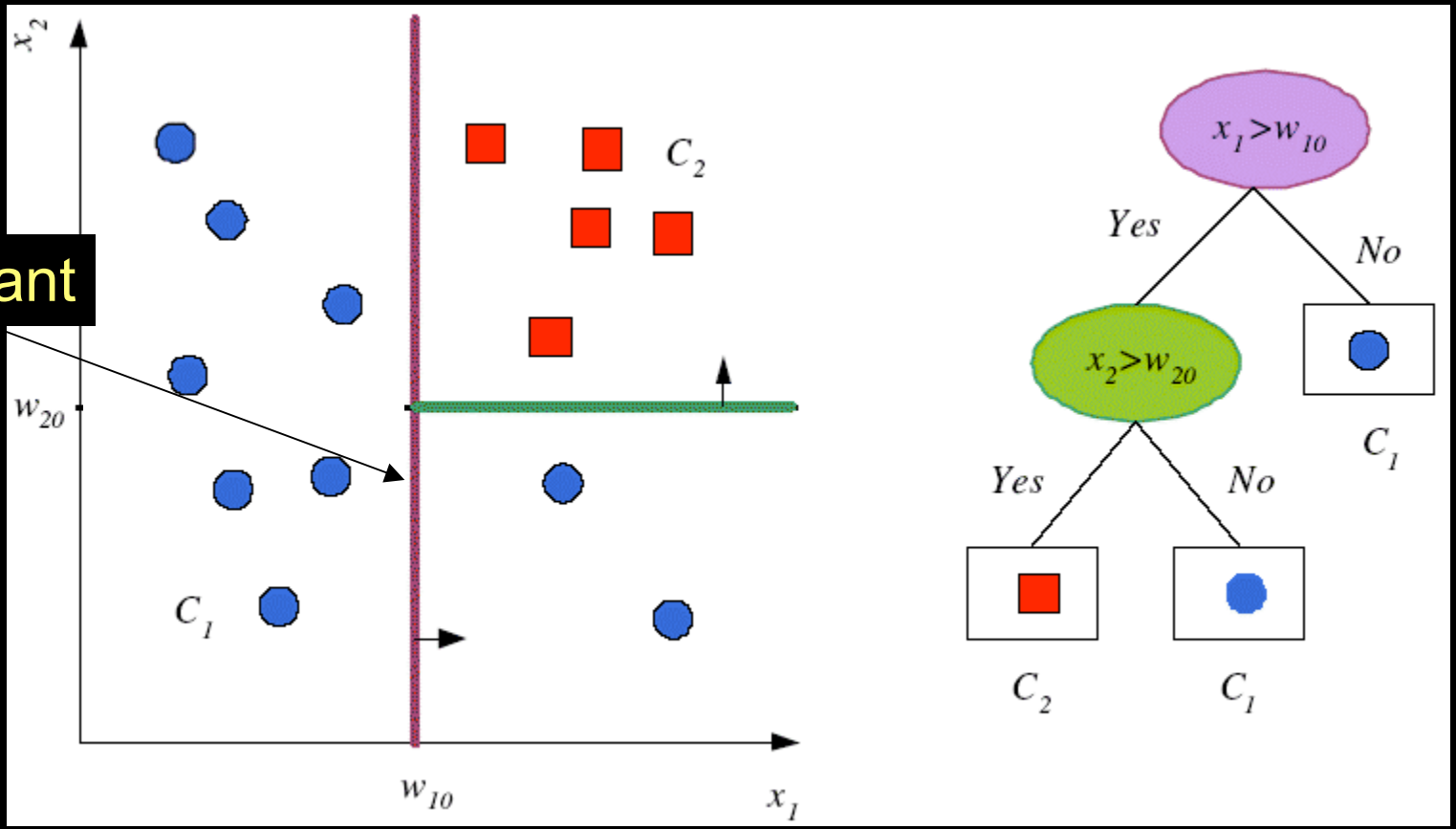
Chapter 9

Decision Trees

- Example: Diagnosis of Psychotic Disorders

(Hyper-)Rectangles == Decision Tree

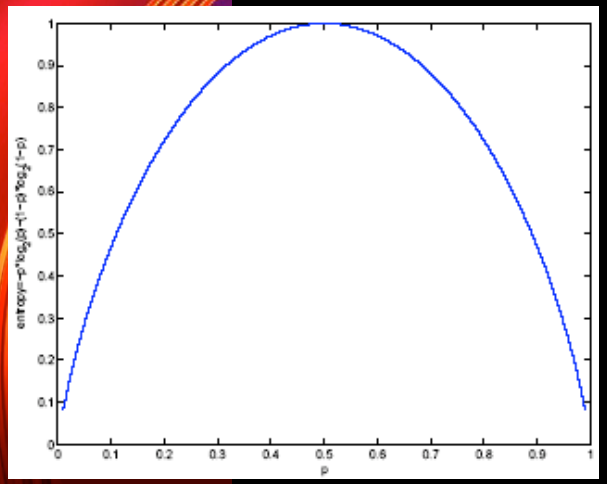
discriminant



Measuring Impurity

$$\hat{P}(C_i \mid \boldsymbol{x}, m) \equiv p_m^i = \frac{N_m^i}{N_m}$$

1. Calculate error using majority label $\mathcal{I}_m = 1 - \max_i(p_m^i)$
   - After a split $$\mathcal{I}'_m = \sum_{j=1}^{n} \frac{N_{mj}}{N_m} 1 - \max_i(p_{mj}^i)$$
2. More sensitive: use entropy
   - For node m, N_m instances reach m, N^i_m belong to C_i

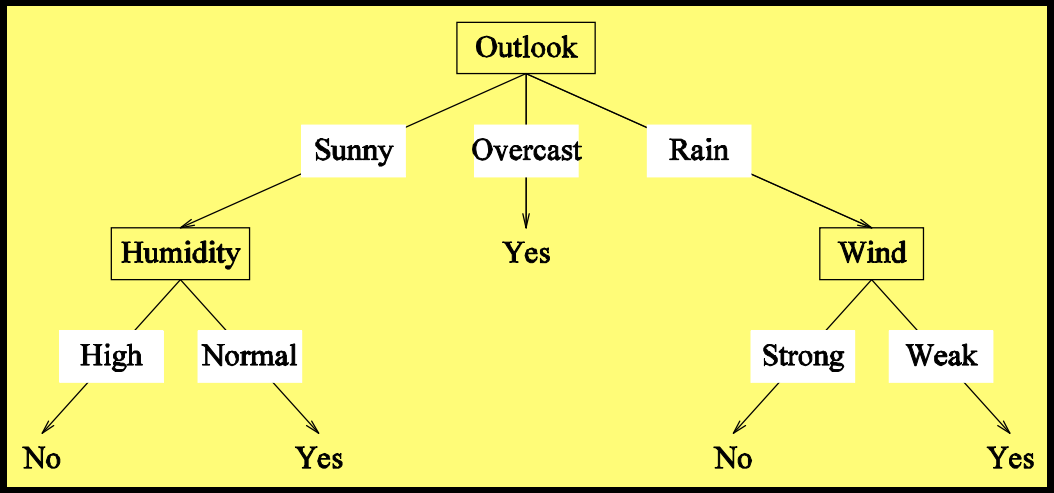


- Node m is pure if p^i_m is 0 or 1
- Entropy: $$\mathcal{I}_m = -\sum_{i=1}^{K} p_m^i \log_2 p_m^i$$
- After a split: $$\mathcal{I}'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{i=1}^{K} p_{mj}^i \log_2 p_{mj}^i$$

Should we play tennis?

Outlook	Temperature	Humidity	Wind	PlayTennis
Training Sets				
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

How well does it generalize?



Outlook	Temperature	Humidity	Wind	PlayTennis
Testing Examples				
Sunny	Mild	Normal	Weak	Yes
Overcast	Hot	High	Strong	Yes
Sunny	Mild	High	Strong	No
Rain	Hot	High	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	High	Strong	Yes

Decision Tree Construction Algorithm

```
GenerateTree(X)
    If NodeEntropy(X)< θ_I /* eq. 9.3
        Create leaf labelled by majority class in X
        Return
    i ← SplitAttribute(X)
    For each branch of x_i
        Find X_i falling in branch
        GenerateTree(X_i)
SplitAttribute(X)
    MinEnt← MAX
    For all attributes i = 1,...,d
        If x_i is discrete with n values
            Split X into X_1,...,X_n by x_i
            e ← SplitEntropy(X_1,...,X_n) /* eq. 9.8 */
            If e<MinEnt MinEnt ← e; bestf ← i
        Else /* x_i is numeric */
            For all possible splits
                Split X into X_1, X_2 on x_i
                e←SplitEntropy(X_1, X_2)
                If e<MinEnt MinEnt ← e; bestf ← i
    Return bestf
```

Decision Trees = Profit!

- PredictionWorks
  - "Increasing customer loyalty through targeted marketing"

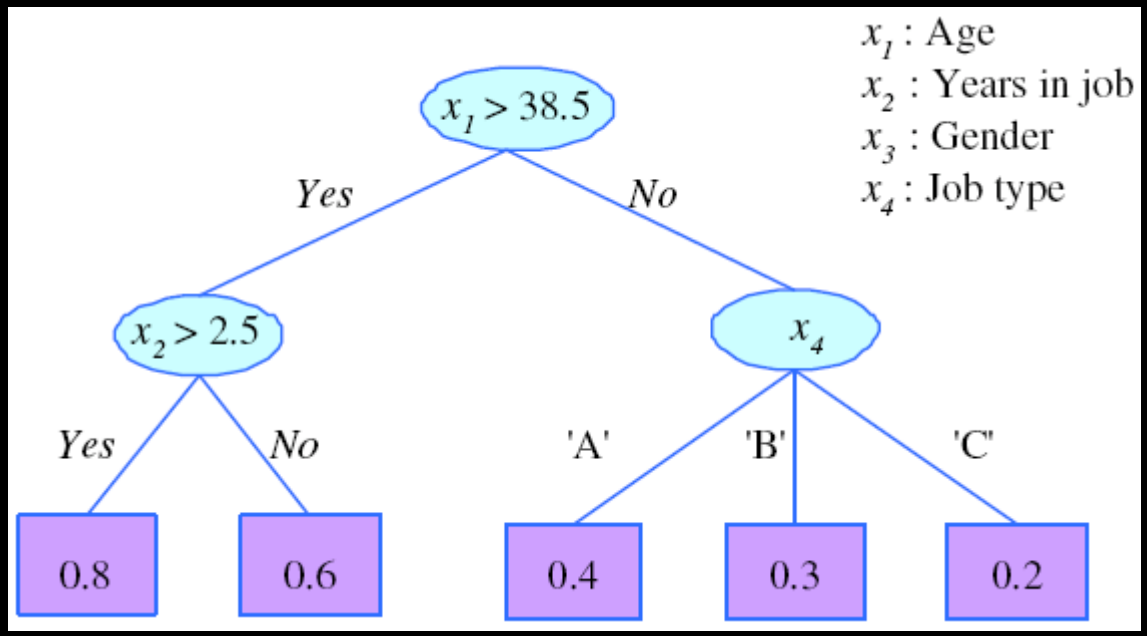
Decision Trees in Weka

- Use Explorer, run J48 on height/weight data
- Who does it misclassify?

Building a Regression Tree

- Same algorithm… different criterion
- Instead of impurity, use Mean Squared Error (in local region)
  - Predict mean output for node
  - Compute training error
  - (Same as computing the variance for the node)
- Keep splitting until node error is acceptable; then it becomes a leaf
  - Acceptable: error < threshold

Turning Trees into Rules



R1: IF (age>38.5) AND (years-in-job>2.5) THEN y =0.8
R2: IF (age>38.5) AND (years-in-job≤2.5) THEN y =0.6
R3: IF (age≤38.5) AND (job-type='A') THEN y =0.4
R4: IF (age≤38.5) AND (job-type='B') THEN y =0.3
R5: IF (age≤38.5) AND (job-type='C') THEN y =0.2

Weka Machine Learning Library

Weka Explorer's Guide

Summary: Key Points for Today

- Supervised Learning
  - Representation: features available
  - Model Selection: which hypothesis to choose?
- Decision Trees
  - Hierarchical, non-parametric, greedy
    - Nodes: test a feature value
    - Leaves: classify items (or predict values)
  - Minimize impurity (%error or entropy)
  - Turning trees into rules
- Evaluation
  - (10-fold) Cross-Validation
  - Confusion Matrix

Next Time

- Support Vector Machines
  (read Ch. 10.1-10.4, 10.6, 10.9)
- Evaluation
  (read Ch. 14.4, 14.5, 14.7, 14.9)

- Questions to answer from the reading:
  - Posted on the website (calendar)