Review Session II

Logic and Reasoning

- John likes any food
- Peanuts can be eaten.
- Anything eaten is food.

- Prove: John likes peanuts

- food(x) => likes(John,x)
- eatable(peanuts)
- eatable(x) => food(x)

{¬food(x) , likes(John,x)
{eatable(peanuts)}
{¬eatable(x) , food(x)}

Conjecture: likes(John,peanuts)
Negation: {¬likes(John,peanuts)}

{¬likes(John,peanuts)} {¬food(x) , likes(John,x) {eatable(peanuts)} {¬eatable(x) , food(x)}

{¬food(peanuts)} {food(peanuts)}

NIL

Decision Trees with Information Gain

Gray	Large	LongNeck	Class
Y	Y	N	E
N	Y	Y	G
N	Y	N	E
Y	Y	N	E

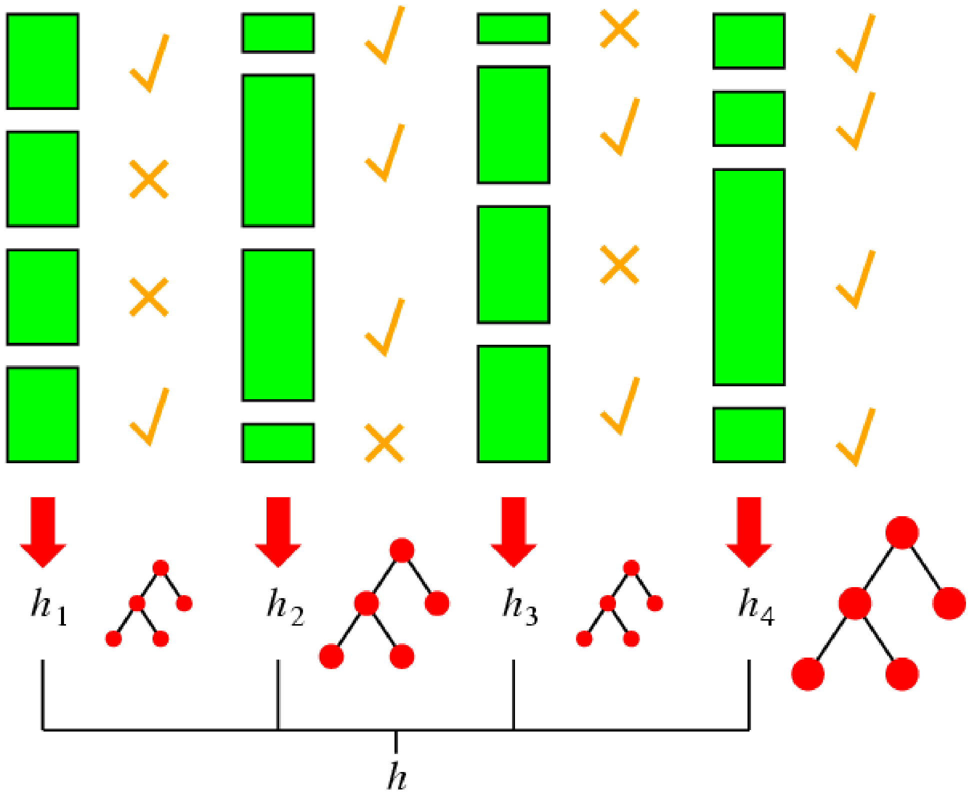
Entropy of root:
$-3/4 \log_2(3/4) - ¼ \log_2(1/4)$
$-.75(-.415) -.25(-2)$
$= .811$

Split on Gray:
Y: {E,E} Entropy 0
N: {G,E} Entropy 1
Gain: .811 -.5(0)-.5(1)
= .311

Split on Large:
Always Y
{E,G,E,E}, same as root.
Gain of Zero.

Split on LongNeck:
Y: {G} Entropy 0
N: {E,E,E} Entropy 0
Gain: .811 – 0 = .811***

Idea of Boosting



ADABoost

- ADABoost boosts the accuracy of the original learning algorithm.

- If the original learning algorithm does slightly better than 50% accuracy, ADABoost with a large enough number of classifiers is guaranteed to classify the training data perfectly.

ADABoost Weight Updating (from Fig 18.34 text)

```
/* First find the sum of the weights of the misclassified samples */
   for j = 1 to N do /* go through training samples */
      if h[m](x_j) <> y_j then error <- error + w_j

/* Now use the ratio of error to 1-error to change the
    weights of the correctly classified samples */
   for j = 1 to N do
      if h[m](x_j) = y_j then w[j] <- w[j] * error/(1-error)
```

Example

- Start with 4 samples of equal weight .25.
- Suppose 1 is misclassified. So error = .25.
- The ratio comes out .25/.75 = .33
- The correctly classified samples get weight of .25*.33 = .0825

.2500
.0825
.0825
.0825

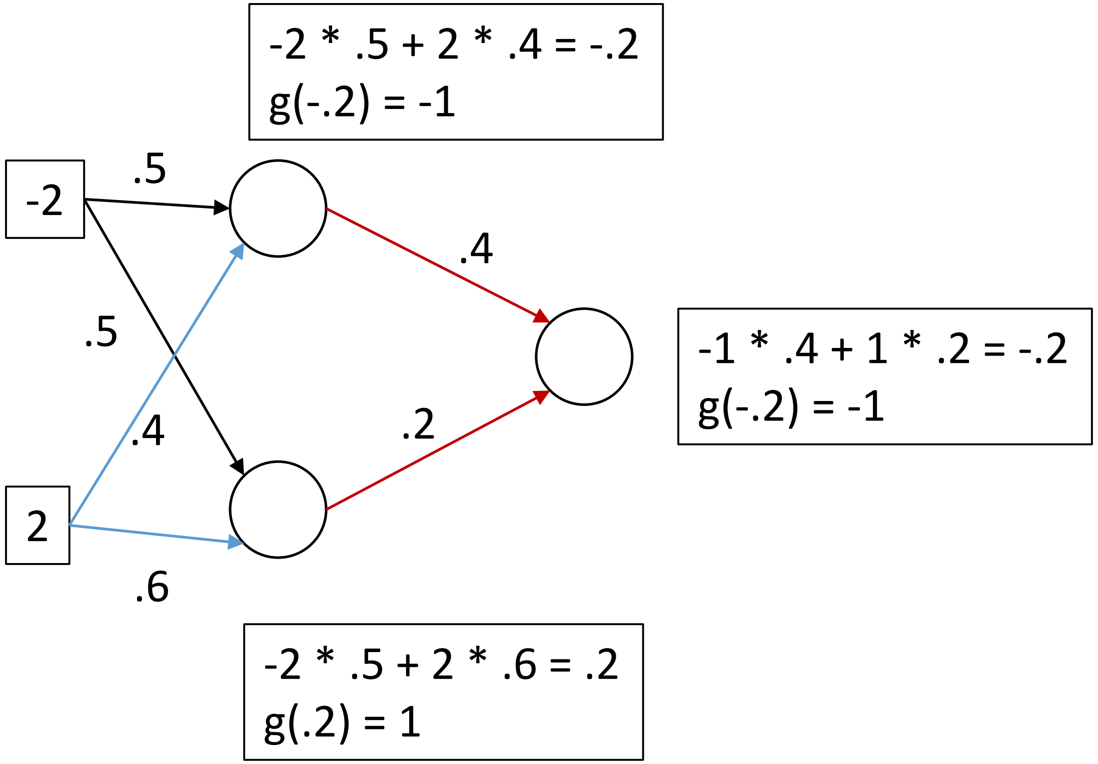
What's wrong? What should we do?

We want them to add up to 1, not .4975.

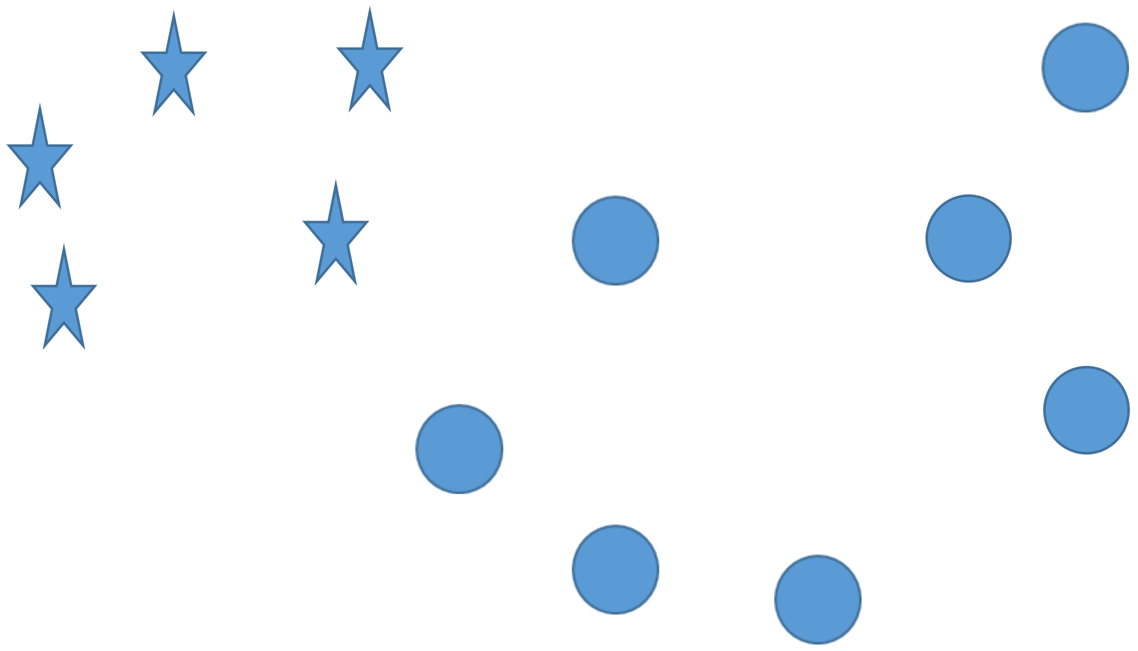
Answer: To normalize, divide each one by their sum (.4975).

.5
.165
.165
.165

Neural Nets



SVMs

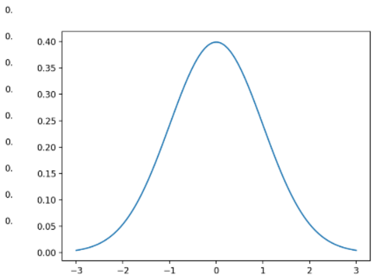


K-means: mean mu

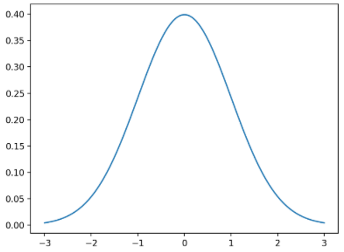
EM: mean mu, covariance ∑, weight W

Yi Li's EM Learning

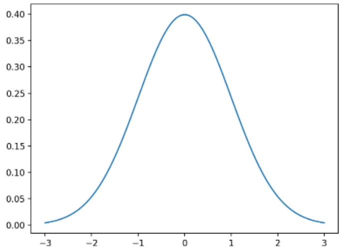
- Method 1: one Gaussian model per object class



Tree model



Stadium Model



Building Model …..

- Method 2: for each class, first use the positive instances to obtain Gaussian clusters in each feature space (color, texture, structure, etc)
- Then use the CLUSTERS to obtain fixed length feature vectors for positive and negative instances of that class and train a neural net

CNNs

- Convolution
- Pooling
- ReLU