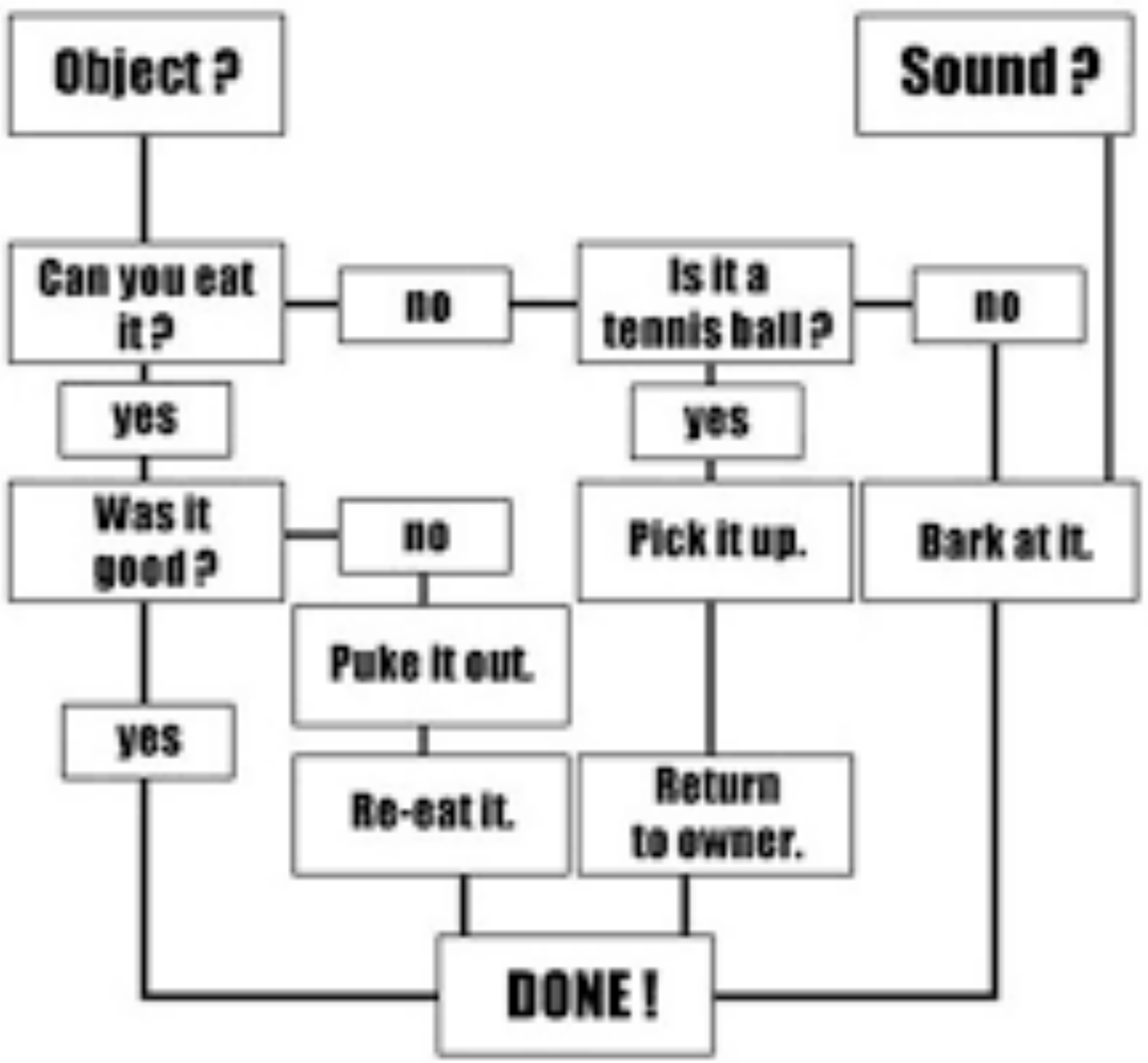
Lecture 24: Decision Trees & Random Forests

Decision tree for a dog



Outline

1. What are decision trees
2. Training decision trees
3. Pros and cons of decision trees
4. Random forests

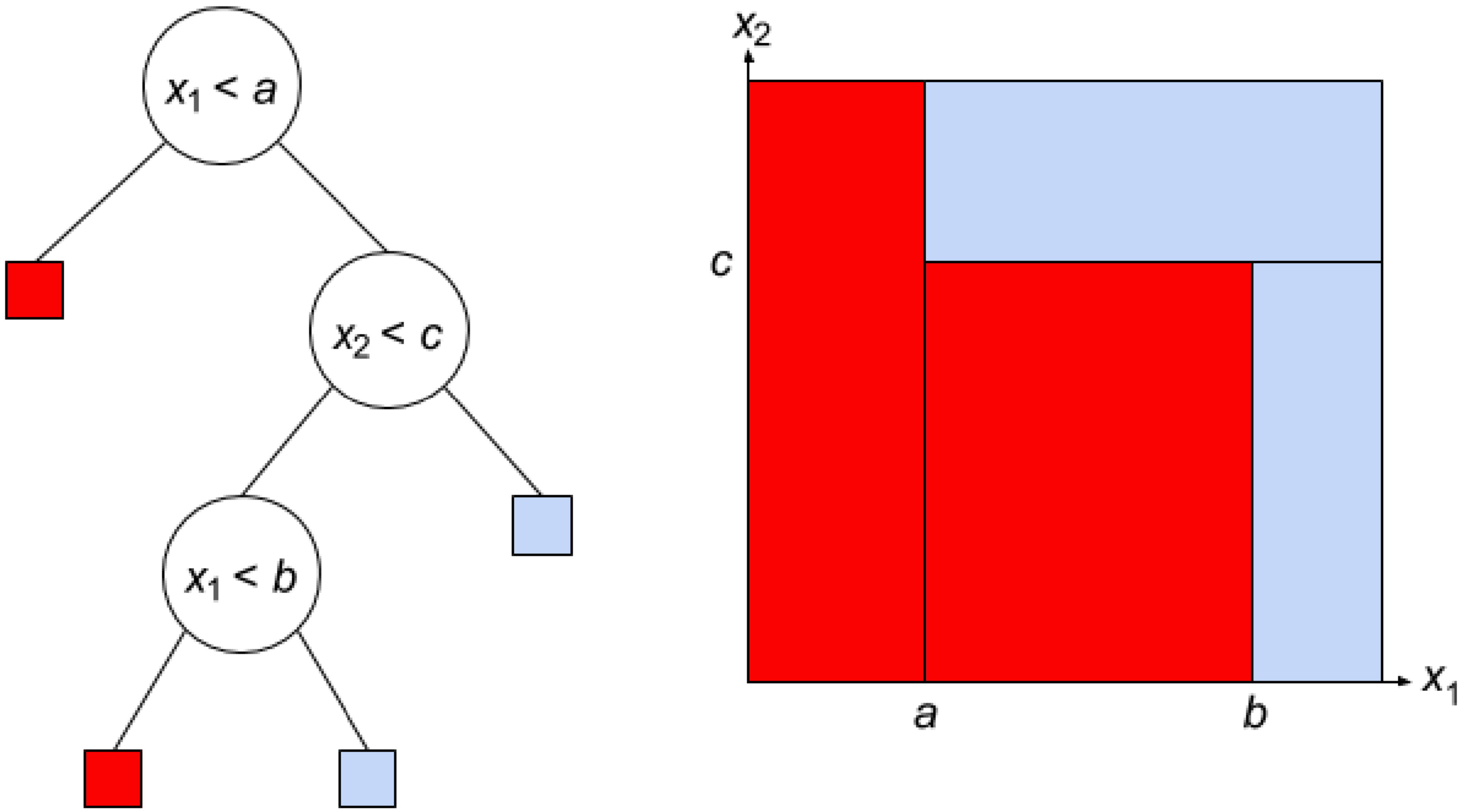
What is a decision tree?

- Make decisions (i.e., predictions)
- At each point, poses a simple series of **tests**. Process can be represented as a tree.
- Can be used for classification or regression

What kinds of decision trees will we consider here?

- Classification
- Tests: have the form "Is feature j less than value v?"
- Even these simple tests can represent arbitrarily complex classifiers

Binary decision trees



Outline

1. What are decision trees
2. **Training decision trees**
3. Pros and cons of decision trees
4. Random forests

Training decision trees

- Decision trees are "grown" recursively
- At each point on the tree, decide whether to split or predict
- If you're going to split, you have to choose *where* to split (i.e., which feature and at which value)
- Choosing splits: consider all splits. Pick the one that is best according to some criterion

Criteria for choosing splits

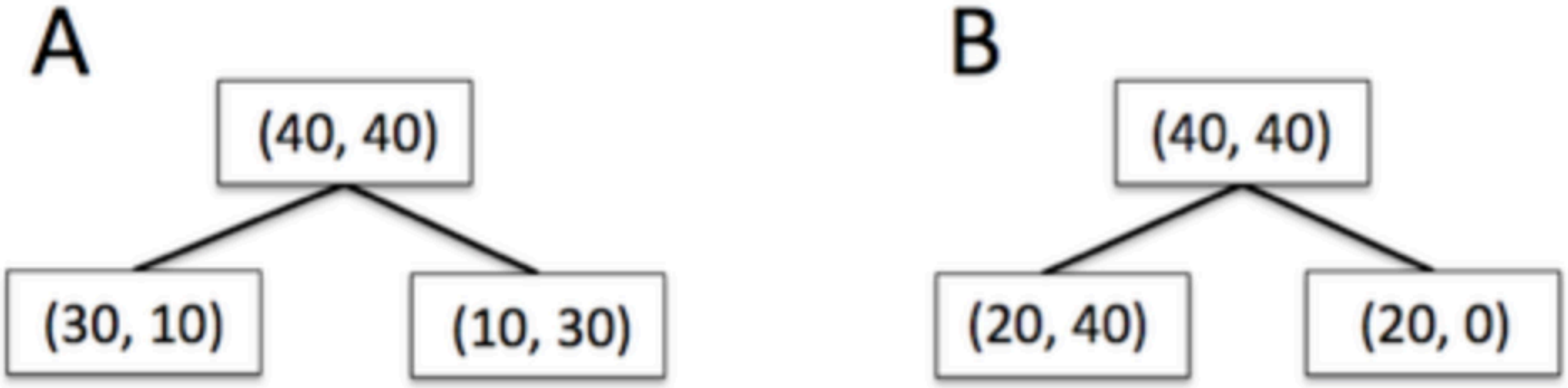
A. Misclassification rate

Misclassification rate

$$M(Y) = 1 - \max_k P(Y = k)$$

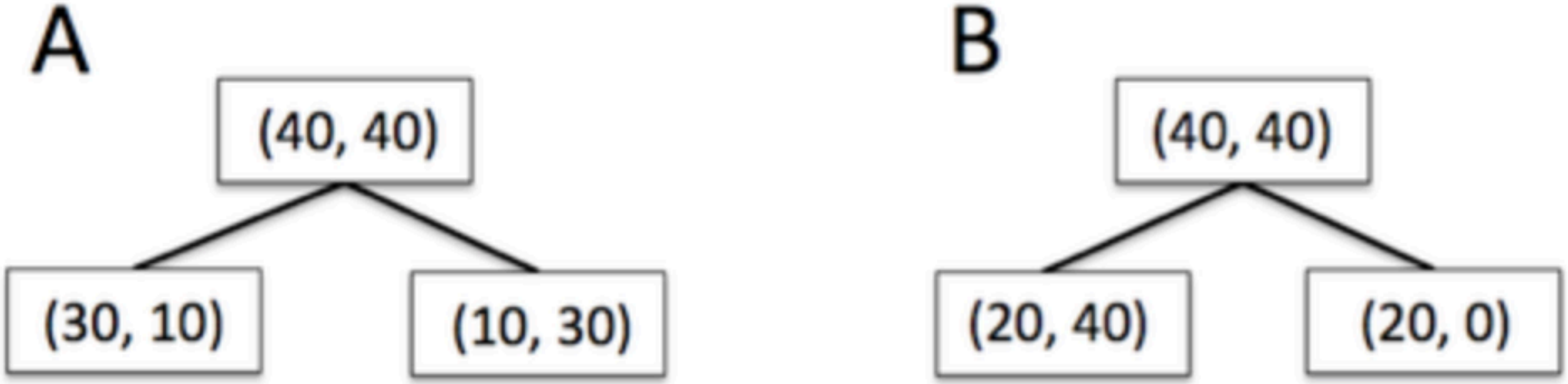
Choose the split that leads to the **biggest reduction** in the misclassification rate

What's wrong with the misclassification rate?



- $M_A(Y) = 0.5 * (1 - 0.75) + 0.5 * (1 - 0.75) = 0.25$
- $M_B(Y) = (3/4) * (1 - (2/3)) + 0.25 * (1 - 1) = 0.25$

What's wrong with the misclassification rate?



- Both splits produce the same reduction in the misclassification rate
- However, the second one is better in the sense that it can completely identify some of the points

Intuition for how to split points

Choose the split that most **reduces the uncertainty** about which class points belong to which classes

Criteria for choosing splits

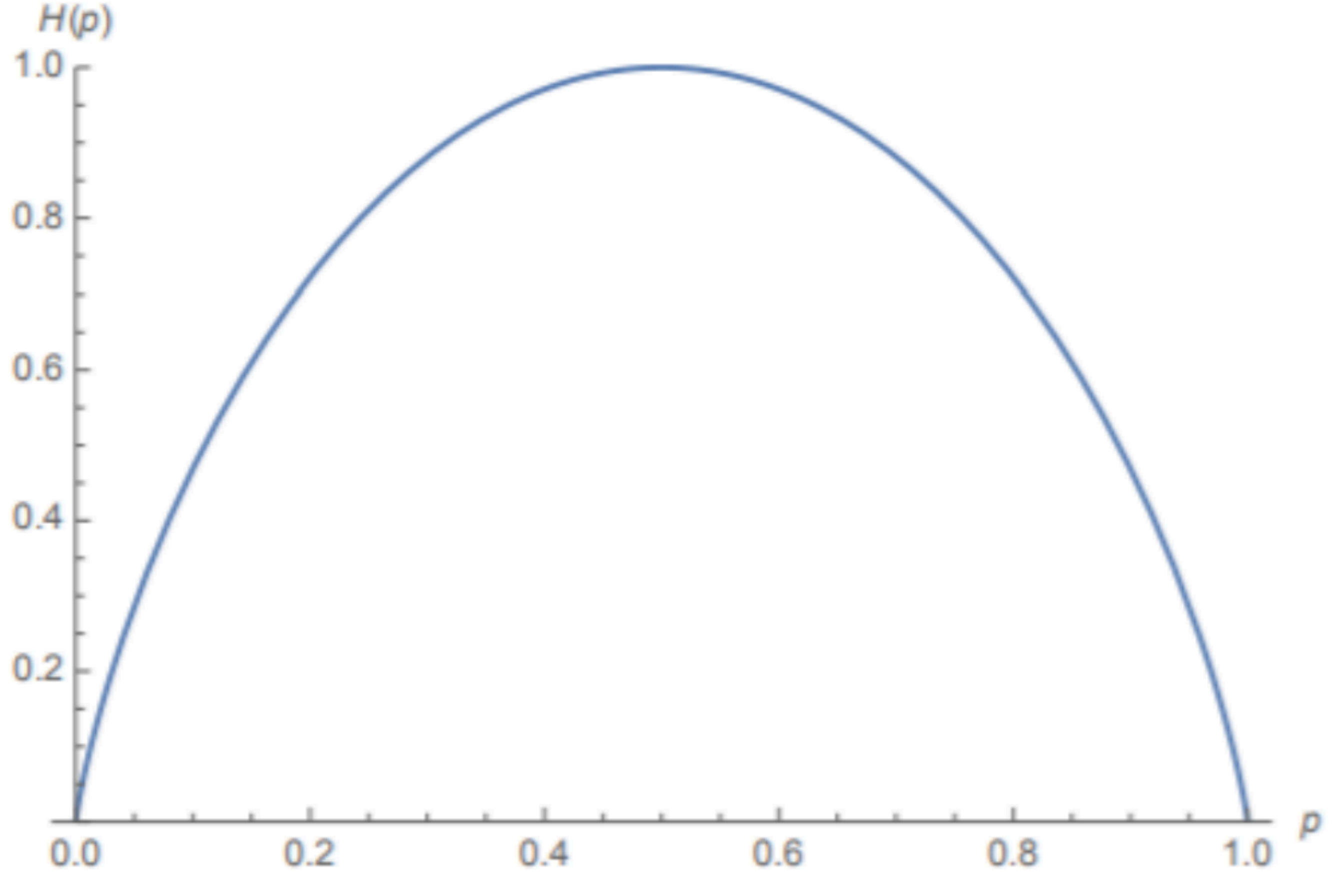
A. Misclassification rate
B. Conditional entropy

Surprise & entropy

- **Surprise**
  - Quantity that goes to $\inf$ for less likely values of the random variable
  - $-\log P(Y = k)$
- **Entropy**
  - Expected surprise
  - Measure of how uncertain your R.V. is. More likely you're surprised, more uncertain, higher entropy

$$H(Y) = -\sum_k P(Y = k) \log P(Y = k)$$

Entropy of Bernoulli random varaible



- General rule: closer to uniform = more entropy
- More skewed = less entropy

How to apply entropy to our fixed dataset

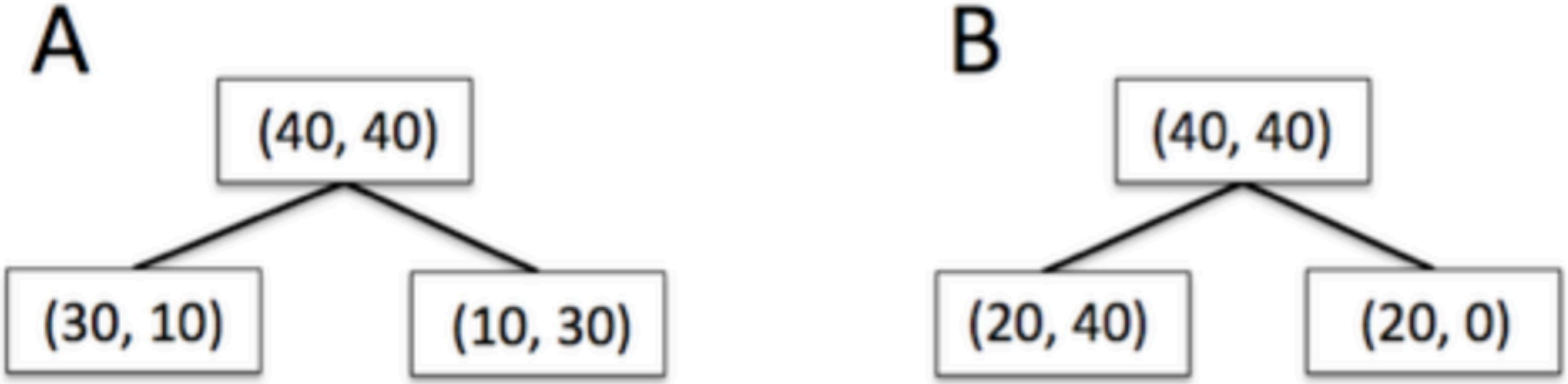
Empirical distribution: discrete r.v. with probability of class k:

$$P(Y = k) = \frac{\#\{y_i = k\}}{n}$$

Applying entropy to choosing splits

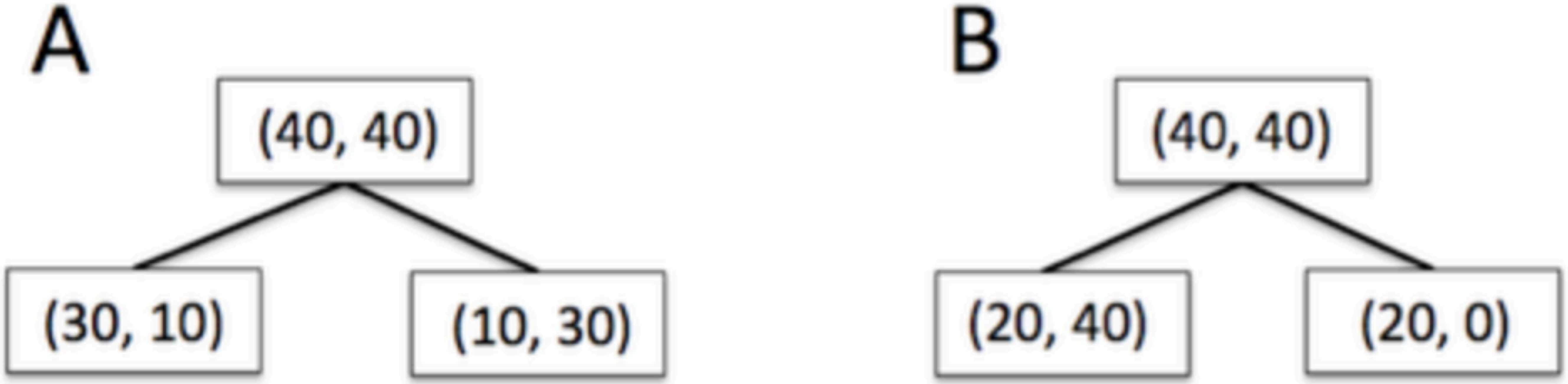
- Maximize the **reduction in entropy** when choosing a given split
- Entropy after the split: **conditional entropy**
- $$H(Y|X_{j,v}) = p(X_j \geq v)H(Y|X_j \geq v)s + p(X_j < v)H(Y|X_j < v)$$

Back to our example



- $H(Y|X_A) = 0.5 * H(Y|X_A = 0) + 0.5 * H(Y|X_A = 1)$
- $H(Y|X_A = 0) = H(Y|X_A = 1) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.81$
- $H(Y|X_A) = 0.81$

Back to our example



- $H(Y|X_B) = 0.75 * H(Y|X_A = 0) + 0.25 * H(Y|X_A = 1)$
- $H(Y|X_B = 0) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 0.92$
- $H(Y|X_B = 1) = -0 - 1.0 * log(1) = 0 \implies H(Y|X_B) = 0.75 * 0.92 = 0.69$

Criteria for choosing splits

A. Misclassification rate
B. Conditional entropy
C. Gini impurity

Gini impurity

- How often would a randomly chosen element from the set be labeled incorrectly?
- $G(Y) = \sum_k P(Y = k) \sum_{j \neq k} P(Y = j) = 1 - \sum_k P(Y = k)^2$
- More computationally efficient than entropy, hence used more in practice

Training decision trees

- Decision trees are "grown" recursively
- At each point on the tree, decide whether to split or predict
- If you're going to split, you have to choose *where* to split (i.e., which feature and at which value)
- Choosing splits: consider all splits. Pick the one that is best according to some criterion

Deciding when to stop splitting

- **Limited depth**: don't split if the node is beyond some fixed depth
- **Node purity**: don't split if nearly all points in the node are of a given class
- **Information gain**: don't split if the information gain / Gini purity are close to zero (i.e., no difference in entropy or Gini impurity)

Decision tree pruning

- Try recombining splits
- If validation error goes down, keep the recombination!

Outline

1. What are decision trees
2. Training decision trees
3. **Pros and cons of decision trees**
4. Random forests

Pros and cons of decision trees

Pros

- Highly interpretable
- Can represent any decision boundary

Cons

- Prone to overfitting

Outline

1. What are decision trees
2. Training decision trees
3. Pros and cons of decision trees
4. **Random forests**

Random forests

Key idea

- Ensemble many decision trees to produce a prediction that has lower variance

Constructing random forests

- Start by finding n random decision trees for your problem
- How to randomize the decision tree process?
  - **bagging** (bootstrap aggregating): train each on a randomly sampled subset of your data
  - **feature randomization**: train each on a randomly sampled subset of features
- Each of the models gets 1 vote in the class chosen

Conclusion

- Random trees are a low-bias highly interpretable way of performing classification
- However, they are high variance
- To reduce the variance, you should use stopping criteria, pruning, and random forests