Decision Tree Instability and Active Learning

Kenneth Dwyer and Robert Holte

University of Alberta

November 14, 2007

Instability and Decision Tree Induction

Quantifying Stability

Instability in Active Learning

Experiments

Results

Conclusions and Future Work

What is Learner Instability?

Definition

A learning algorithm is said to be **unstable** if it is sensitive to **small changes** in the training data

What is Learner Instability?

Definition

A learning algorithm is said to be **unstable** if it is sensitive to **small changes** in the training data

Problems caused by instability

- Estimates of predictive accuracy can exhibit high variance
- Difficult to extract knowledge from the model; or the knowledge that is obtained may be unreliable

What is Learner Instability?

Example

Understanding low yield in a manufacturing process:

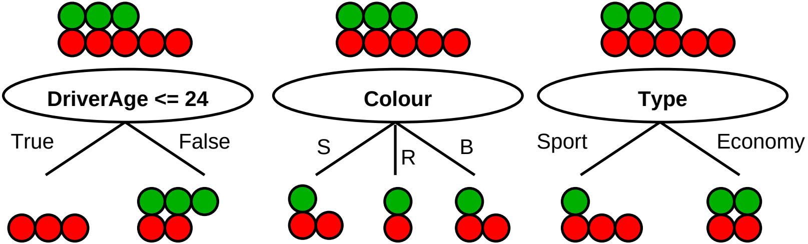
- "The engineers frequently have good reasons for believing that the causes of low yield are relatively constant over time. Therefore the engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees, even when we can demonstrate that the trees have high predictive accuracy." [Turney, 1995]

Review: Decision Tree Induction

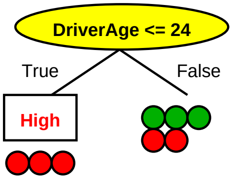
- Using the C4.5 decision tree software [Quinlan, 1996]
- Task: Given a collection of **labelled** examples, build a decision tree that accurately predicts the class labels of **unseen** examples

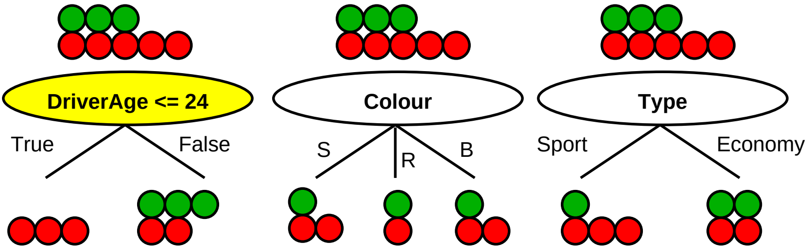
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low

Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low

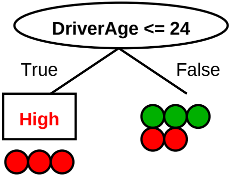


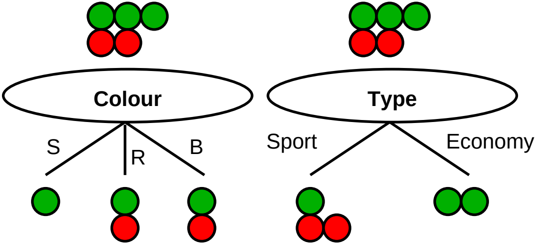
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low



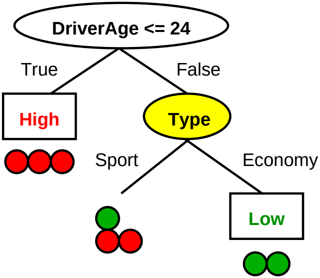


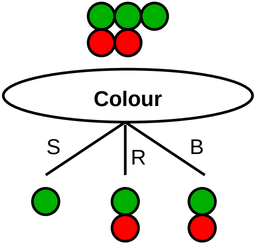
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low

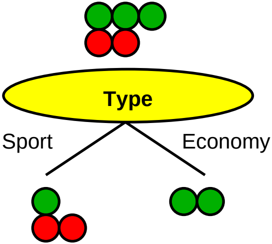




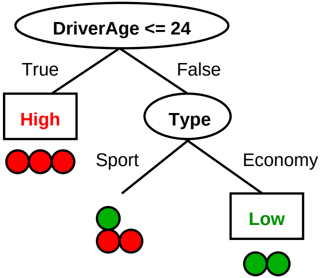
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low



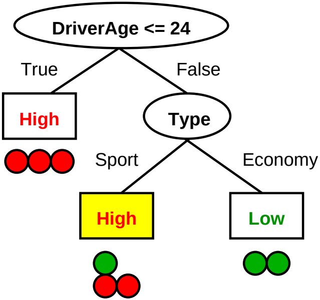


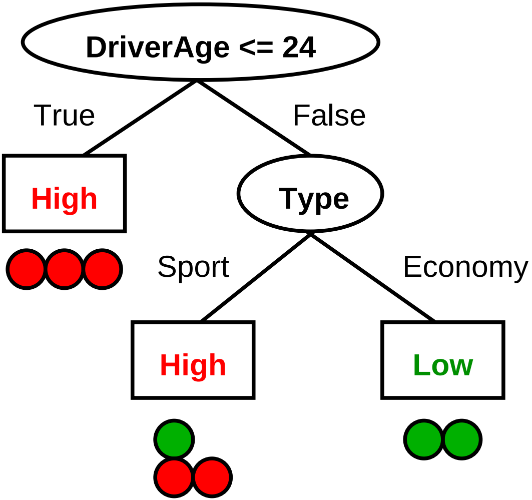


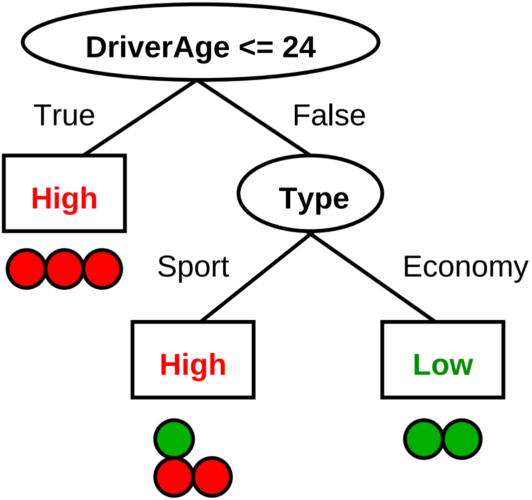
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low



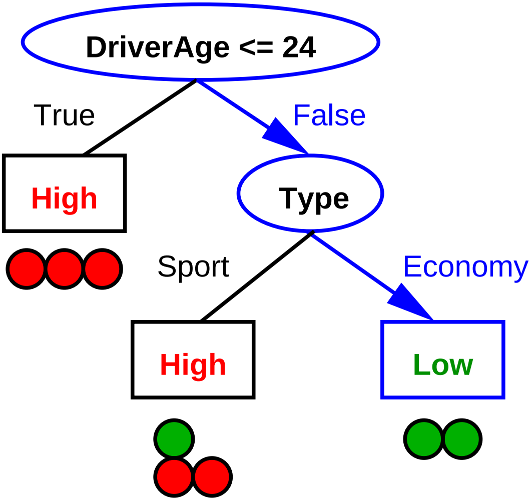
Type	Colour	DriverAge	Risk
Sport	Silver	24	High
Sport	Red	37	High
Economy	Black	19	High
Economy	Silver	21	High
Sport	Black	39	High
Sport	Silver	46	Low
Economy	Black	62	Low
Economy	Red	26	Low




DriverAge <= 24
True
False
High
Type
Sport
Economy
High
Low



- Classify an unseen example:
  - DriverAge=**32**, Type=**Economy**, Colour=**Black**



- Classify an unseen example:
  - DriverAge=**32**, Type=**Economy**, Colour=**Black**

Decision Tree Splitting Criteria

- The best attribute and split at a given node are determined by a **splitting criterion**
- Each criterion is defined by an impurity function $f(p_+, p_-)$
  - Here, p_+ and p_- represent the probabilities of each class within a given subset of examples formed by the split

Decision Tree Splitting Criteria

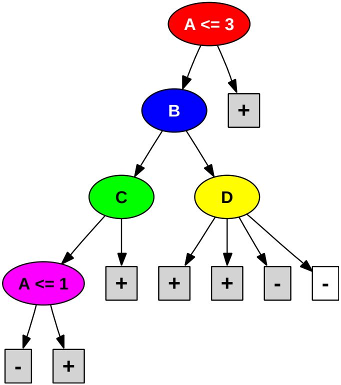
- The best attribute and split at a given node are determined by a **splitting criterion**
- Each criterion is defined by an impurity function $f(p_+, p_-)$
  - Here, p_+ and p_- represent the probabilities of each class within a given subset of examples formed by the split
- C4.5 uses an **entropy**-based criterion (i.e. gain ratio)
  - $f(p_+, p_-) = (p_+)\log_2(p_+) + (p_-)\log_2(p_-)$

Decision Tree Splitting Criteria

- The best attribute and split at a given node are determined by a **splitting criterion**
- Each criterion is defined by an impurity function $f(p_+, p_-)$
  - Here, p_+ and p_- represent the probabilities of each class within a given subset of examples formed by the split
- C4.5 uses an **entropy**-based criterion (i.e. gain ratio)
  - $f(p_+, p_-) = (p_+)\log_2(p_+) + (p_-)\log_2(p_-)$
- Another impurity function, called **DKM**, was proposed by Dietterich, Kearns, and Mansour [Dietterich et al., 1996]
  - $f(p_+, p_-) = \sqrt{2 \cdot p_+ \cdot p_-}$

Decision Tree Instability (C4.5 algorithm)

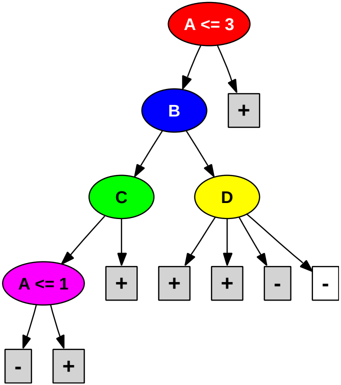
UCI Lymphography dataset
(attributes renamed)



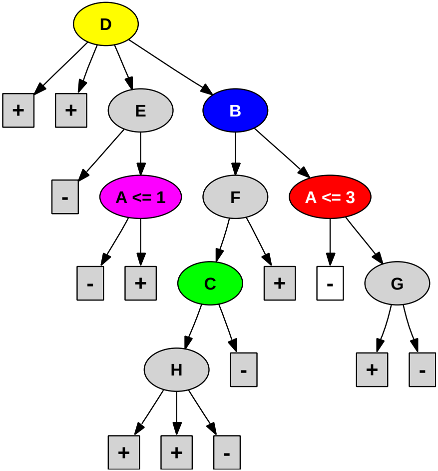
106 training examples

Decision Tree Instability (C4.5 algorithm)

UCI Lymphography dataset
(attributes renamed)



106 training examples



107 training examples

Instability and Decision Tree Induction

Quantifying Stability

Instability in Active Learning

Experiments

Results

Conclusions and Future Work

Types of Stability

- We distinguish between two types of stability: **semantic** and **structural** stability

Types of Stability

- We distinguish between two types of stability: **semantic** and **structural** stability
- Given "similar" data samples, a decision tree learning algorithm is:
  - **semantically stable** if it produces trees that make similar predictions
  - **structurally stable** if it produces trees that are syntactically similar

Quantifying Stability

Semantic stability

Measure the **expected agreement** between two decision trees

- Defined as the probability that two trees predict the same class label for a randomly chosen example [Turney, 1995]
- Estimate the agreement of two trees by having the trees classify a set of randomly chosen **unlabelled examples**

Quantifying Stability

Semantic stability

Measure the **expected agreement** between two decision trees

- Defined as the probability that two trees predict the same class label for a randomly chosen example [Turney, 1995]
- Estimate the agreement of two trees by having the trees classify a set of randomly chosen **unlabelled examples**

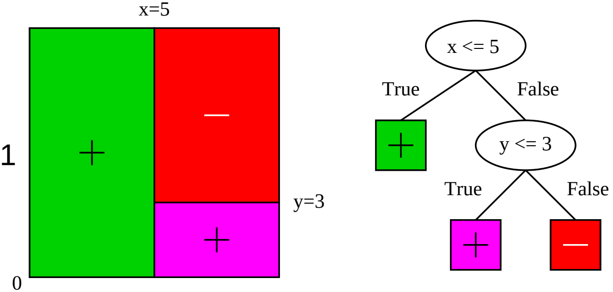
Structural stability

No widely-accepted measure exists for decision trees

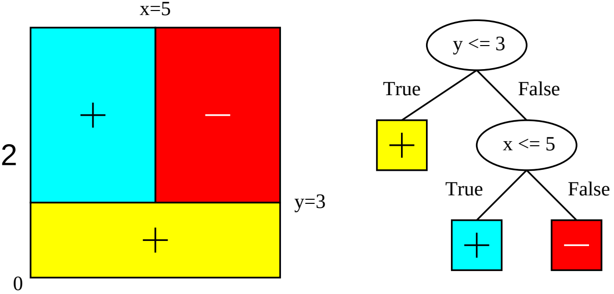
- We propose a novel measure, called **region stability**
- Compare the decision regions (or leaves) in one tree with those of another

Semantic Stability (Example)

Tree 1

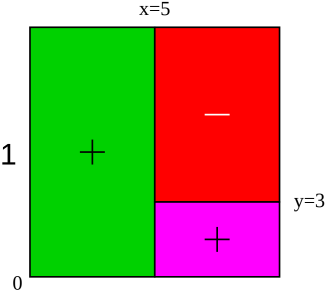


Tree 2

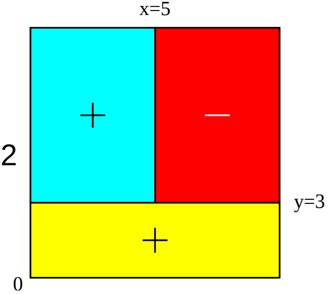


Semantic Stability (Example)

Tree 1



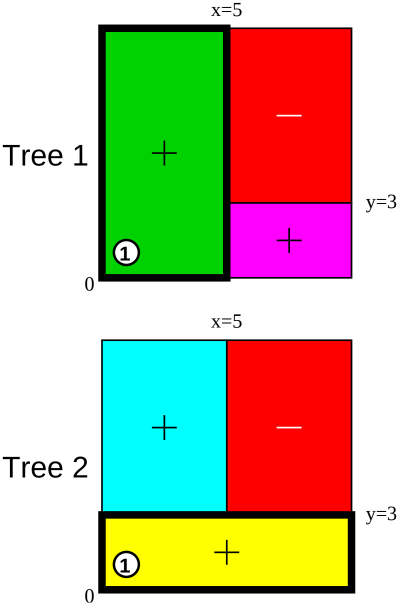
Tree 2



Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Semantic Stability (Example)



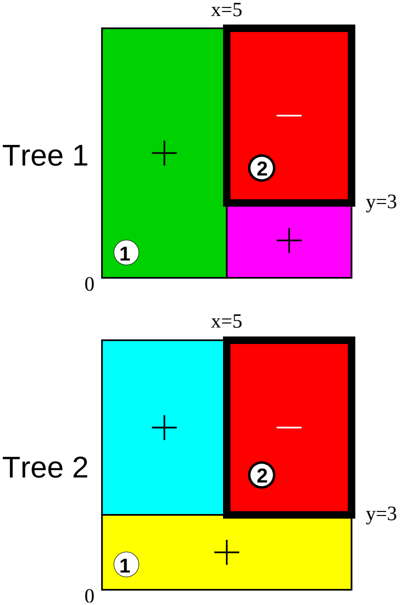
Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Classify unlabelled examples

1. x=1, y=1 (same label) ✓

Semantic Stability (Example)



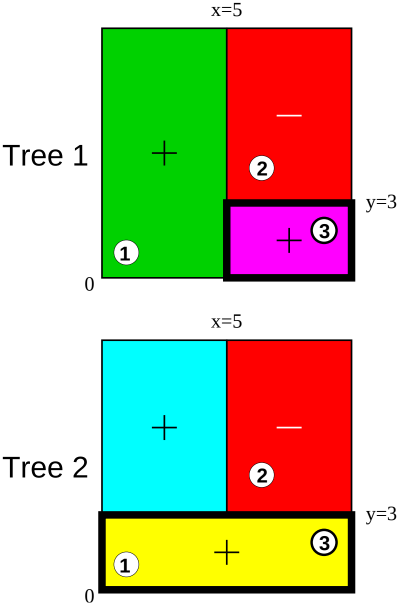
Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Classify unlabelled examples

1. x=1, y=1 (same label) ✓
2. x=6, y=4 (same label) ✓

Semantic Stability (Example)



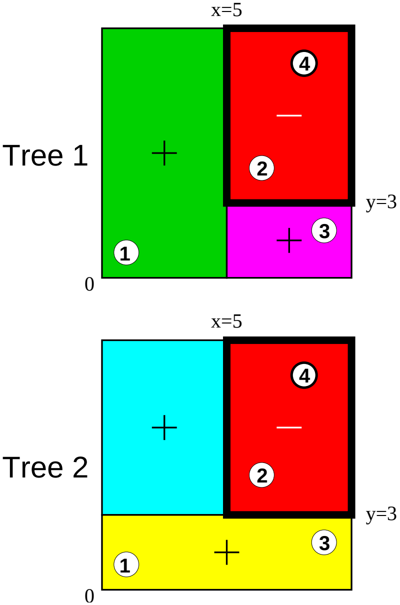
Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Classify unlabelled examples

1. x=1, y=1 (same label) ✓
2. x=6, y=4 (same label) ✓
3. x=9, y=2 (same label) ✓

Semantic Stability (Example)



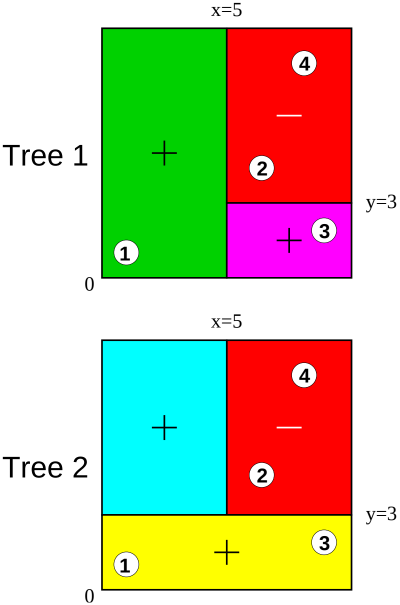
Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Classify unlabelled examples

1. x=1, y=1 (same label) ✓
2. x=6, y=4 (same label) ✓
3. x=9, y=2 (same label) ✓
4. x=8, y=8 (same label) ✓

Semantic Stability (Example)



Semantic Stability

The probability that the two trees assign the same class label to an unseen example

Classify unlabelled examples

1. x=1, y=1 (same label) ✓
2. x=6, y=4 (same label) ✓
3. x=9, y=2 (same label) ✓
4. x=8, y=8 (same label) ✓

- **Score = 4/4 = 1**

Region Stability

Region Stability

- Each leaf in a decision tree is a **decision region**
  - Defined by the unordered set of tests along the path from the root to the leaf

Region Stability

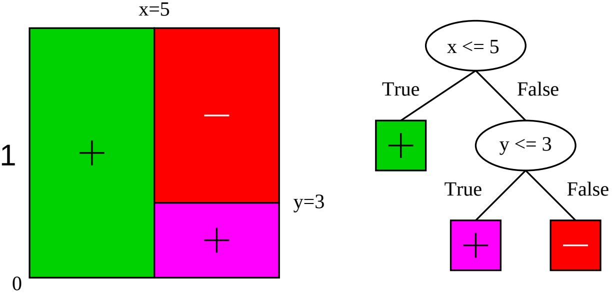
- Each leaf in a decision tree is a **decision region**
  - Defined by the unordered set of tests along the path from the root to the leaf
- Two decision regions are "equivalent" if they perform the **same set of tests** and predict the **same class label**

Region Stability

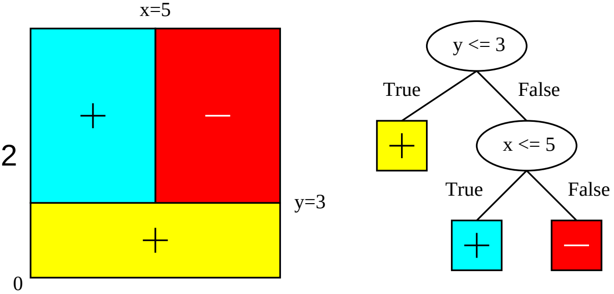
- Each leaf in a decision tree is a **decision region**
  - Defined by the unordered set of tests along the path from the root to the leaf
- Two decision regions are "equivalent" if they perform the **same set of tests** and predict the **same class label**
- We estimate the region stability of two trees by having the trees classify a set of randomly chosen **unlabelled examples**

Region Stability (Example)

Tree 1

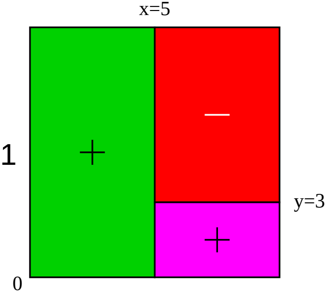


Tree 2

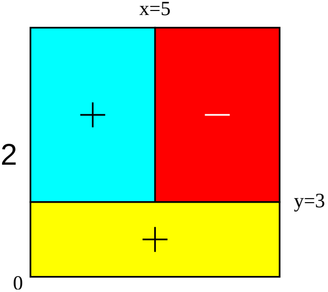


Region Stability (Example)

Tree 1



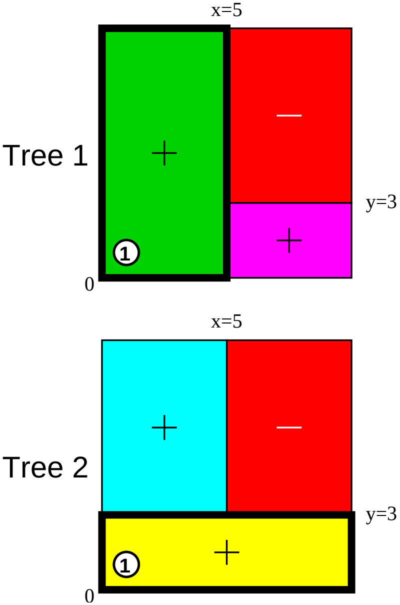
Tree 2



Region Stability

The probability that the two trees classify an unseen example in "equivalent" decision regions

Region Stability (Example)



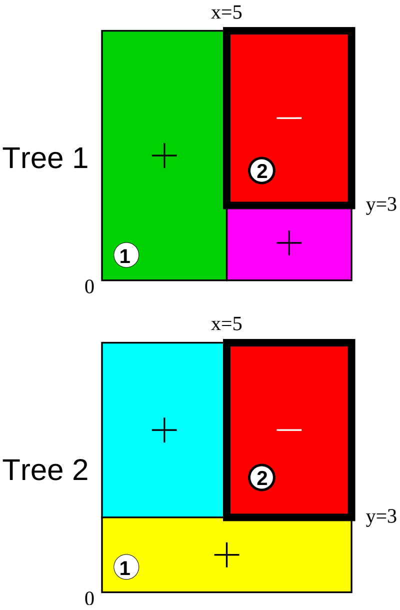
Region Stability

The probability that the two trees classify an unseen example in "equivalent" decision regions

Classify unlabelled examples

1. x=1, y=1 (different)

Region Stability (Example)



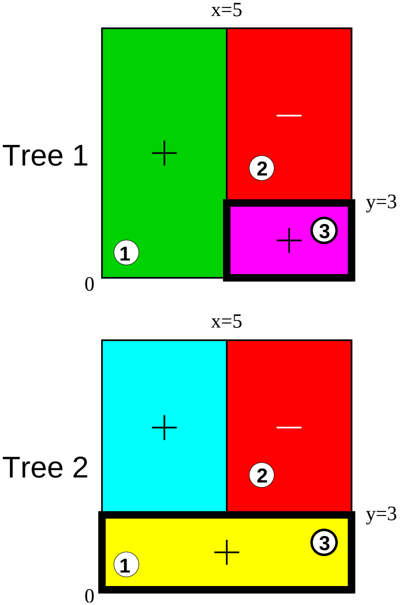
Region Stability

The probability that the two trees classify an unseen example in "equivalent" decision regions

Classify unlabelled examples

1. x=1, y=1 (different)
2. x=6, y=4 (equivalent) ✓

Region Stability (Example)



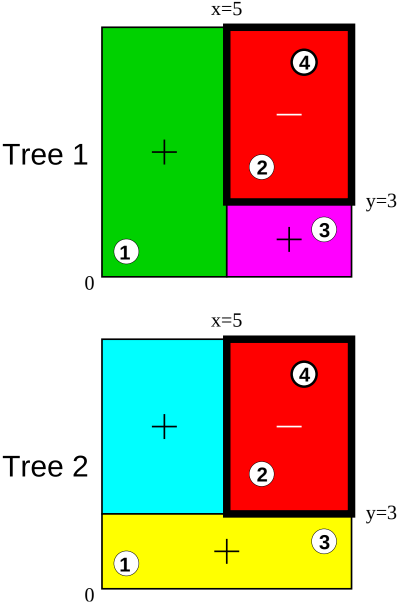
Region Stability

The probability that the two trees classify an unseen example in "equivalent" decision regions

Classify unlabelled examples

1. x=1, y=1 (different)
2. x=6, y=4 (equivalent) ✓
3. x=9, y=2 (different)

Region Stability (Example)



Region Stability

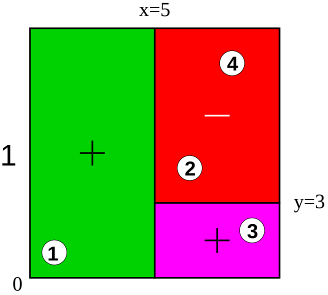
The probability that the two trees classify an unseen example in "equivalent" decision regions

Classify unlabelled examples

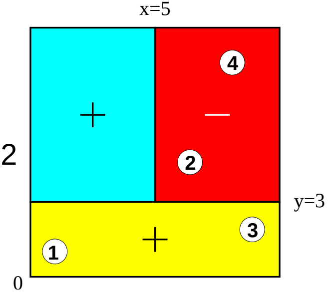
1. x=1, y=1 (different)
2. x=6, y=4 (equivalent) ✓
3. x=9, y=2 (different)
4. x=8, y=8 (equivalent) ✓

Region Stability (Example)

Tree 1



Tree 2



Region Stability

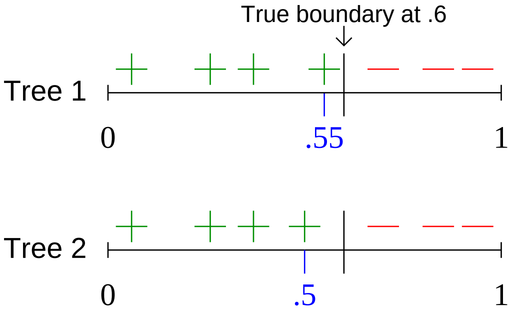
The probability that the two trees classify an unseen example in "equivalent" decision regions

Classify unlabelled examples

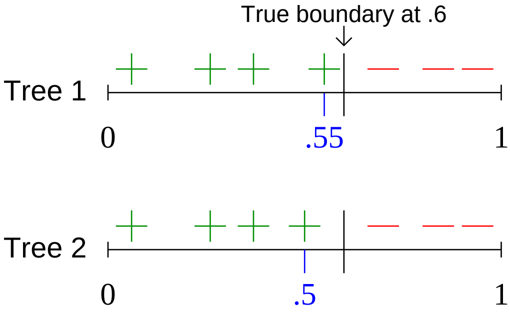
1. x=1, y=1 (different)
2. x=6, y=4 (equivalent) ✓
3. x=9, y=2 (different)
4. x=8, y=8 (equivalent) ✓

- **Score = 2/4 = 0.5**

Region Stability: Continuous Attributes



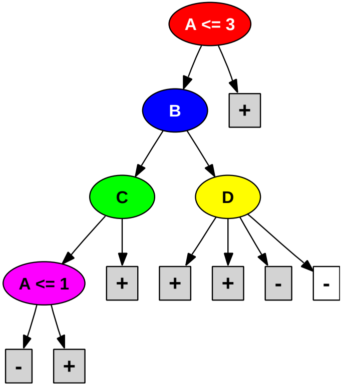
Region Stability: Continuous Attributes



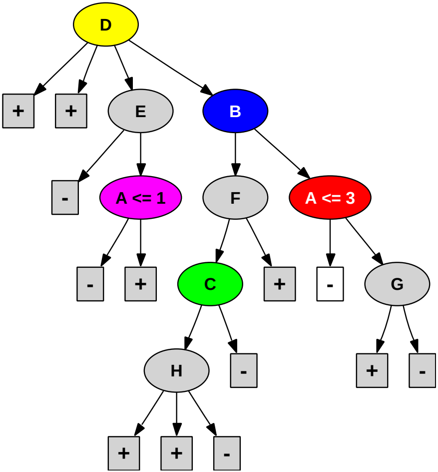
- Specify a value $\varepsilon \in [0, 100]\%$
- Thresholds that are within this range of one another are considered to be equal

C4.5 Instability Example

UCI Lymphography dataset
(attributes renamed)



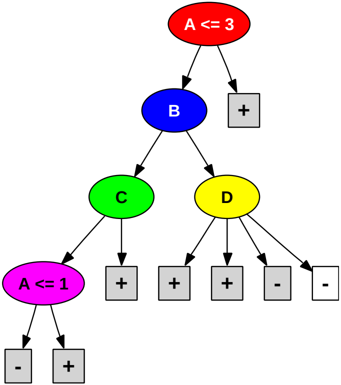
106 training examples

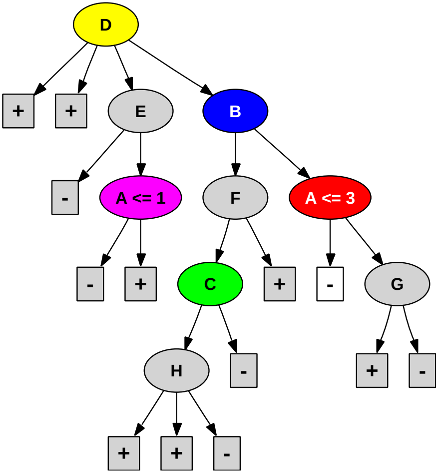


107 training examples

C4.5 Instability Example

UCI Lymphography dataset
(attributes renamed)





106 training examples $\longrightarrow$ **Active Learning** $\longrightarrow$ 107 training examples

Active Learning

Active Learning

- In a **passive learning** setting, the learner is provided with a set of training examples (typically drawn at random)

Active Learning

- In a **passive learning** setting, the learner is provided with a set of training examples (typically drawn at random)
- In **active learning** [Cohn et al., 1992], the learner controls the examples that it uses to train a classifier
- Three main active learning paradigms:
  1. Pool-based
  2. Stream-based
  3. Membership queries

Active Learning

- In a **passive learning** setting, the learner is provided with a set of training examples (typically drawn at random)
- In **active learning** [Cohn et al., 1992], the learner controls the examples that it uses to train a classifier
- Three main active learning paradigms:
  1. Pool-based
  2. Stream-based
  3. Membership queries

  - We focus on pool-based active learning, or **selective sampling**

Active Learning

- In a **passive learning** setting, the learner is provided with a set of training examples (typically drawn at random)
- In **active learning** [Cohn et al., 1992], the learner controls the examples that it uses to train a classifier
- Three main active learning paradigms:
  1. Pool-based
  2. Stream-based
  3. Membership queries
  - We focus on pool-based active learning, or **selective sampling**
- Active learning methods have been shown to make more efficient use of unlabelled data
  - Yet, no attention has been given to their stability

Selective Sampling

Given: A pool of unlabelled data U and some labelled data L

Repeat until (some stopping criterion is met):

1. Train a classifier on the labelled data L
2. Select a **batch** of m examples from the pool U, obtain their labels, and add them to the training set L

Selective Sampling

Given: A pool of unlabelled data U and some labelled data L

Repeat until (some stopping criterion is met):

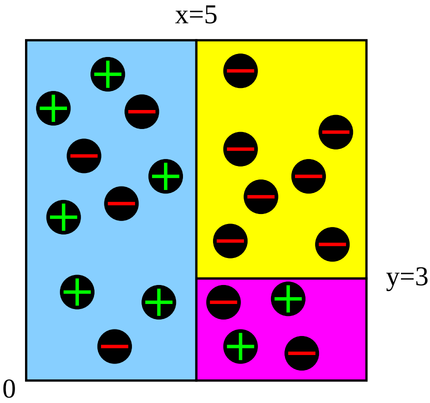
1. Train a classifier on the labelled data L
2. Select a **batch** of m examples from the pool U, obtain their labels, and add them to the training set L

We empirically studied 4 selective sampling methods that can use C4.5 as a base learner:

1. **Uncertainty sampling** [Lewis and Catlett, 1994]
2. **Query-by-bagging** [Abe and Mamitsuka, 1998]
3. **Query-by-boosting** [Abe and Mamitsuka, 1998]
4. **Bootstrap-LV** [Saar-Tsechansky and Provost, 2004]

- **Random sampling** served as a baseline comparison

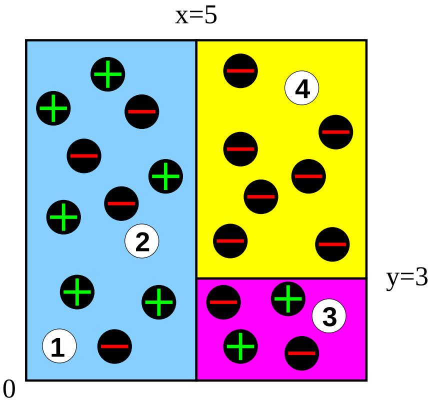
Uncertainty Sampling



Sampling strategy

- Select the examples for which the current prediction is least confident

Uncertainty Sampling

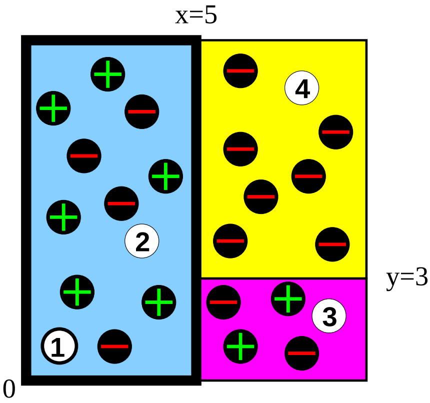


Sampling strategy

- Select the examples for which the current prediction is least confident

Unlabelled data (the pool)

Uncertainty Sampling



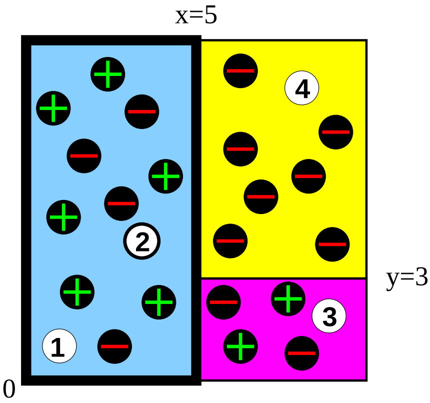
Sampling strategy

- Select the examples for which the current prediction is least confident

Unlabelled data (the pool)

1. x=1, y=1 (Conf: $6/10 = 0.6$)

Uncertainty Sampling



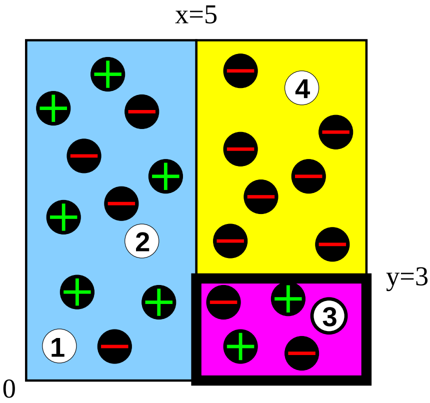
Sampling strategy

- Select the examples for which the current prediction is least confident

Unlabelled data (the pool)

1. x=1, y=1 (Conf: $6/10 = 0.6$)
2. x=3, y=4 (Conf: $6/10 = 0.6$)

Uncertainty Sampling



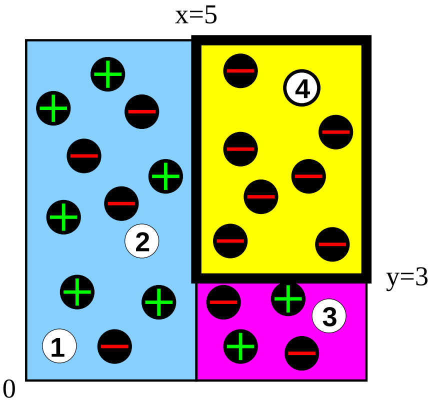
Sampling strategy

- Select the examples for which the current prediction is least confident

Unlabelled data (the pool)

1. x=1, y=1 (Conf: $6/10 = 0.6$)
2. x=3, y=4 (Conf: $6/10 = 0.6$)
3. x=9, y=2 (Conf: $2/4 = 0.5$)

Uncertainty Sampling



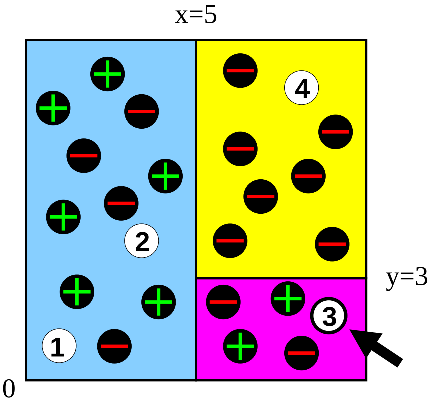
Sampling strategy

- Select the examples for which the current prediction is least confident

Unlabelled data (the pool)

1. x=1, y=1 (Conf: 6/10 = 0.6)
2. x=3, y=4 (Conf: 6/10 = 0.6)
3. x=9, y=2 (Conf: 2/4 = 0.5)
4. x=8, y=8 (Conf: 7/7 = 1)

Uncertainty Sampling



Sampling strategy

- Select the examples for which the current prediction is least confident

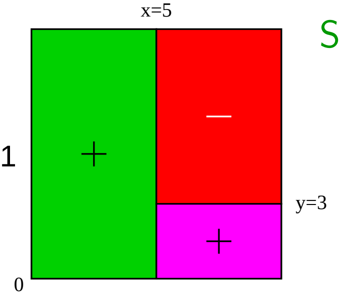
Unlabelled data (the pool)

1. x=1, y=1 (Conf: $6/10 = 0.6$)
2. x=3, y=4 (Conf: $6/10 = 0.6$)
3. x=9, y=2 (Conf: $2/4 = 0.5$)
4. x=8, y=8 (Conf: $7/7 = 1$)

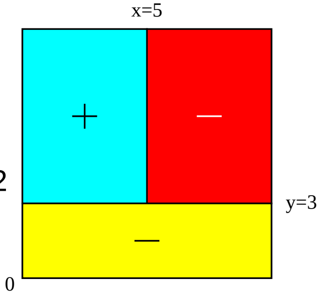
- **Request the label for 3**

Query-by-Bagging

Member 1



Member 2

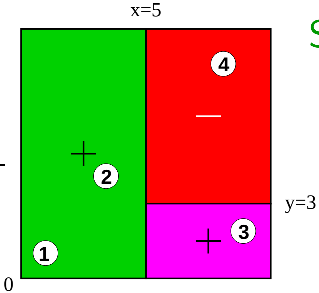


Sampling strategy

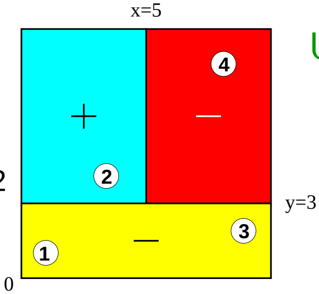
- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Query-by-Bagging

Member 1



Member 2



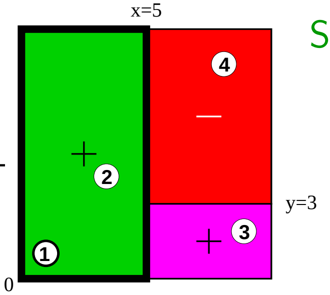
Sampling strategy

- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Unlabelled data (the pool)

Query-by-Bagging

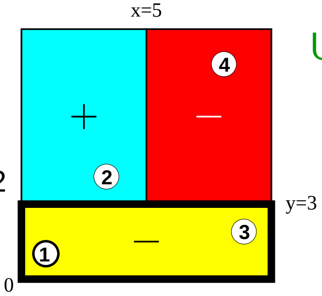
Member 1



Sampling strategy

- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Member 2

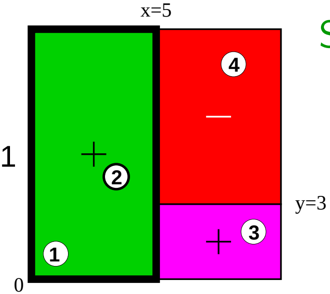


Unlabelled data (the pool)

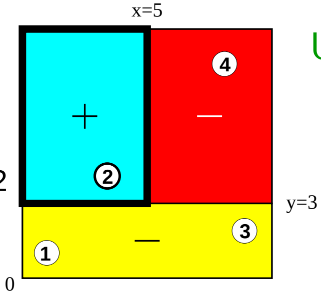
1. x=1, y=1 (Disagree: +,–)

Query-by-Bagging

Member 1



Member 2



Sampling strategy

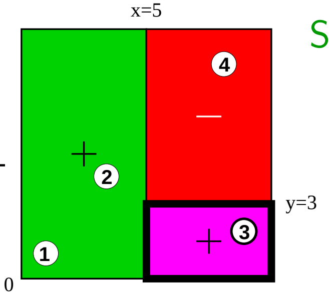
- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Unlabelled data (the pool)

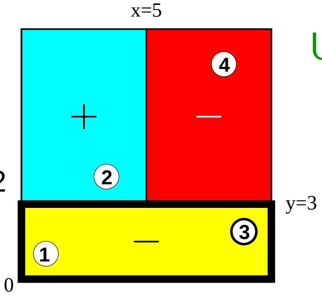
1. x=1, y=1 (Disagree: +,–)
2. x=3, y=4 (Agree: +,+)

Query-by-Bagging

Member 1



Member 2



Sampling strategy

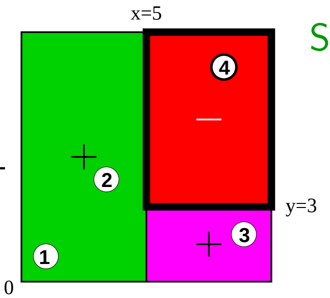
- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Unlabelled data (the pool)

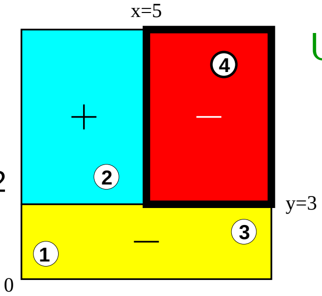
1. x=1, y=1 (Disagree: +,–)
2. x=3, y=4 (Agree: +,+)
3. x=9, y=2 (Disagree: +,–)

Query-by-Bagging

Member 1



Member 2



Sampling strategy

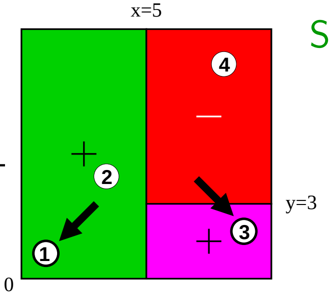
- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Unlabelled data (the pool)

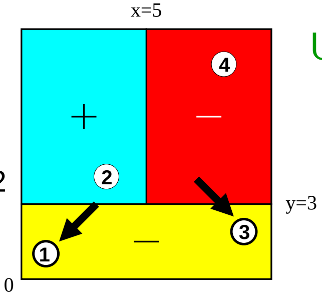
1. $x=1$, $y=1$ (Disagree: $+$,–)
2. $x=3$, $y=4$ (Agree: $+$,$+$)
3. $x=9$, $y=2$ (Disagree: $+$,–)
4. $x=8$, $y=8$ (Agree: –,–)

Query-by-Bagging

Member 1



Member 2



Sampling strategy

- Build a committee (of trees) from the labelled data
- Select the examples for which the committee "vote" is most evenly split

Unlabelled data (the pool)

1. x=1, y=1 (Disagree: +,–)
2. x=3, y=4 (Agree: +,+)
3. x=9, y=2 (Disagree: +,–)
4. x=8, y=8 (Agree: –,–)

Other Sampling Methods

Query-by-Boosting

- Committee is formed using the **AdaBoost.M1** algorithm [Freund and Schapire, 1996]
- Committee member t_i has **voting weight** $\beta_i = \frac{\epsilon_i}{1-\epsilon_i}$, where ϵ_i is the weighted error rate of t_i

Other Sampling Methods

Query-by-Boosting

- Committee is formed using the **AdaBoost.M1** algorithm [Freund and Schapire, 1996]
- Committee member t_i has **voting weight** $\beta_i = \frac{\epsilon_i}{1-\epsilon_i}$, where ϵ_i is the weighted error rate of t_i

Bootstrap-LV (**L**ocal **V**ariance)

- Bagging; Examples are selected by sampling (without replacement) from the distribution $D(\mathbf{x})$, $\mathbf{x} \in U$
  - $D_i(\mathbf{x})$ is inversely proportional to the **variance** in the class probability estimates (CPEs) for example $\mathbf{x}_i$

Other Sampling Methods

Query-by-Boosting

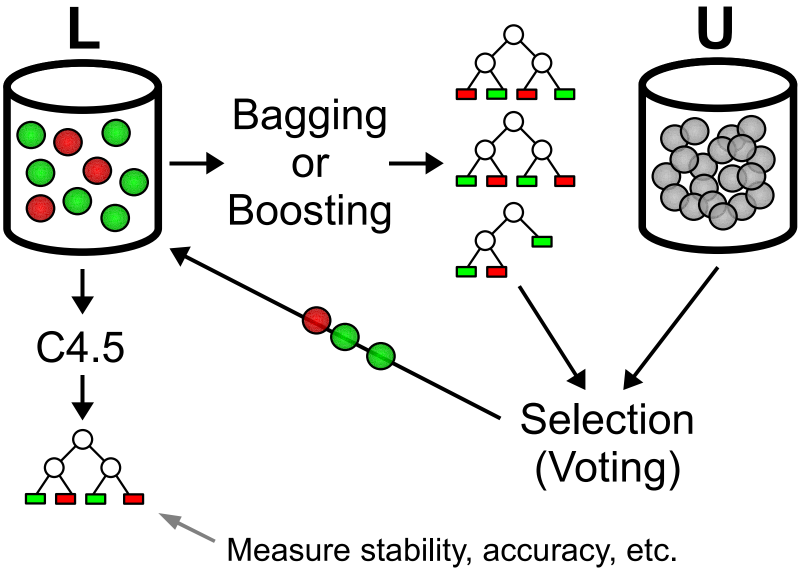
- Committee is formed using the **AdaBoost.M1** algorithm [Freund and Schapire, 1996]
- Committee member t_i has **voting weight** $\beta_i = \frac{\epsilon_i}{1-\epsilon_i}$, where ϵ_i is the weighted error rate of t_i

Bootstrap-LV (**L**ocal **V**ariance)

- Bagging; Examples are selected by sampling (without replacement) from the distribution $D(\mathbf{x})$, $\mathbf{x} \in U$
  - $D_i(\mathbf{x})$ is inversely proportional to the **variance** in the class probability estimates (CPEs) for example $\mathbf{x}_i$

Direct selection versus **Weight sampling**

Committee-based Selective Sampling



Experiments

Questions being addressed

- Do certain selective sampling methods grow **more stable** decision trees than others?

Experiments

Questions being addressed

- Do certain selective sampling methods grow **more stable** decision trees than others?
- Are **committee-based** sampling methods effective at selecting examples for training a **single** decision tree?

Experiments

Questions being addressed

- Do certain selective sampling methods grow **more stable** decision trees than others?
- Are **committee-based** sampling methods effective at selecting examples for training a **single** decision tree?
- Can changing C4.5's **splitting criterion** improve stability?

Experimental Procedure

- 16 UCI datasets [Newman et al., 1998]
  - Only datasets that contained at least 500 examples
  - Multi-class problems converted to two-class
  - Missing values removed

Experimental Procedure

- 16 UCI datasets [Newman et al., 1998]
  - Only datasets that contained at least 500 examples
  - Multi-class problems converted to two-class
  - Missing values removed
- Each dataset was partitioned as follows:

Initial	Unlabelled(Pool)	Evaluation
15%	52%	33%

Experimental Procedure

- 16 UCI datasets [Newman et al., 1998]
  - Only datasets that contained at least 500 examples
  - Multi-class problems converted to two-class
  - Missing values removed
- Each dataset was partitioned as follows:

Initial 15%	Unlabelled(Pool) 52%	Evaluation 33%

- Other parameters:
  - Learning stopped once 2/3 of the pool examples labelled
  - Committees consisted of 10 classifiers
  - Region stability computed using $\epsilon = \{0, 5, 10\}$%
  - Results averaged over 25 runs (diff. initial training data)

Experimental Procedure (Continued)

- We measured three (3) types of active learning stability
- Tree i was compared with...

$$
\begin{array}{l}
L_{01} \rightarrow t_{01,1} \rightarrow t_{01,2} \rightarrow t_{01,3} \rightarrow \ldots \rightarrow t_{01,n} \\
L_{02} \rightarrow t_{02,1} \rightarrow t_{02,2} \rightarrow t_{02,3} \rightarrow \ldots \rightarrow t_{02,n} \\
\quad \vdots \\
L_{25} \rightarrow t_{25,1} \rightarrow t_{25,2} \rightarrow t_{25,3} \rightarrow \ldots \rightarrow t_{25,n}
\end{array}
$$

Experimental Procedure (Continued)

- We measured three (3) types of active learning stability
- Tree i was compared with...
  - the tree grown on iteration $i-1$ (previous tree)

$$
\begin{array}{ccccccccccc}
 & & \downarrow & & \downarrow & & & & & & \\
L_{01} & \longrightarrow & t_{01,1} & \longrightarrow & t_{01,2} & \longrightarrow & t_{01,3} & \longrightarrow & \ldots & \longrightarrow & t_{01,n} \\
L_{02} & \longrightarrow & t_{02,1} & \longrightarrow & t_{02,2} & \longrightarrow & t_{02,3} & \longrightarrow & \ldots & \longrightarrow & t_{02,n} \\
\vdots & & & & & & & & & & \\
L_{25} & \longrightarrow & t_{25,1} & \longrightarrow & t_{25,2} & \longrightarrow & t_{25,3} & \longrightarrow & \ldots & \longrightarrow & t_{25,n}
\end{array}
$$

Experimental Procedure (Continued)

- We measured three (3) types of active learning stability
- Tree i was compared with...
  - the tree grown on iteration $i-1$ (previous tree)
  - the tree grown on iteration n (final tree)

$$
\begin{array}{ccccccccccc}
 & & \downarrow & & \downarrow & & & & & & \\
L_{01} & \longrightarrow & t_{01,1} & \longrightarrow & t_{01,2} & \longrightarrow & t_{01,3} & \longrightarrow & \ldots & \longrightarrow & t_{01,n} \\
L_{02} & \longrightarrow & t_{02,1} & \longrightarrow & t_{02,2} & \longrightarrow & t_{02,3} & \longrightarrow & \ldots & \longrightarrow & t_{02,n} \\
\vdots & & & & & & & & & & \\
L_{25} & \longrightarrow & t_{25,1} & \longrightarrow & t_{25,2} & \longrightarrow & t_{25,3} & \longrightarrow & \ldots & \longrightarrow & t_{25,n} \\
 & & \uparrow & & & & & & & & \uparrow
\end{array}
$$

Experimental Procedure (Continued)

- We measured three (3) types of active learning stability
- Tree i was compared with...
  - the tree grown on iteration $i - 1$ (previous tree)
  - the tree grown on iteration n (final tree)
  - the trees grown on iteration i when given different initial training data L

$$
\begin{array}{ccccccccccc}
L_{01} & \longrightarrow & t_{01,1} & \longrightarrow & t_{01,2} & \longrightarrow & t_{01,3} & \longrightarrow & \ldots & \longrightarrow & t_{01,n} \\
L_{02} & \longrightarrow & t_{02,1} & \longrightarrow & t_{02,2} & \longrightarrow & t_{02,3} & \longrightarrow & \ldots & \longrightarrow & t_{02,n} \\
\vdots & & & & \updownarrow & & & & & & \\
L_{25} & \longrightarrow & t_{25,1} & \longrightarrow & t_{25,2} & \longrightarrow & t_{25,3} & \longrightarrow & \ldots & \longrightarrow & t_{25,n}
\end{array}
$$

Experimental Procedure (Continued)

- We measured three (3) types of active learning stability
- Tree i was compared with...
  - the tree grown on iteration $i - 1$ (previous tree) ■
  - the tree grown on iteration n (final tree) ■
  - the trees grown on iteration i when given different initial training data L ■

$$
\begin{array}{ccccccccccc}
 & & \downarrow & & \downarrow & & & & & & \\
L_{01} & \longrightarrow & t_{01,1} & \longrightarrow & t_{01,2} & \longrightarrow & t_{01,3} & \longrightarrow & \ldots & \longrightarrow & t_{01,n} \\
L_{02} & \longrightarrow & t_{02,1} & \longrightarrow & t_{02,2} & \longrightarrow & t_{02,3} & \longrightarrow & \ldots & \longrightarrow & t_{02,n} \\
\vdots & & & & \updownarrow & & & & & & \\
L_{25} & \longrightarrow & t_{25,1} & \longrightarrow & t_{25,2} & \longrightarrow & t_{25,3} & \longrightarrow & \ldots & \longrightarrow & t_{25,n} \\
 & & \uparrow & & & & & & & & \uparrow
\end{array}
$$

These are called **PrevStab**, **FinalStab**, and **RunStab**

Evaluation

- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Evaluation

- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Example

	Method 1	Method 2	Method 3	Method 4
Dataset 1				
Dataset 2				
Dataset 3				
Avg. Rank				

Evaluation

- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Example

	Method 1	Method 2	Method 3	Method 4
Dataset 1	1	4	2	3
Dataset 2				
Dataset 3				
Avg. Rank				

Evaluation

- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Example

	Method 1	Method 2	Method 3	Method 4
Dataset 1	1	4	2	3
Dataset 2	2	3	1	4
Dataset 3				
Avg. Rank				

Evaluation

- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Example

	Method 1	Method 2	Method 3	Method 4
Dataset 1	1	4	2	3
Dataset 2	2	3	1	4
Dataset 3	1	4	2.5	2.5
Avg. Rank				

Evaluation

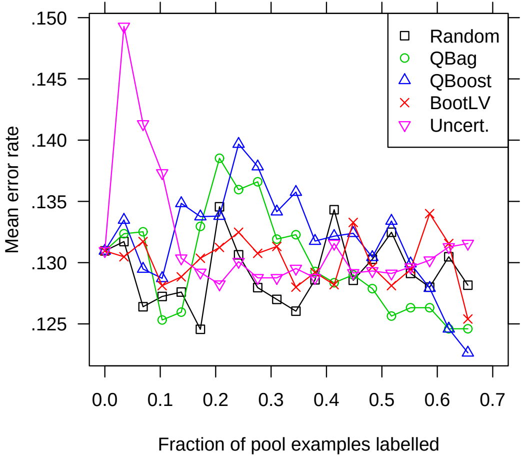
- Statistical significance was assessed by comparing the average ranks of the sampling methods.
  - Recommended procedure for comparing multiple learning methods [Demšar, 2006].

Example

	Method 1	Method 2	Method 3	Method 4
Dataset 1	1	4	2	3
Dataset 2	2	3	1	4
Dataset 3	1	4	2.5	2.5
Avg. Rank	**1.333**	3.667	1.833	3.167

Evaluation (Continued)

- For a given {statistic, sampling method, splitting criterion, data set} tuple, we get a sequence of scores
- How do we **rank** the sampling methods?



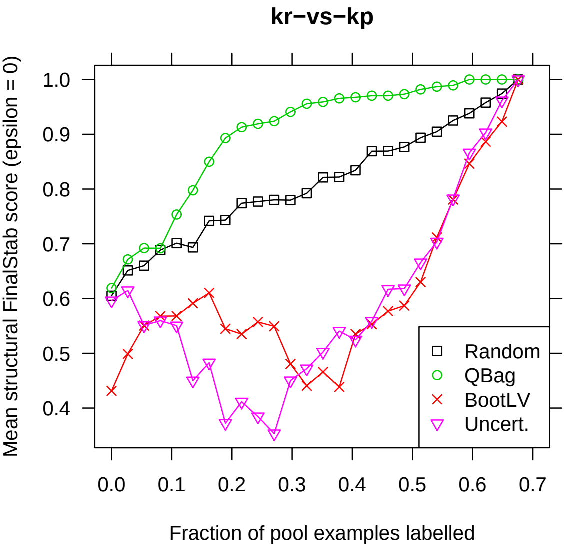
Averaging Scores

- Summary statistic: sequence of scores $\longrightarrow$ single number
  1. Compute the average score s_i at each iteration i (i.e. over the 25 runs)
  2. The overall score is a weighted average $\frac{1}{n}\sum_{i=1}^{n} w_i \cdot s_i$, where $w_i = \frac{2i}{n(n+1)}$

Averaging Scores

- Summary statistic: sequence of scores $\longrightarrow$ single number
  1. Compute the average score s_i at each iteration i (i.e. over the 25 runs)
  2. The overall score is a weighted average $\frac{1}{n}\sum_{i=1}^{n} w_i \cdot s_i$, where $w_i = \frac{2i}{n(n+1)}$
- The weight increases linearly as a function of i
  - We argue that stability and accuracy are most important in the later stages of active learning
  - e.g. Stability in early rounds is of little value if stability deteriorates in later rounds

Example: Averaging Scores and Ranking



Ranks/Scores

1. QBag (.953)
2. Random (.858)
3. BootLV (.644)
4. Uncert (.638)

Statistical Significance [Demšar, 2006]

Dataset	Random (R)	QBag (G)	QBoost (T)	BootLV (L)	Uncert (U)
anneal	.144 (4)	**.121** (1)	.135 (3)	.125 (2)	.150 (5)
australian	**.129** (1.5)	**.129** (1.5)	.131 (5)	.130 (3.5)	.130 (3.5)
car	.090 (5)	**.077** (1)	.082 (4)	.078 (2)	.081 (3)
german	.293 (5)	**.274** (1)	.285 (2)	.290 (4)	.289 (3)
hypothyroid	.006 (5)	**.002** (2)	**.002** (2)	**.002** (2)	.004 (4)
kr-vs-kp	.014 (5)	**.007** (1.5)	.008 (3)	**.007** (1.5)	.010 (4)
letter	.015 (5)	**.011** (2)	**.011** (2)	**.011** (2)	.013 (4)
nursery	.056 (5)	**.038** (1.5)	.039 (3)	**.038** (1.5)	.044 (4)
pendigits	.016 (5)	**.010** (1.5)	**.010** (1.5)	.012 (4)	.011 (3)
pima-indians	.286 (5)	.283 (2)	**.280** (1)	.284 (3)	.285 (4)
segment	.020 (5)	**.011** (1)	.012 (2.5)	.012 (2.5)	.019 (4)
tic-tac-toe	.217 (5)	**.197** (1)	.201 (2)	.207 (3)	.211 (4)
vehicle	**.227** (1)	.231 (5)	.229 (3.5)	.228 (2)	.229 (3.5)
vowel	.056 (5)	**.033** (1)	.036 (2)	.037 (3)	.049 (4)
wdbc	.073 (4)	.068 (2)	**.067** (1)	.069 (3)	.076 (5)
yeast	.256 (4.5)	**.250** (1)	.253 (2.5)	.256 (4.5)	.253 (2.5)
Avg. rank	(4.375)	(1.625) **R**,**U**	(2.500) **R**	(2.719) **R**	(3.781)

- Apply the Friedman and Nemenyi significance tests
  - e.g. At $\alpha = .05$, the critical difference is 1.527

Instability and Decision Tree Induction

Quantifying Stability

Instability in Active Learning

Experiments

Results

Conclusions and Future Work

Error Rates

Error Rates

- The **committee-based** sampling methods achieved lower error rates than did **Uncertainty** or **Random**
  - At first glance, this might not appear to be a novel or interesting result

Error Rates

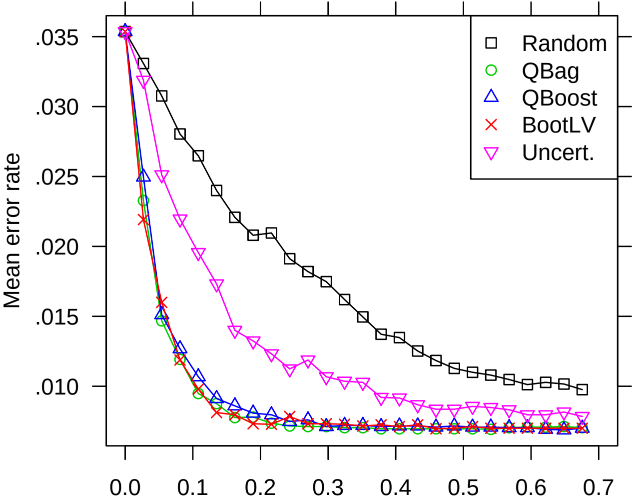
- The **committee-based** sampling methods achieved lower error rates than did **Uncertainty** or **Random**
  - At first glance, this might not appear to be a novel or interesting result
- Important difference from previous active learning studies:
  - A **committee** of C4.5 trees selected examples that were used to train a **single C4.5 tree**, which was evaluated
  - In prior research, e.g., Query-by-bagging selected examples for training a **bagged ensemble of trees**

Error Rates

- The **committee-based** sampling methods achieved lower error rates than did **Uncertainty** or **Random**
  - At first glance, this might not appear to be a novel or interesting result
- Important difference from previous active learning studies:
  - A **committee** of C4.5 trees selected examples that were used to train a **single C4.5 tree**, which was evaluated
  - In prior research, e.g., Query-by-bagging selected examples for training a **bagged ensemble of trees**
- When trained on the same data sample, a committee of trees is likely to be more accurate than a single tree
  - Yet, a committee of trees is no longer interpretable [Breiman, 1996]

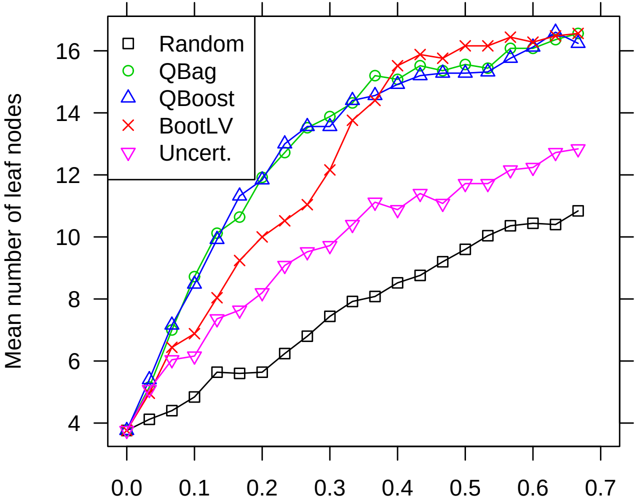
Error Rates (Continued)

- We typically observed a "banana" shape, indicating efficient use of unlabelled data (below: kr-vs-kp)



Tree Size

- The selective sampling methods consistently yielded larger trees than did **Random** sampling (below: vowel)



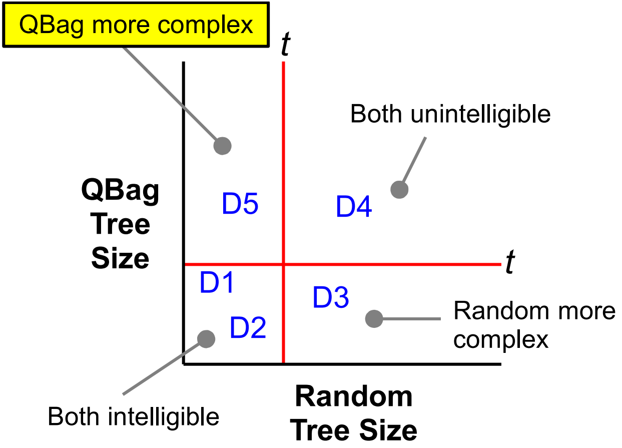
Tree Size and Intelligibility

- Trees grown using Query-by-bagging (QBag) contained 38 percent more leaves, on average, than those of Random
  - Yet, we argue that this did not usually result in a loss of intelligibility

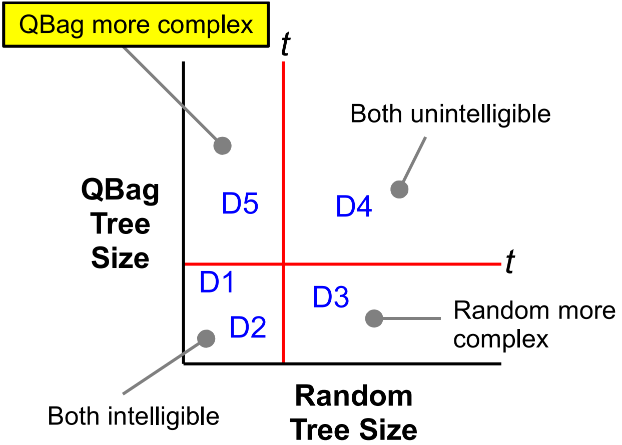
Tree Size and Intelligibility

- Trees grown using Query-by-bagging (QBag) contained 38 percent more leaves, on average, than those of Random
  - Yet, we argue that this did not usually result in a loss of intelligibility
- There is no agreed-upon criterion for distinguishing between a tree that is interpretable and a tree that is not
- Let's consider one simple criterion:
  - There might exist a threshold t, such that any tree containing more than t leaves is uninterpretable
  - On a given dataset, if QBag's leaf count is greater than t while Random's is at most t, then QBag has sacrificed intelligibility

Tree Size and Intelligibility (Continued)

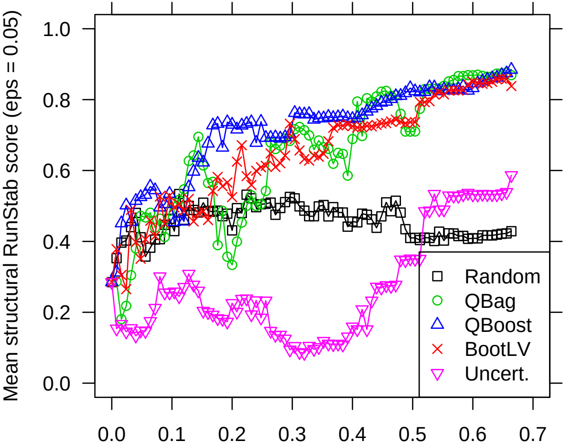


Tree Size and Intelligibility (Continued)



- We examined all integer values of t between 1 and 25, and found QBag to be more complex on at most 5 datasets ($t = 13$)

- **Query-by-bagging** (QBag) grew the most semantically and structurally stable trees
  - Its stability gains **across runs** were highly significant



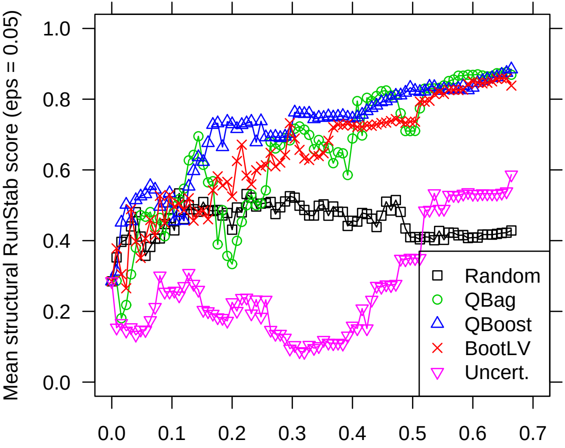
Avg. Ranks

RunStab, $\epsilon = .05$

1. QBag (1.66)
2. QBoost (2.19)
3. BootLV (2.59)
4. Random (4.19)
5. Uncert (4.38)

- Left: letter

- **Query-by-bagging** (QBag) grew the most semantically and structurally stable trees
  - Its stability gains **across runs** were highly significant



Avg. Ranks

RunStab, $\epsilon = .05$

1. QBag (1.66)
2. QBoost (2.19)
3. BootLV (2.59)
4. Random (4.19)
5. Uncert (4.38)

- Left: letter

Direct selection vs. Weight sampling
Committee of trees vs. Single tree

Splitting Criteria: Entropy vs. DKM

- We employed the Wilcoxon signed-ranks test
- **DKM** was more **structurally stable** *and* more **accurate** than entropy
- Structural stability of all 5 sampling methods improved when using DKM
  - The best method, QBag, exhibited even better performance when paired with DKM
- Differences in semantic stability and tree size were, for the most part, insignificant

Main Contributions

1. How should decision tree (in)stability be **measured**?

We proposed a novel structural stability measure for d-trees, called **region stability**, along with active learning versions

Main Contributions

1. How should decision tree (in)stability be **measured**?

We proposed a novel structural stability measure for d-trees, called **region stability**, along with active learning versions

2. How stable are some well-known **active learning** methods that use the **C4.5** decision tree learner?

Query-by-bagging was found to be more stable and more accurate than its competitors

Main Contributions

1. How should decision tree (in)stability be **measured**?

We proposed a novel structural stability measure for d-trees, called **region stability**, along with active learning versions

2. How stable are some well-known **active learning** methods that use the **C4.5** decision tree learner?

Query-by-bagging was found to be more stable and more accurate than its competitors

3. Can stability be improved in this setting by changing C4.5's **splitting criterion**?

The **DKM** splitting criterion was shown to improve the stability and accuracy of C4.5 in active learning

Future Work

Incremental Tree Induction [Utgoff et al., 1997]

- Tree is restructured when new training data arrive
  - On average, requires less computation than growing a new tree from scratch
- Error-correction mode: Only add a new example if the existing tree would misclassify it
- Alternatively, we could add all new examples, but only update the tree if an example is misclassified
  - These “good enough” trees might be more stable

Future Work (Continued)

Learning under Covariate Shift [Bickel et al., 2007]

- Active learning constructs a training set whose distribution may differ arbitrarily from the original
  - I could be the case that $p_{train}(x) \neq p_{test}(x)$
- The expected loss is minimized when training examples are weighted by:

$$\frac{p_{test}(x)}{p_{train}(x)}$$

- Is such a correction beneficial in active learning?
- Are techniques for dealing with class imbalance are more appropriate?

Conclusions

- When training a single C4.5 tree in an active learning setting, one should use the DKM splitting criterion and select examples with Query-by-bagging
  - This combination yields the most stable and accurate decision trees

Conclusions

- When training a single C4.5 tree in an active learning setting, one should use the DKM splitting criterion and select examples with Query-by-bagging
  - This combination yields the most stable and accurate decision trees
- We should be aware of the potential instability of machine learning algorithms, particularly when attempting to extract knowledge from a classifier

Thank You!

?

Selected References

- Abe, N. and Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proc. ICML '98*, pages 1–9.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Cohn, D. A., Atlas, L. E., and Ladner, R. E. (1992). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30.
- Dietterich, T. G., Kearns, M., and Mansour, Y. (1996). Applying the weak learning framework to understand and improve C4.5. In *Proc. ICML '96*, pages 96–104.
- Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proc. ICML '94*, pages 148–156.
- Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *JAIR*, 4:77–90.
- Saar-Tsechansky, M. and Provost, F. (2004). Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2):153–178.
- Turney, P. D. (1995). Bias and the quantification of stability. *Machine Learning*, 20(1-2):23–33.