Decision Trees: Evaluation

Mauro Sozio*

*slides adapted from the course: Introduction to data mining, Steinbach, Kumar

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than running time, scalability, etc.
- Confusion Matrix:

	PREDICTED Class	
		Class=Yes	Class=No
ACTUAL Class	Class=Yes	TP	FN
	Class=No	FP	TN

TP (true positive)
FN (false negative)
FP (false positive)
TN (true negative)

Metrics for Performance Evaluation…

	PREDICTED Class	
	Class=Yes	**Class=No**
ACTUAL Class Class=Yes	TP	FN
ACTUAL Class Class=No	FP	TN

- Most widely-used metric:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

Limitation of Accuracy

- Consider a 2-Class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be Class 0, accuracy is 9990/10000 = 99.9 %
  - Misleading: model does not detect any Class 1 example

Cost Matrix and Weighted Accuracy

C(i|j): Cost of misclassifying Class j record as Class i

		PREDICTED Class	
	C(i|j)	**Class=Yes**	**Class=No**
ACTUAL Class	**Class=Yes**	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

$$\textbf{Weighted Acc.} = \frac{\text{TP x C(Yes|YES)} + \text{TN x C(No|No)}}{\text{TP x C(Yes|YES)} + \text{FN x C(No|yes)} + \text{FP x C (Yes|No)} + \text{TN x C(No|No)}}$$

Cost of Classification

Cost Matrix	PREDICTED Class		
ACTUAL Class	C(i|j)	+	-
	+	-1	100
	-	1	0

Cost Matrix Penalizing FN

Cost = TP x c(TP) + TN x c(TN) + FP x c(FP) + FN x c(FN)

Model M_1	PREDICTED Class		
ACTUAL Class		+	-
	+	150	40
	-	60	250

Model M_2	PREDICTED Class		
ACTUAL Class		+	-
	+	205	65
	-	30	200

Accuracy = 400 / 500= 80%
Cost = 3910

Accuracy = 405 /500 =81%
Cost = 6325

Precision, Recall, F-measure

$$\text{Precision (p)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall (r)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Easy to get only high precision: ‘Yes’ only for instances we are “sure” about (e.g. easy to classify “robot” if 1000 pages/sec. retrieved) => low recall

Trivial to get only high recall: classify all instances as ‘Yes’ => low precision

Precision, Recall, F-measure

F-measure is the harmonic mean of p and r, which penalizes very small recall or very small precision

$$\text{F-measure (F)} = \frac{2rp}{r+p} \quad \text{(Harmonic Mean of p and r)}$$

E.g. : Harmonic Mean (1,2,4) = 1/ (1/3 (1+1/2+/1/4)) = 12/7 (reciprocal of the arithmetic mean of the reciprocals)

It can be shown *min* (x1,...,xn) <= H(x1,...,xn) <= n * *min* (x1,...,xn)

=> min is very small then harmonic mean is very small.

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  – Class distribution
  – Cost of misclassification
  – Size of training and test sets

Methods of Estimation

- Holdout
  - Reserve random 2/3 for training and 1/3 for testing
- Random subsampling
  - Repeated holdout (compute the average)
- Cross validation
  - Partition data into k disjoint subsets of the same size
  - k-fold: train on k-1 sets, test on the remaining one. Repeat k times, each subset being used exactly once for testing. Compute the average of the results.
  - Leave-one-out: k=n, (n-1 sets training test, rest test set)
- Bootstrap
  - Sampling with replacement