

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
	Class=Yes	TP	FN
	Class=No	FP	TN

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

Data Mining: Decision Trees

Metrics for Performance Evaluation...

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

- | Most widely-used metric:

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

Data Mining: Decision Trees

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	$C(\text{Yes} \text{Yes})$	$C(\text{No} \text{Yes})$
	Class=No	$C(\text{Yes} \text{No})$	$C(\text{No} \text{No})$

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

Data Mining: Decision Trees

Cost of Classification

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

Cost Matrix Penalizing FN

$$Cost = TP \times c(TP) + TN \times c(TN) + FP \times c(FP) + FN \times c(FN)$$

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

Accuracy = $400 / 500 = 80\%$
 Cost = 3910

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

Accuracy = $405 / 500 = 81\%$
 Cost = 6325

Data Mining: Decision Trees

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

Data Mining: Decision Trees

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 $\min (x_1, \dots, x_n) \leq H(x_1, \dots, x_n) \leq n * \min (x_1, \dots, x_n)$
 \Rightarrow 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: $k=n$, (1 record training test, rest test set)
- | Bootstrap
 - Sampling with replacement

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?

Test of Significance

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

- | Given two models:
 - Model M1: accuracy = 85%, tested on 30 instances
 - Model M2: accuracy = 75%, tested on 5000 instances
- | Can we say M1 is better than M2?
- | How much confidence can we place on accuracy of M1, M2?

Data Mining: Decision Trees

Confidence Interval for Accuracy

- | Estimate accuracy by running classifier on all possible instances, **unfeasible!**
- | If we have a sample of all records (uniform and independent), then, determining CI for accuracy is related to the problem:
 - *we are given a coin with p being the probability of getting a head. We flip it N times, getting x heads.*
 - ◆ is $p' = x/N$ close to p ?
 - ◆ how much confidence on the accuracy of p' ?

Bernoulli Experiment

- | Given a coin with probability p of getting a 'head', flipped N times. Let X be the random variable denoting the number of heads.
- | $\text{Prob}(X = x) = \binom{N}{x} p^x (1-p)^{N-x}$
- | Example: coin is fair ($p=0.5$), flipped 50 times, then the prob. of having 20 heads = 0.0419.
- | $X \sim \text{Bin}(N, p)$, we have $E[X]=Np$, $\text{Var}(X)=Np(1-p)$

Data Mining: Decision Trees

Confidence Interval for Accuracy

- | If N is large with high probability we get Np heads.
- | Then, if N is large enough we can easily estimate p (accuracy of our classifier) as $\#heads/N$.
- | N : how large? How confident on our estimation?
- | From *central limit theorem** it follows that we can approximate binomial distributions by normal distributions.....

* arithmetic mean of a (sufficiently large) number of independent r.v. with variance σ^2 can be approximated by a normal distribution with variance σ^2 .

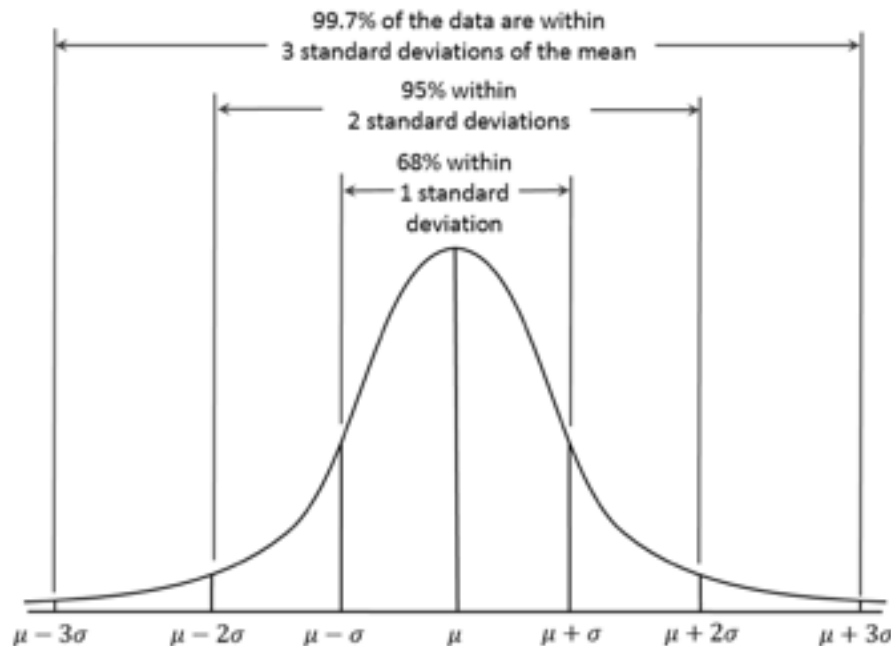
Data Mining: Decision Trees

Normal distribution

- Standard normal distribution $X \sim N(\mu, \sigma)$ is described by the following probability density function:

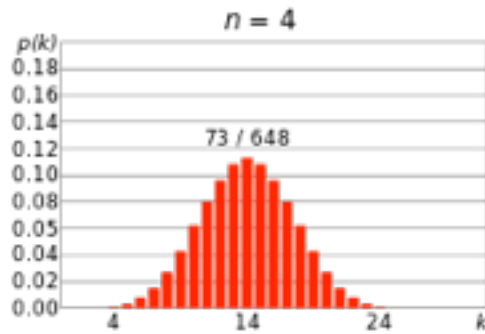
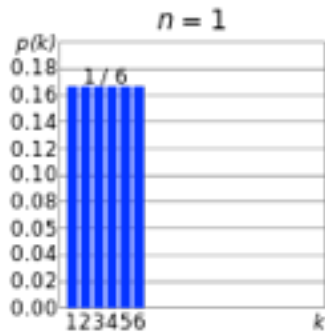
$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Intuitively, one can think of $P(x) dx$ as the probability of X falling within the infinitesimal interval $[x, x+dx]$

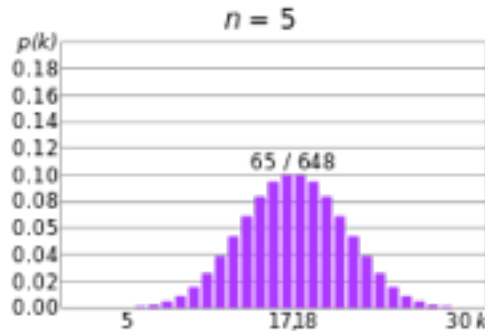
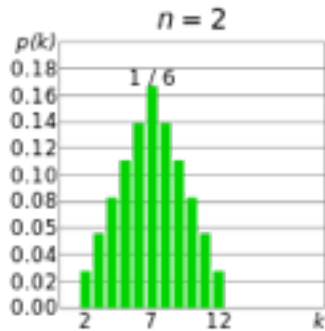


Data Mining: Decision Trees

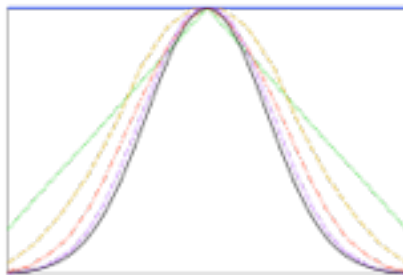
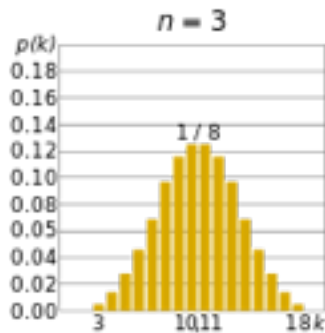
Approximating Binomial Distribution



A dice is rolled n times, each graph shows the probability $p(k)$ that the sum of the values of the dice be k .



The last plot shows the normal distribution (black curve) and the other curves.

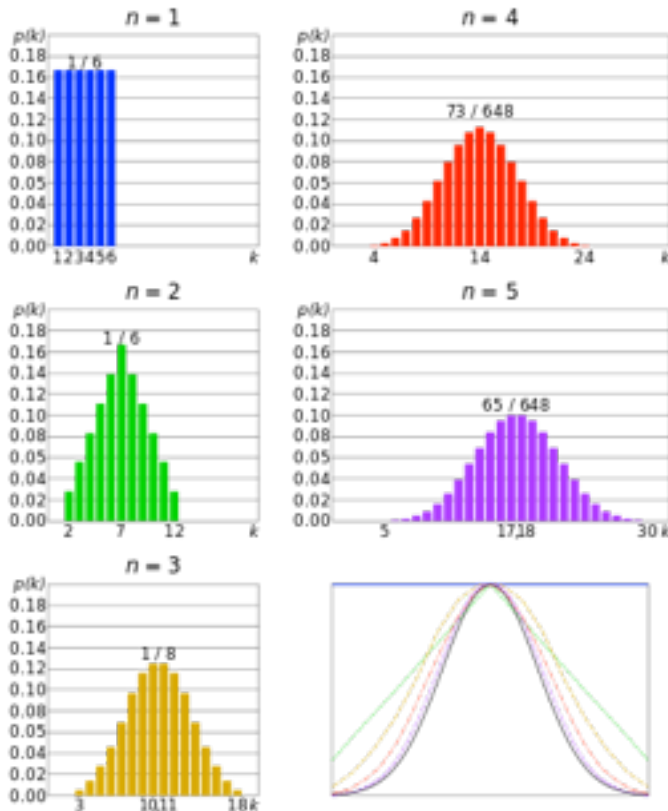


The normal distribution gives a good approximation as n gets larger.

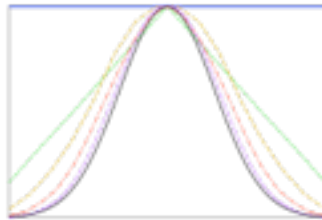
Data Mining: Decision Trees

Confidence Interval for Accuracy

- What is the probability that the number of heads is in $[a,b]$? Approximated by the AUC (integral) of the normal distribution in $[a,b]$ (N large, e.g. $N > 30$)



Probability that the average sum of N dice rolls is in $[a,b]$ can be approximated by the integral in $[a,b]$ of the normal distribution (bottom right)



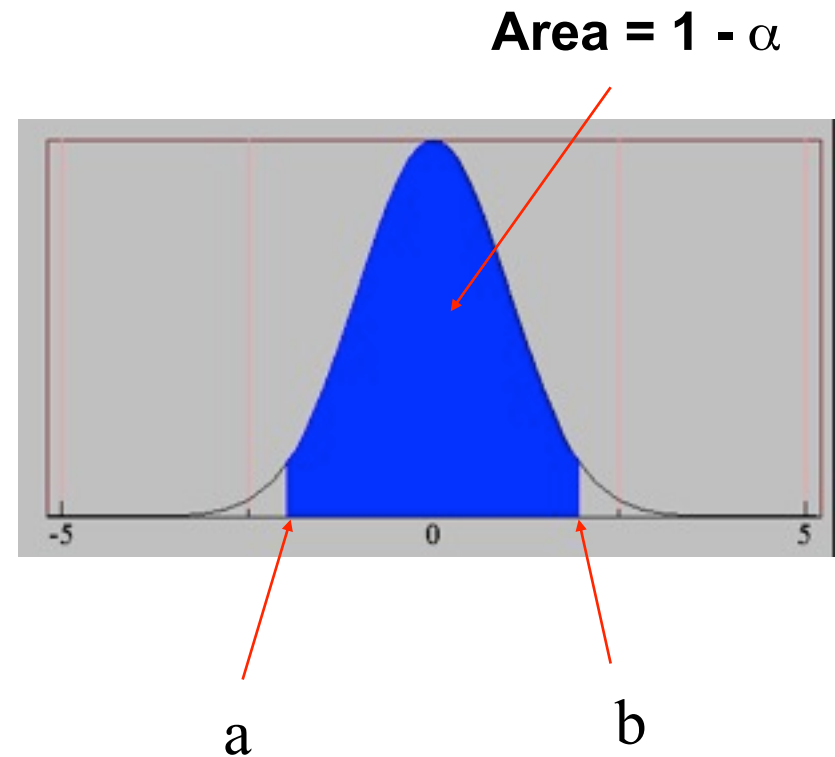
Data Mining: Decision Trees

Confidence Interval for Accuracy

- For large test sets ($N > 30$), $\text{acc} = (\text{TP} + \text{TN}) / N$ has normal distribution with mean p and variance $\sigma^2 = p(1-p)/N$ ($N = \text{\#test instances}$, $p = \text{real accuracy}$ which is unknown.)

If we want to have confidence $1 - \alpha$ we need to find a, b such that the integral of that normal distribution in $[a, b]$ is $1 - \alpha$.

Usually focus on confidence = 0.95.



Confidence Interval for Accuracy

- | Values of those integrals have been pre-computed.
- | E.g. consider a model that produces an accuracy of 80% when evaluated on 100 test instances:
 - $N=100$, $\text{acc} = 0.8$
 - Let $1-\alpha = 0.95$ (95% confidence)

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.824	0.811

According to the precomputed tables the model has accuracy in $[0.711, 0.866]$ with confidence 95%,

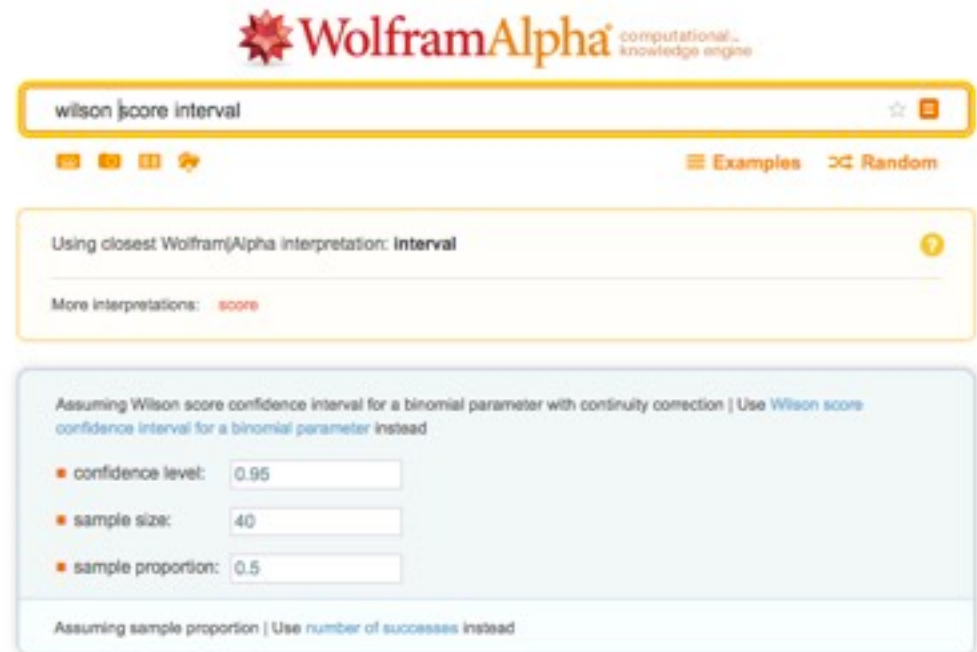
Intuition for Confidence Intervals

- | Prob. (in a sample of N test instances, Np test instances are classified successfully and p is in $[0.711, 0.866]$) = 0.95. In other words...
- | If we run the classifier again on a sample of N test instances, there is chance of 0.95 that Np test instances are classified correctly, p in $[0.711, 0.866]$.
- | As N gets larger, the confidence interval converges to the real accuracy of the classifier (i.e., when we run the classifier on all possible test instances).

Approximating Normal Distribution

- | Normal distribution:
 - does not work well if $N < 30$, or p is close to 0 or 1
 - rule of thumb: $Np > 5$ and $N(1-p) > 5$
- | More sophisticated techniques: Wilson score interval, Jeffrey intervals, Clopper-Pearson interval..

Most are available in
Wolfram Alpha



The screenshot shows the Wolfram Alpha interface. At the top, the search bar contains the text "wilson score interval". Below the search bar, there are icons for input methods (text, voice, image, etc.) and links for "Examples" and "Random". The main content area displays the result: "Using closest Wolfram|Alpha interpretation: Interval". Below this, it says "More interpretations: score". At the bottom, there is a section titled "Assuming Wilson score confidence interval for a binomial parameter with continuity correction | Use Wilson score confidence interval for a binomial parameter instead". This section contains three input fields: "confidence level:" with the value "0.95", "sample size:" with the value "40", and "sample proportion:" with the value "0.5". At the very bottom of this section, it says "Assuming sample proportion | Use number of successes instead".

Data Mining: Decision Trees

A few additional comments on CI

- | We are making the assumption that test instances are drawn uniformly and independently at random, which is often not the case.
- | Observe that the results of the classifier on the test instances are usually not independent, this is not an issue.
- | Overall CI are reliable and give valuable information on the performance of a classifier.