

Year 2000 Confidence Assessment **– *A Bayesian Tutorial***

John A. Vitkevich, Jr.
22 December 1997

This presentation describes a simple application of Bayes' theorem to Year 2000 (Y2K) confidence assessment. The method uses geometrical diagrams to help visualize the relative importance of true positives, false positives, true negatives, and false negatives, to assess the confidence of surviving the Y2K transition. When a multi-cycle resolution approach is implemented, prior Y2K confidence estimates can be revised at the completion of each cycle to reflect the most recent results of error testing activities, to obtain improved posterior probability estimates of success.

Outline

- 0 Objectives
- 0 Motivation for using Bayes' theorem
- 0 Assumptions
- 0 Multi-cycle Y2K error resolution process
- 0 Geometrical representations of Bayes' theorem
- 0 Example
- 0 Conclusions

MITRE

Acknowledgments

The author acknowledges the contributions of Dennis Mangsen, who gave much helpful guidance on project issues, and made the suggestion for developing a pictorial representation of Bayes' theorem that could provide a more intuitive way of understanding its application to confidence assessment than is found in standard textbooks on the subject. His comments about how to interpret the Bayesian pictures have been particularly insightful. Also greatly appreciated were the useful discussions with George Stark about Bayesian problem formulation and methods for utilizing available test data for confidence analysis.

Objectives

- 0 **Present work in progress by MITRE to understand Year 2000 (Y2K) error resolution confidence and related issues**
- 0 **Describe a quantitative approach based on Bayes' theorem to estimate confidence probabilities at the end of an error resolution cycle**

MITRE

A simple application of Bayes' theorem provides a method for assessing the confidence of surviving the Y2K computer date problem. Several key concepts in Bayesian analysis are introduced with geometrical diagrams to help clarify its potential application and improve understanding of the method. For reference purposes, the mathematical formulas for Bayes' theorem are given in the backup charts.

Motivation for Using Bayes' Theorem

- 0 **Estimates probabilities of four possible outcomes**
 - True positives
 - False positives
 - True negatives
 - False negatives
- 0 **Want to be able to "see where we are" after completing a given error resolution cycle**
- 0 **Bayes' theorem provides a simple quantitative method for making posterior probability assessments using the results of prior information**
 - **Applicable to multi-cycle iterative processes**
- 0 **Can use either subjective assessments or available quantitative data for prior probability inputs**

MITRE

Introduced into statistical analysis more than 200 years ago, Bayes' theorem provides formulas for adjusting prior knowledge of probabilities to reflect the impact of more than one condition being present. The Bayesian formulas are valid mathematical expressions of conditional probability theory, but many students of introductory probability and statistics often find that the application of Bayes' theorem is hard to grasp.

Assumptions

- 0 **“Resolution cycle” is the complete process of finding and fixing critical Year 2000 errors during a given error resolution iteration**
 - **Errors found by reviewing source code**
 - = **Software errors (logic, date syntax, constants, etc.)**
 - **Errors in data files**
 - **Errors in firmware/microcode/microchips in hardware**
 - **Errors induced by testing with “live data”**
- 0 **The process of searching for Year 2000 errors may be repeated as many cycles as needed to achieve the desired level of confidence**
 - **As allowed by available resources and schedule**
- 0 **For a computer program that has Year 2000 problems, we need to estimate the probability that any given line of code is infected with a Year 2000 critical error**

MITRE

For many computer systems, the goal is to survive the Year 2000 rollover by finding all the critical errors that can bring a system down or otherwise impair mission operations. It is very likely the initially planned resolution cycle will not find all the critical Y2K errors, and it will be necessary to follow up with a second Y2K error resolution cycle. Bayes' theorem can be used to estimate the confidence that errors have been found at the end of each resolution cycle. More importantly, it can also estimate the probability that undetected errors escaped the testing process. This confidence can be updated at the end of each additional cycle. Although software errors are treated in the remainder of this briefing, it is also possible to use a similar approach for assessing confidence with other types of errors found in data files, firmware/microcode, commercial off-the-shelf products, and operational tests.

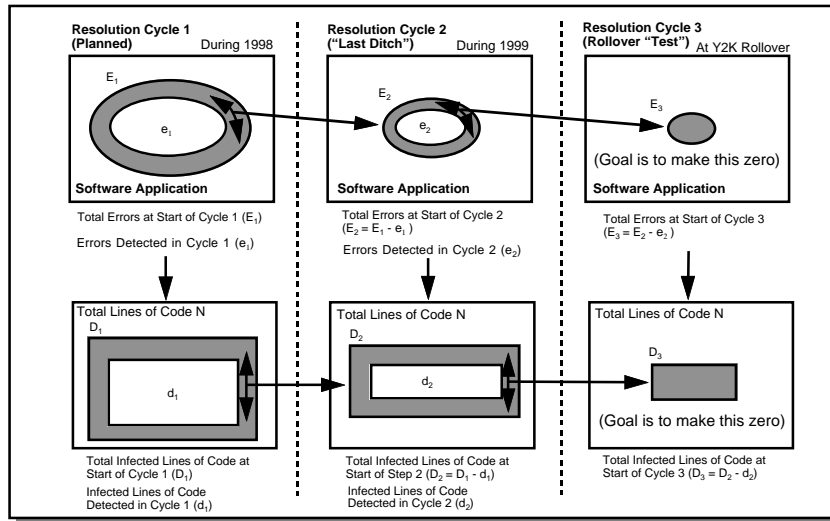
Assumptions (continued)

- 0 **Two problem resolution cycles (or more) may be needed**
 - One initial “planned” resolution cycle, ending in December 1998
 - One “last ditch sweep” before Year 2000 rollover occurs, ending in December 1999
- 0 **The probability of successful error detection after a complete problem resolution cycle must be estimated**
 - Likelihood of finding an error, given that the error actually exists (applies in theory to all errors found by the error detection process)
 - Can be measured by seeding the code or system with deliberate errors
 - Probability ranges may need to be estimated from available data or experience with the software maintenance process

MITRE

When a multi-cycle resolution process is implemented, prior Y2K confidence estimates can be revised at the end of each resolution cycle to reflect the most recent results from error testing activities, to obtain improved estimates of a successful outcome.

Multi-Cycle Y2K Error Resolution Process



MITRE

Many systems will complete their initial resolution cycle by the end of 1998. This initial cycle will most likely find and eliminate most of the Y2K threat. However, the possibility that critical Y2K errors may still exist after the initial resolution cycle is completed will make it necessary to perform one (or possibly more) "last ditch" resolution cycle in 1999 to fix remaining errors. During 1999 this "last ditch" effort must be done in earnest to reduce all critical Y2K errors to zero. The ultimate "test" will occur at Y2K rollover, where hopefully no emergency actions will be necessary to deal with crises.

Notional Logical Representation of Bayes' Theorem

		Prior Probability	
		Code is Infected A_1	Code is Not Infected A_2
Detection Likelihood	Error is Detected (B = True)	There is a code error and testing finds one "Good Hits" (True Positives)	There is not a code error and testing finds one "False Alarms" (False Positives)
	Error is Not Detected (B = False)	There is a code error and testing does not find one "Undetected Errors" (False Negatives)	There is not a code error and testing does not find one "No News is Good News" (True Negatives)

MITRE

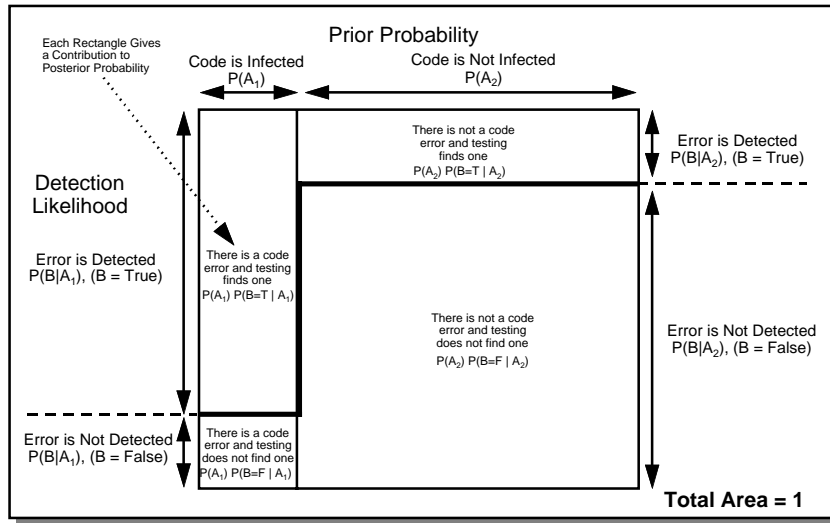
With Bayes' theorem, it is possible at the end of a given "resolution cycle" to estimate the probability of four types of outcomes, which are also called posterior probabilities. A resolution cycle is defined here to mean one of several iterations attempting to find, correct, and replace those parts of the system that are infected with critical Y2K errors. The posterior probability depends on knowing estimates of three things (also called *a priori* estimates or prior probabilities), namely:

- estimated prior probability that an error is present
- estimated likelihood that an error will be detected in the presence of an actual error
- estimated likelihood that an error will be detected in the absence of an actual error

The outcome is a true positive or "good hit" when an error is detected and the error is actually there. There will be a "false alarm" (also called false positive) when an error is detected, and no errors are actually present. An "undetected error" or false negative will occur when an error is not detected, but an error is actually present. Lastly, there are no errors (i.e., "no news is good news" or true negative) when an error is not detected and no errors are actually present.

The Y2K problem is successfully resolved for a given system when the number of remaining undetected critical errors is reduced to zero before the Y2K rollover occurs. In actual practice, it might take several resolution cycles to reduce the probability of undetected errors to an acceptably low level. Knowledge of the number of errors corrected during a previous resolution cycle can be used to revise the prior probability estimate for applying Bayes' theorem to the follow-on resolution cycle.

Notional Geometrical Representation of Bayes' Theorem



MITRE

The four possible outcomes that can be inferred by Bayes' theorem can be represented geometrically by four distinct rectangular areas, in which the sum of the areas equals one. The area of each rectangle reflects the absolute probability of the corresponding outcome. The areas of the two upper rectangles in this diagram (above the dark line) are proportional to the probability that an error is detected. The areas of the two lower rectangles (below the dark line) are proportional to the probability that an error is not detected. The areas of the two rectangles on the left side are proportional to the probability that an error is actually present. The areas of the two rectangles on the right side are proportional to the probability that an error is not actually present.

Bayes' Theorem Example — Numerical Calculation

Single Point			
Probability of Code Being Infected	10.00%		
Probability of Detecting an Error When One is Present	90%		
Probability of Detecting an Error When One is Not Present	10%		

Contribution to Posterior Probability		Code Infected	Code Not Infected	Likelihood	
P(A)	P(B A)	0.10	0.90	Error is Detected	Error is Not Detected
Error is Detected	0.90	0.0900	0.0900	0.10	0.10
Error is Not Detected	0.10	0.0100	0.8100	0.90	0.90
Total Area = 1					

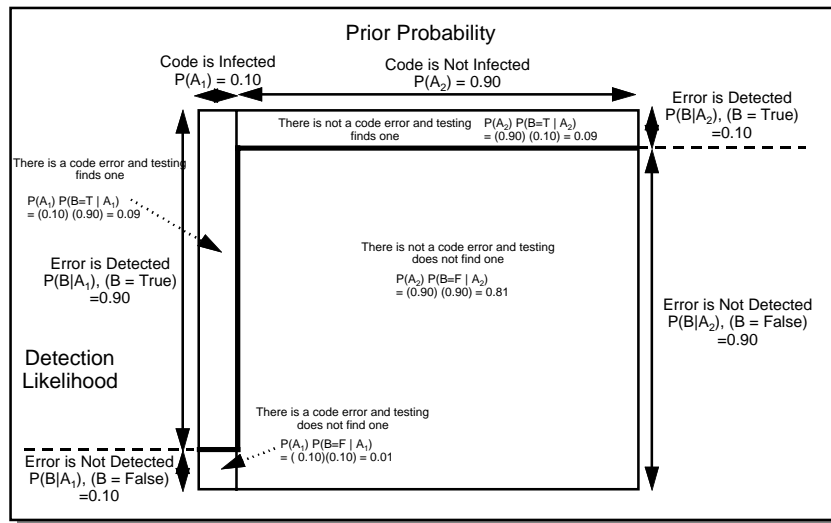
Error Detected (B = True)					
A_i	$P(A_i)$	$P(B A_i)$	$P(A_i)P(B A_i)$	$P(A_i B)$	
Code is infected (A_1)	0.10	0.90	0.0900	0.50000	Probability there is a code error, given testing finds one - $P(A_1 B=T)$ (Good Hits)
Code is not infected (A_2)	0.90	0.10	0.0900	0.50000	Probability there is not a code error, given testing finds one - $P(A_2 B=T)$ (False Alarms)
Total	1.00		0.1800	1.00000	

Error Not Detected (B=False)					
A_i	$P(A_i)$	$P(B A_i)$	$P(A_i)P(B A_i)$	$P(A_i B)$	
Code is infected (A_1)	0.10	0.10	0.0100	0.01220	Probability there is a code error, given testing does not find one - $P(A_1 B=F)$ (Undetected Errors)
Code is not infected (A_2)	0.90	0.90	0.8100	0.98780	Probability there is not a code error, given testing does not find one - $P(A_2 B=F)$ (No News is Good News)
Total	1.00		0.8200	1.00000	

The sum of these two parts should add up to one MITRE

In the example presented here, it is initially estimated *a priori* that there is a 10 percent probability that the code is infected with errors. From past knowledge of the testing process, the likelihood of detecting an error when an error is actually present is estimated to be 90 percent. The likelihood of detecting an error when an error is not actually present is estimated to be 10 percent. Using the Bayesian formulas in the backup charts of this briefing, we calculate that if an error is detected, the relative probability of a “good hit” (50 percent) equals the relative probability of a “false alarm” (50 percent). This result is surprisingly low, despite our ability to detect errors 90 percent of the time with our testing process.

Bayes' Theorem Example — Geometrical Representation



MITRE

The resulting low probability to correctly detect an error predicted by Bayes' theorem, runs counter to our intuition, which is based on our experience with the testing process. However, the explanation for this counter-intuitive result can be easily explained by the geometrical picture. The probability of a good hit is equal to the area of the upper left rectangle, whose area is given by probability of infection times the likelihood of detection, or 0.10 times 0.90, which equals 0.09. On the other hand, the probability of a false positive is equal to the area of the upper right rectangle, whose area is given by the probability that the code is not infected times the likelihood an error is detected when one is not there, or 0.90 times 0.10, which also equals 0.09. Therefore, in this example, whenever an error is detected, we are only 50 percent confident that an error is actually present, even though the likelihood of detection is very high.

Example — Results

<u>Outcome Description</u>	<u>Absolute Probability</u>	<u>Relative Confidence by Test Outcome</u>
Good Hits (True Positives)	9 %	50 % (Tests positive)
False Alarms (False Positives)	9 %	50 % (Tests positive)
<hr style="border-top: 1px dashed black;"/>		
Undetected Errors (False Negatives)	1 %	1 % (Tests negative)
No News is Good News (True Negatives)	81 %	99 % (Tests negative)
	100 %	

MITRE

The picture also reveals something even more important, namely, the rectangles representing true hits and false positives make up only a very small fraction of the total possible area. The very large rectangle at the bottom right of the picture, representing true negatives, means that when no errors are detected we have a very high probability that no errors are actually present. Finding large numbers of errors might actually be a “red herring,” if the testing process still misses the few critical Y2K undetected errors that will cause system failures and/or data corruption. The primary purpose of Y2K resolution is to reduce the number of undetected errors to zero, not to maximize the number of total errors detected. The expense of finding an error (either a true or a false error) during pre-rollover testing and correcting will be nowhere nearly as costly as the occurrence of a single catastrophic undetected error during Y2K rollover. Consequently, the scarce resources available for fixing and testing Y2K errors should be used to reduce undetected errors.

The 50 percent posterior probability estimated for both the true positives and false positives are surprisingly high, in view of the 90 percent detection likelihood. However, the probability of an undetected error is only one percent, which is a better measurement of our ability to avoid catastrophe.

Suppose the current resolution cycle reduces the error infection rate to 5 percent, and the next resolution cycle can achieve an improved error detection likelihood of 95 percent, then the resulting posterior undetected error probability predicted by Bayes’ theorem would drop to 0.25 percent, or a factor of four improvement during the next cycle. On the other hand, the posterior probability estimates for true positives and false positives would still each be 50 percent, which illustrates how looking only for positive test results can be misleading.

Conclusions

- **A method based on Bayes' theorem can be used to estimate the probability (i.e., confidence) that errors are no longer present after completing a given error resolution cycle**
- **If no errors are detected, we get a large increase in confidence that no errors are present**
- **The quality of the prior probability estimates (based on past errors detected or judgment) used by Bayesian analysis determines the "goodness" of the result**
- **False alarm rate and undetected error rate provide important insights into overall confidence**
- **Recommend that Bayesian analysis be used on systems**
 - **Find out "where you are" quickly**
 - **Ability to update posterior probabilities using results of prior iteration (e.g., adjustments as code is corrected)**

MITRE

Bayes' theorem can be useful for analyzing the confidence level associated with the process to resolve the Y2K problem. Bayesian confidence assessment can be applied at various points between now and the Y2K rollover. Each system must provide its own specific input data needed to perform the Bayesian confidence assessment, especially at the end of a completed resolution cycle. Initially the quality of the data might be low, due to the need to rely more heavily on judgment for the *a priori* estimates when there is larger uncertainty.

When input data is highly uncertain, especially earlier in the process, data should be estimated as ranges, perhaps by using the results of past testing experience or by using "gray beard" teams to assure independence and greater objectivity. Gray beard teams are used by industry and government to bring a structured review process to a particular problem or question. The team members represent an appropriate combination of expertise and experience needed to address the key issues. A key service for Y2K that could be performed by gray beard teams would be to determine the relative expected percentage mix of minor, major, and critical Y2K errors in each system. In addition, the results of test measurements can be used to estimate the *a priori* probabilities and detection likelihood needed for Bayesian analysis. Subjective assessments can be subsequently revised as experience with working the Y2K problem grows.

BACKUP CHARTS

MITRE

Bayes' Theorem Formulas

<u>Outcome</u>	<u>Description</u>	<u>Bayesian Probability Formula</u>
There is a code error and testing finds one	Good Hits (True Positives)	$P(A_1 B=T) = \frac{P(A_1) P(B=T A_1)}{P(A_1) P(B=T A_1) + P(A_2) P(B=T A_2)}$
There is not a code error and testing finds one	False Alarms (False Positives)	$P(A_2 B=T) = \frac{P(A_2) P(B=T A_2)}{P(A_1) P(B=T A_1) + P(A_2) P(B=T A_2)}$
There is a code error and testing does not find one	Undetected Errors (False Negatives)	$P(A_1 B=F) = \frac{P(A_1) P(B=F A_1)}{P(A_1) P(B=F A_1) + P(A_2) P(B=F A_2)}$
There is not a code error and testing does not find one	No News is Good News (True Negatives)	$P(A_2 B=F) = \frac{P(A_2) P(B=F A_2)}{P(A_1) P(B=F A_1) + P(A_2) P(B=F A_2)}$

MITRE

Definitions

E_k = Total errors at the beginning of Cycle k
 e_k = Number of errors found, fixed, and tested during Cycle k
 N = Total lines of code in the software application

Errors per total lines of code at the beginning of Cycle k
 $= E_k / N$

Errors per total lines of code at the end of Cycle k
 $= (E_k - e_k) / N$

MITRE