Developing an Intelligent Decision Support System for the Proactive Implementation of Traffic Safety Strategies

Final Report

Prepared by:

Hongyi Chen
Fang Chen
Chris Anderson

Department of Mechanical and Industrial Engineering
Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

CTS 13-12

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
CTS 13-12			
4. Title and Subtitle Developing an Intelligent Decision Support System for the Proactive Implementation of Traffic Safety Strategies		5. Report Date March 2013 6.	
7. Author(s) Hongyi Chen, Fang Chen, and Chris Anderson		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Mechanical and Industrial Engineering University of Minnesota Duluth 1303 Ordean Court Duluth, MN 55812		10. Project/Task/Work Unit No. CTS Project #2011015 11. Contract (C) or Grant (G) No.	
12. Sponsoring Organization Name and Address Intelligent Transportation Systems Institute Center for Transportation Studies University of Minnesota 200 Transportation and Safety Building 511 Washington Ave. SE Minneapolis, Minnesota 55455		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code	
15. Supplementary Notes http://www.its.umn.edu/Publications/ResearchReports/			

16. Abstract (Limit: 250 words)

The growing number of traffic safety strategies, including the Intelligent Transportation Systems (ITS) and low-cost proactive safety improvement (LCPSI), call for an integrated approach to optimize resource allocation systematically and proactively. While most of the currently used standard methods such as the six-step method that identify and eliminate hazardous locations serve their purpose well, they represent a reactive approach that seeks improvement after crashes happen. In this project, a decision support system with Geographic Information System (GIS) interface is developed to proactively optimize the resource allocation of traffic safety improvement strategies. With its optimization function, the decision support system is able to suggest a systematically optimized implementation plan together with the associated cost once the concerned areas and possible countermeasures are selected. It proactively improves the overall traffic safety by implementing the most effective safety strategies that meet the budget to decrease the total number of crashes to the maximum degree. The GIS interface of the decision support system enables the users to select concerned areas directly from the map and calculates certain inputs automatically from parameters related to the geometric design and traffic control features. An associated database is also designed to support the system so that as more data are input into the system, the calibration factors and crash modification functions used to calculate the expected number of crashes will be continuously updated and refined.

17. Document Analysis/Descriptors	18. Availability Statement		
Traffic safety, Decision support systems, Optimization, Geographic information systems	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312		
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 81	22. Price

Developing an Intelligent Decision Support System for the Proactive Implementation of Traffic Safety Strategies

Final Report

Prepared by:

Hongyi Chen
Fang Chen
Chris Anderson

Department of Mechanical and Industrial Engineering
Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

March 2013

Published by:

Intelligent Transportation Systems Institute
Center for Transportation Studies
University of Minnesota
200 Transportation and Safety Building
511 Washington Ave. S.E.
Minneapolis, Minnesota 55455

Acknowledgements

The authors wish to acknowledge those who made this research possible. The study was funded by the Intelligent Transportation Systems (ITS) Institute, a program of the University of Minnesota's Center for Transportation Studies (CTS). Financial support was provided by the United States Department of Transportation's Research and Innovative Technologies Administration (RITA). The project was also supported by the Northland Advanced Transportation Systems Research Laboratories (NATSRL), a cooperative research program of the Minnesota Department of Transportation, the ITS Institute, and the University of Minnesota Duluth College of Science and Engineering.

Additionally, the great suggestions, inspirations and support from Dr. Eil Kwon, NATSRL director, were highly appreciated. We thank Professor Zhuanyi Liu from the Mathematics and Statistics Department at UMD for providing valuable suggestions on the algorithm development during the early stage of this project. We would also like to acknowledge help received from traffic engineers, including Brad Estochen, Victor Lund, Brian Larson, Robert R. Edge from MnDOT, Minnesota District 1 and Saint Louis County, throughout the project as well as their support staffs that provided GIS shape files and crash data. Last but not the least, we want to thank Stacy Stark and Steve Graham from the Geographic Information Science (GIS) lab at the University of Minnesota Duluth (UMD) for allowing us to access their lab facilities and providing technical help when needed.

Table of Contents

List of Tables

List of Figures

Executive Summary

The growing number of traffic safety strategies, including the Intelligent Transportation Systems (ITS) and low-cost proactive safety improvement (LCPSI), call for an integrated approach to optimize resource allocation systematically and proactively. While most of the currently used standard methods such as the six-step method that identify and eliminate hazardous locations serve their purpose well, they represent a reactive approach that seeks improvement after crashes happen. To proactively improve traffic safety, hazardous road conditions and their expected impact should be forecast and considered in the process of deciding on the sites to be treated. With limited budget, it is desired to have the resource allocated in an optimized way to reduce the expected number of crashes to the maximum degree. Though an optimization model is provided in the *Highway Safety Manual* [1], it ignores some aspects of the problem and thus will lead to sub-optimal solutions. To address these issues and assist decision makers at different levels to select treatment sites and the corresponding safety strategies proactively, an intelligent decision support system is proposed in this project.

Based on detailed review of the traditional methods and comparisons with one another, new methods are developed in this project. We developed an optimization model that minimizes the total expected number of crashes while satisfying all the budget constraints. Combining the traditional methods and extrapolation, we provide a three-step method to forecast the expected number of crashes of a treatment site with and without certain treatment. This provides input to the optimization model. In situations where data is limited to conduct a statistical analysis, we use the crash modification functions provided in the *Highway Safety Manual* [1] to forecast the number of crashes. But as more data are being collected and stored in the decision support system's database, we expect to keep refining the calculation of key parameters such as the calibration factor used in the manual.

Also, as data collection and preparation are usually the most difficult and time consuming tasks during the entire decision-making process, we proposed and developed a GIS (Geographic Information Systems) based tool to facilitate the data collection, information extraction and calculation efforts in such process. With a GIS interface, the decision support system enables the traffic engineers to select the sites under consideration directly from the GIS map, review the site information such as the road identification (ID) and Annual Average Daily Traffic (AADT), and calculate parameters related to the geometric design features and traffic control features. The GIS tool eliminates the need to manually input a major part of the required data to calculate the crash modification factors and thus saves the traffic engineers tremendous time. With the decision support system proposed in this project, once the treatment sites under considerations and their possible countermeasures are selected from the map and a total budget is input, optimized scenario(s) can be generated to suggest the sites to be treated together with their corresponding countermeasures.

Future benefits of this project include the possibility to connect the MnDOT GIS database with the Safety Analyst software and complement its incomplete optimization function with the one proposed in this project. The data collection and preparation time is expected to be shortened tremendously as many of the inputs required by the software will be generated automatically by our GIS tool. Accuracy of the optimization decision can also be improved.

Chapter 1. Introduction

The growing number of traffic safety strategies, including the Intelligent Transportation Systems (ITS) and low-cost proactive safety improvement (LCPSI), call for an integrated approach to optimize resource allocation systematically and proactively. While the currently used standard methods such as the six-step method that identify and eliminate hazardous locations serve their purpose well, they represent a reactive approach that seeks improvement after crashes happen. To assist decision makers at different levels to select treatment sites and the corresponding safety strategies proactively, an intelligent decision support system is proposed in this project. The system will help assess the effectiveness of individual as well as combined traffic safety strategies under different road and weather conditions, identify possible dependencies among safety improvements in related locations, and optimize the implementation of traffic safety strategies in a proactive way. Literature in related areas was reviewed and an optimization model was developed. The algorithms were symbolically deducted and programmed. Therefore, as more data are collected and input to the system, the assessment results for existing and emerging traffic safety strategies will be continuously improved.

During the early stage of the project, a case study was conducted to help analyze the budget allocation problem faced by District 1 engineers to determine an optimal solution to distribute 22 million dollars to two competing safety improvement projects on highway 169 in the Ely area in Minnesota in fall 2010. During this case study, data collection and information input were identified as the most time consuming tasks. Part of the problem was due to that data needed for analysis were stored in different formats at various agents. To help organize data in a uniform format and accelerate the analysis process, we propose a Geographic Information Systems (GIS) based decision support system in this project. Using this system, data capturing the characteristics of road segments/intersections as well as crashes can be stored in shapefiles and linked to the system to calculate the crash rate automatically without the need of manual input. In addition, the GIS interface enables the users to select concerned areas directly from the map, view the related information, conduct analysis, and calculate the expected crash rates. Once an estimated budget is input, optimized safety strategy implementation plans for the chosen areas will be suggested, together with the total investment needed for implementation, operation and maintenance.

This report is organized in the following way: Chapter 2 reviews the traditional methods used in site selection and countermeasure evaluation. It starts with the six-step method, a standard procedure, and the optimization process proposed in the *Highway Safety Manual* [1] that identifies and eliminates hazardous locations to discuss their use and shortcomings. Then, several methods that assess the crash reduction effect of a treatment were discussed in detail with a comparison of their advantages, disadvantages, and data needs. Those methods include (1) the simple before-and-after study method, (2) the before-and-after study with Comparison Group(CG) method, (3) the before-and-after study with the Empirical Bayes (EB) method, and (4) the more coherent method introduced by Hauer [2]. Examples and mathematical deduction were given to illustrate and verify the methods especially in cases where the original sources provide either incomplete or incorrect deduction. In Chapter 3, an optimization model was proposed to proactively and systematically allocate resources for traffic safety strategies. It addresses the problems identified in Chapter 2 when reviewing the optimization model proposed

in the *Highway Safety Manual* [1] and provide a proactive way for traffic safety improvement. Then a three-step method is introduced with detailed examples and illustrations to provide input for the optimization model. The three steps are: (1) Data collection and preparation; (2) Forecast the expected accident count of treatment site without treatment; and (3) Forecast the expected accident count of treatment site after treatment. Limitations that call for attention and future work of the algorithm are also discussed. Chapter 4 shows the entire picture of the GIS interfaced decision support system to be developed based on algorithms reviewed and developed in this report. Example codes are provided with detailed analysis. Chapter 5 concludes the project.

Next, we review and discuss in details the traditional methods used in site selection and countermeasure evaluation.

Chapter 2. Site and Countermeasure Evaluation and Selection Methods

The standard procedure for identifying and eliminating hazardous locations, or "hot spots," which are locations with relatively high crash rate, consists of the following six steps [3]:

1. Indentify the highly hazardous locations according to crash reports;
2. Analyze the potential design problems for these locations;
3. Indentify feasible countermeasures to deal with the design problems;
4. Predict the effect of the potential countermeasures according to the crash reduction number;
5. Implement the countermeasures with the highest cost effectiveness ratio;
6. Estimate the effect of the countermeasure after implementing the countermeasures.

This process of improving traffic safety is by far the most pervasive process in real practice. It is straightforward and easy to use. However, it represents a reactive decision-making process, which searches for remedy after traffic accident happens. An effective system should make predictive analysis beforehand and choose to implement the best possible countermeasure(s) to prevent accidents from happening. Also, at the fourth step, the "cost effectiveness ratio" is the only criterion used to select countermeasures. This represents a lack of systematic optimization and long term planning in the resource allocation decision.

A more proactive approach to deal with resource allocation and countermeasure selection was recently published by the American Association of State Highway Transportation Officials (AASHTO) in the *Highway Safety Manual* [1]. Based on a large quantity of literature, it developed the method to predict the expected average crash frequency by facility and site types for rural two-lane, two-way roads and rural multilane highways. The white papers on the Safety Analyst software also documented the optimization process that the software is based on to help select countermeasures for a group of concerned facilities and sites.

The optimization process proposed in the *Highway Safety Manual* [1] and used in the Safety Analyst software suggests the following objective function that maximizes the total benefit, quantified in dollars, for the countermeasure implementation decision.

Maximize $TB = \sum_{j=1}^{y} \sum_{k=1}^{z} (PSB_{jk} - CC_{jk}) X_{jk}$

Where X_{jk} is a value indicating whether countermeasure k at site j is selected, PSB_{jk} is the present value of safety benefits of countermeasure k at site j, and CC_{jk} is the construction cost for countermeasure k at site j.

Maximizing the total net benefit is effective in the way that the construction cost of the countermeasures are considered and included in the optimization. However, potential problems exist in this approach. Suppose a highly hazardous site involves most of the fatal accidents can be best treated by a high cost countermeasure and several less hazardous sites can be incrementally improved by a few low-cost countermeasures. There is a good chance that the

differences of the benefits and the costs, or the net benefits, of treating the highly hazardous site and the few less hazardous sites are the same. Using the above objective function will lead to a conclusion that is indifferent between treating the highly hazardous site and incrementally improving the group of less hazardous sites. To avoid this situation, alternative solutions need to be forced from the optimization model and additional analysis will be needed to prioritize the sites. Another issue associated with this optimization model is that the present value of the net benefit is calculated using the projected service life of the facility, which is usually tens of years, as the discounting period. This means that the benefit, or the reduced number of crashes for the treatment site, is predicted for the next 10 to 30 years and is included in the calculation. How reliable the forecasted benefit will be in such a long time period becomes a major concern. The forecasted benefit in the long term may need to be depreciated as compared to the benefit forecasted for the near future. But the current objective function does not address this concern. A third problem with this model is that it assumes a linear additive relationship among the benefits of the countermeasures at each treated site. This may not be true in many situations: the net benefit brought to a treated site maybe less than the sum of the individual benefits most of the time and greater if synergy has been achieved by the individuals. These issues will be addressed in this project as we propose a new optimization model in Chapter 3.

To assess the crash reduction effect of a treatment, several methods have been proposed [1-4][7]. The standard way is to perform a before-and-after study. With this method, a measure of the accident prior to the treatment(s) is obtained and compared with a similar measure obtained after the implementation of the countermeasure to estimate the effectiveness of this countermeasure. Two main tasks need to be carried out in the before and after studies: predicting the number of crashes at a treatment site in the after period had the treatment not been implemented, and estimating the number of crashes at the same site in the same period after treatment is implemented. The following four steps are usually followed to calculate the necessary values [2] (a "^" is a symbol represents the estimated value):

1. Estimate the expected reduction in the number of accident count of a specific entity, denoted as $\hat{\delta}$, and calculated as

$$\hat{\delta} = \hat{\pi} - \hat{\lambda} \text{ [2]} \qquad (2.1)$$

where $\hat{\pi}$ is the estimated expected number of accidents of this entity in the after period with no treatment, and $\hat{\lambda}$ is the estimated expected number of accident of this entity in the after period after treatment.

2. Estimate the variance of $\hat{\delta}$, calculated as

$$\hat{\text{var}}(\hat{\delta}) = \hat{\text{var}}(\hat{\pi}) + \hat{\text{var}}(\hat{\lambda}) \text{ [2]} \qquad (2.2)$$

as $\hat{\pi}$ and $\hat{\lambda}$ are statistically independent.

3. Estimate the crash reduction factor, $\hat{\theta}$.

The crash reduction factor is defined as the expected accident count in the after period after treatment divide by the expected accident count in the after period without treatment. Namely, $\theta = \frac{\lambda}{\pi}$. Intuitively, it can be calculated by $\hat{\theta}^* = \frac{\hat{\lambda}}{\hat{\pi}}$. However, although $\hat{\pi}$ and $\hat{\lambda}$ are unbiased estimates of π and λ respectively, the faction $\frac{\hat{\lambda}}{\hat{\pi}}$ is biased estimation of θ. An approximately unbiased estimate is given by

$$\hat{\theta} = \frac{\hat{\lambda}}{\hat{\pi}} / [1 + \hat{\mathrm{var}}(\hat{\pi})/\hat{\pi}^2] \quad [2] \tag{2.3}$$

where $[1 + \hat{\mathrm{var}}(\hat{\pi})/\hat{\pi}^2]$ serves as a correction factor to remove the bias. Since [2] provides an incomplete proof, we give a more accurate proof below.

Proof of equation (2.3) as an unbiased estimate of θ:

Since $\hat{\theta}^*$ is a function of $\hat{\pi}$ and $\hat{\lambda}$, denoted by $\hat{\theta}^* = f(\hat{\pi}, \hat{\lambda}) = \frac{\hat{\lambda}}{\hat{\pi}}$. Using the Taylor Expansion, we have:

$$f(\hat{\pi}, \hat{\lambda}) \triangleq f(\pi, \lambda) + \frac{(\hat{\pi}-\pi)}{1!} * \frac{\partial f(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}}\Bigg|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + \frac{(\hat{\lambda}-\lambda)}{1!} * \frac{\partial f(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}}\Bigg|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + \frac{(\hat{\pi}-\pi)^2}{2!} * \frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}^2}\Bigg|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + \frac{(\hat{\lambda}-\lambda)^2}{2!} * \frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}^2}\Bigg|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}}. \tag{2.4}$$

Then $E(\hat{\theta}^*) = E\left(f(\hat{\pi}, \hat{\lambda})\right) \triangleq f(\pi, \lambda) + E(\hat{\pi} - \pi) * \frac{\partial f(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + E(\hat{\lambda} - \lambda) * \frac{\partial f(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + \frac{1}{2} * E(\hat{\pi} - \pi)^2 * \frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}^2}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}} + \frac{1}{2} * E(\hat{\lambda} - \lambda)^2 * \frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}^2}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}}$

Since $(\frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}^2}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}}) = \frac{2\hat{\lambda}}{\hat{\pi}^3}$, $(\frac{\partial^2 f(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}^2}\Big|_{\substack{\hat{\pi}=\pi \\ \hat{\lambda}=\lambda}}) = 0$, $E(\hat{\pi} - \pi)^2 = \mathrm{var}(\hat{\pi} - \pi) - (E(\hat{\pi} - \pi))^2 = \mathrm{var}(\hat{\pi})$, and $E(\hat{\lambda} - \lambda)^2 = \mathrm{var}(\hat{\lambda} - \lambda) - (E(\hat{\lambda} - \lambda))^2 = \mathrm{var}(\hat{\lambda})$,

therefore

$$E(\hat{\theta}^*) = \frac{\lambda}{\pi} * \left[1 + \frac{\mathrm{var}(\hat{\pi})}{\pi^2}\right]. \tag{2.5}$$

Equation (2.5) indicates that if we estimate θ by $\hat{\theta}^* = \frac{\hat{\lambda}}{\hat{\pi}}$ repeatedly, the result would be greater than the actual value by a factor of $[1 + \frac{\mathrm{var}(\hat{\pi})}{\pi^2}]$. Therefore the correction factor need to be introduced into the expression which $\hat{\theta} = \frac{\hat{\lambda}}{\hat{\pi}} / [1 + \hat{\mathrm{var}}(\hat{\pi})/\hat{\pi}^2]$.

4. Estimate the variance of $\hat{\theta}$, which is calculated as

$$\hat{\mathrm{var}}(\hat{\theta}) = (\frac{\hat{\lambda}}{\hat{\pi}})^2 \left[\frac{\hat{\mathrm{var}}(\hat{\lambda})}{\hat{\lambda}^2} + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}\right] / [1 + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}]^2 \text{ [2].} \qquad (2.6)$$

Note in [2], (2.6) is written as

$\hat{\mathrm{var}}(\hat{\theta}) = (\hat{\theta})^2 \left[\frac{\hat{\mathrm{var}}(\hat{\lambda})}{\hat{\lambda}^2} + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}\right] / [1 + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}]^2$, but it is not accurate. Since [2] did not provide the proof of equation (2.6), we prove it here:

<u>Proof of (2.6)</u>

According to the previous deduction of (2.5), the question now will become that what will happen when we estimate θ by $\hat{\theta} = \frac{\hat{\lambda}}{\hat{\pi}} / [1 + \hat{\mathrm{var}}(\hat{\pi})/\hat{\pi}^2]$?

$\hat{\theta}$ is a function of $\hat{\pi}$ and $\hat{\lambda}$, denote by $\hat{\theta} = g(\hat{\pi}, \hat{\lambda}) = \frac{\hat{\lambda}}{\hat{\pi}} / [1 + \hat{\mathrm{var}}(\hat{\pi})/\hat{\pi}^2]$.The value of $[1 + \mathrm{var}(\hat{\pi})/\pi^2]$ is usually close to 1.Representing this value as a constant "a", then $\hat{\theta} = \frac{\hat{\lambda}}{\hat{\pi}}/a$. By equation (2.4)

$$\mathrm{Var}(\hat{\theta}) = 0 + \mathrm{Var}(\hat{\pi} - \pi) * (\frac{\partial g(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}})^2 \Big|_{\substack{\hat{\pi} = \pi \\ \hat{\lambda} = \lambda}} + \mathrm{Var}(\hat{\lambda} - \lambda) * (\frac{\partial g(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}})^2 \Big|_{\substack{\hat{\pi} = \pi. \\ \hat{\lambda} = \lambda}}$$

Since $\frac{g(\hat{\pi},\hat{\lambda})}{\partial \hat{\pi}} = -\frac{\hat{\lambda}}{\hat{\pi}^2 a}$, $\frac{g(\hat{\pi},\hat{\lambda})}{\partial \hat{\lambda}} = \frac{1}{\hat{\pi} a}$, then

$$\mathrm{Var}(\hat{\theta}) = (-\frac{\lambda}{\pi^2 a})^2 * \mathrm{Var}(\hat{\pi}) + (\frac{1}{\pi a})^2 * \mathrm{Var}(\hat{\lambda}) = (\frac{\lambda}{\pi})^2 \left[\frac{\mathrm{var}(\hat{\pi})}{\pi^2} + \frac{\mathrm{Var}(\hat{\lambda})}{\lambda^2}\right] / a^2$$

Therefore $\hat{\mathrm{var}}(\hat{\theta}) = (\frac{\hat{\lambda}}{\hat{\pi}})^2 \left[\frac{\hat{\mathrm{var}}(\hat{\lambda})}{\hat{\lambda}^2} + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}\right] / [1 + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}]^2$.

The formula in [2] is written as $\hat{\mathrm{var}}(\hat{\theta}) = (\hat{\theta})^2 \left[\frac{\hat{\mathrm{var}}(\hat{\lambda})}{\hat{\lambda}^2} + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}\right] / [1 + \frac{\hat{\mathrm{var}}(\hat{\pi})}{\hat{\pi}^2}]^2$, which is inaccurate.

To get these measurements, three types of methods are used to perform the before-and-after studies [4]. They are:

1. The simple before-and-after study method;
2. The before-and-after study with Comparison Group(CG) method;
3. The before-and-after study with the Empirical Bayes (EB) method.

Next, we discuss each of them in details.

2.1 Simple before-and-after study method

The essence of the simple before-and-after study is based on the assumption that the number of the crashes in the after period is expected to be the same as the before period if no improvement has been made [2]. To apply this method, we use the accident count before implementation to estimate what would have happened during the after period had the treatment not been implemented. For example, consider a simple before and after study with 100 accidents count in the before year and 66 accident count in the after year after treatment. By the assumption, if no treatment has been implemented, we would expect the same number of accident count in the after period. Hence $\hat{\pi} = 100$. Using the Four-Step, the effect of the treatment would be estimated to be $\hat{\delta} = \hat{\pi} - \hat{\lambda}$ =100-66=34 accidents. Also assume the happen of the accident is Poisson distributed, therefore, $\hat{\text{var}}(\hat{\delta}) = \hat{\text{var}}(\hat{\pi}) + \hat{\text{var}}(\hat{\lambda}) = 100 + 66 = 166$. Also,

$$\hat{\theta} = \frac{\frac{\hat{\lambda}}{\hat{\pi}}}{\left[1+\frac{\hat{\text{var}}(\hat{\pi})}{\hat{\pi}^2}\right]} = \frac{66}{100}\frac{1}{\left[1+\frac{100}{100^2}\right]} = 0.6666,$$

$$\hat{\text{var}}(\hat{\theta}) = \frac{(\frac{\hat{\lambda}}{\hat{\pi}})^2\left[\frac{\hat{\text{var}}(\hat{\lambda})}{\hat{\lambda}^2}+\frac{\hat{\text{var}}(\hat{\pi})}{\hat{\pi}^2}\right]}{\left[1+\frac{\hat{\text{var}}(\hat{\pi})}{\hat{\pi}^2}\right]^2} = 0.66^2\left[\frac{66}{66^2}+\frac{100}{100^2}\right] = 0.0110.$$

The logic of the simple before-and-after study method is straight forward and easy to use. However, this method ignores several factors including regression to the mean, crash migration, maturation, as well as external causal factors [3] that can distort the estimates and lead to an inaccurate estimate.

Regression to the mean (RTM) is a statistical phenomenon that can make natural variation in repeated data looks like real change. It happens when unusual large or small measurements tend to follow measurements that are close to the mean [5]. This problem is the most frequently cited problem in before-and-after studies. Usually the location with the large number of crashes will be selected for treatment. Because of the existence of RTM, the extreme crash frequencies would likely be followed by less extreme values even when the countermeasure is completely ineffective. In this case, we may overestimate the effectiveness of the improvement. The following examples will show how the regression to the mean can affect the result of the analysis.

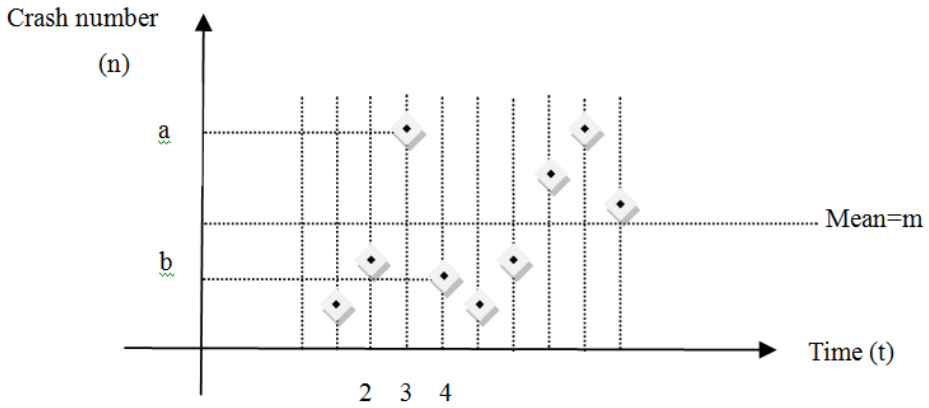

Figure 2.1: An Example Showing the RTM Effect

Figure 2.1 shows a pattern called the regression to the mean effect in transportation studies. Suppose in a certain location i, we manually divide the before period and the after period by the beginning of year 3, and each period a year long. (Before period- beginning of year 2 to beginning of year 3; after period-beginning of year 3 to beginning of year 4) The variation of the accident frequencies in years represents the natural change of the number of crashes around the mean. Notice that no treatment has been implemented in this location throughout the study period. At the end of year 2, we observe accident count a. At the end of year 3, the observation is b and b<a. By the logic of simple before and after method, we would estimate the expected accident count to be "a" in the after period if there was no treatment. However, the actual value is b. This difference indicates that even if the treatment is totally ineffective, there could still be reduction in accident number. As a result, we may overestimate (or underestimate) the effect of the treatment and produce an inaccurate estimate result.

This issue is illustrated with a specific example: Imagine a city with 100 two lane highways segments that were equipped with rumble strips at the end of year 2005. Assume that for each of these road segments, the expected number of accidents is 3 at year 2005. In practice, the observations would vary from site to site. Suppose that the accident count was Poisson distributed with mean3, then $f(K|\kappa = 3) = \frac{e^{-3} * 3^K}{K!}$. The observation values can be estimated. For example, the probability that a site would expect to have 0 accidents at year 2005 would be $f(K = 0|\kappa = 3) = \frac{e^{-3} * 3^0}{0!} = 0.0498$ and the expected number of sites with 0 accidents can be estimated by $0.0498 * 100 \triangleq 5$. Any other probabilities with certain accident counts can be calculated in the same fashion. Table 2.1 shows a list of results:

Table 2.1: Expected number of sites with K accidents

Accident Count(K)	Prob. That site has K accidents	Expected number of sites with K accidents
0	.0498	5
1	.1494	15
2	.2240	22
3	.2240	22
4	.1680	17
5	.1008	10
6	.0504	5
7	.0216	2
8	.0081	1
9	.0027	1
10 and above	.0008	-

At the end of year 2005, the transportation planners decided to equip the rumble strips on the road segments with accident count greater than 5. Table 2.1 shows that there are (10+5+2+1+1=19) road segments in total that recorded more than 5 accidents in year 2005. Suppose this strategy reduces the correctable accident count by 10%. Then one would estimate there are $(0.9 * 3 = 2.7)$ accidents expected to occur on each of these 19 road segments. However, have we estimated by the simple before and after study method, the expected number of before accident counts would be (5*10+6*5+7*2+8*1+9*1=111), and the expected number of after accident counts would be: (2.7*19=51.3). Hence, the estimated crash reduction would be ($\hat{\theta}$ =(51.3/111)/(1+1/111)=0.46). The crash reduction factor appears to be (1-0.46=54%). However, there is only 10% of reduction. The difference comes from the fact that this 19 road segments had recorded an unusually high number of accidents. Actually, this is exactly why they were selected to get improved. In this example, the existence of RTM tends to be overestimate the effect of the traffic safety strategies.

The pervious example shows how the RTM Phenomenon will become a disturbing factor. In fact, it is the most pervasive problem in before-and-after studies and needs to be settled properly. Other than the RTM effect, the potential existence of other disturbing factors can also be destructive and attention needs to be paid to them.

Crash migration is the phenomenon that the crash rate or crash severity apparently raises at the untreated sites but adjacent to treated sites as a result of the treatment [3]. When crash migration occurs, crash rates in the treated sites may decline whereas they may increase in the surrounding area. Boyle and Wright (1984) first pointed out the potential existence of the crash migration [6]. Many researchers tried to demonstrate the existence of this phenomenon; some find no evidence to support it. Elvik (1997), reviewing United Kingdom and other accident studies, found that very little of accident reduction could be directly attributed to this factor [7]. If this was a genuine effect, attention should be paid to individual links within a roadwork, which can help avoid missing potential system-wide effects.

Maturation refers to the effect of collision trends over time [8]. For example, in a treated site, the crash frequency is reduced between the before and the after period, this change could fully or

partially be due to an extension of a continuing decreasing trend which has been occurring for years. Maturation could be another disturbing factor in the study that it overestimates the effectiveness of the improvement. The difference between maturation and regression to the mean is that maturation occurs due to change in external factors such as traffic flow, economy and weather conditions. While regression to the mean is merely a statistical phenomenon that occurs whenever you have a nonrandom sample from a population and two measures that are imperfectly correlated [9].

2.2 The before-and-after study with comparison group method

The simple before and after study cannot distinguish between what is caused by the treatment and what is caused by many other influences such as regression to the mean, maturation and crash migration. The comparison group method was developed to solve the maturation and external causal factors. This method can potentially provide more accurate estimates than the simple before and after method.

The comparison group is a group of sites that have similar traffic or geometric conditions as the treated sites [2]. Conceptually, the comparison group method estimates the number of crashes that would have been occurred if no improvements have been made at a treatment site in the after period. Hauer claims that this method is based on two assumptions [2]: First, the sundry factors that affect safety have changed from the before to the after period in the same manner on both the treatment and the comparison group. Second, this change in the sundry factors influences the safety of both groups in the same way.

The comparison group method is based on the hope that, without the implementation of the treatment, the ratio of the accident count of the before and after period in the treatment site should be the same as in the comparison group. In the table below K, M, L, N denote the observed accident count in different periods, and $\kappa, \mu, \lambda, \nu$ denote the corresponding expectation values [2].

Table 2.2: Observed and expected accident counts in comparison group method

Accident count and expected values	Treatment Group	Comparison Group
Before	K, κ	M, μ
After w/o treatment	π	N, ν
After w treatment	L, λ	--

Define: $r_c \cong \nu/\mu$ to be the ratio of the expected accident count for the comparison group.
$r_t \cong \pi/\kappa$ to be the ratio of the expected accident count for the treatment group.

Our hope is $r_c = r_t$. Hence, $\pi = r_t\kappa = r_c\kappa$. The estimate of π now requires information about r_c and κ. The algorithm for comparison before and after study method is listed below. The proof of the before and after study with comparison group method is similar to the simple before and after study method.

$\hat{\lambda} = L$ [2] (2.7)

$$\hat{r}_t = \hat{r}_c = \frac{\frac{N}{M}}{1+\frac{N}{M}} \text{ [2]} \quad (2.8)$$

$$\hat{\pi} = \hat{r}_c * K \text{ [2]} \quad (2.9)$$

$$\hat{var}(\hat{\lambda}) = L \text{ [2]} \quad (2.10)$$

$$\frac{\hat{var}(\hat{r}_t)}{\hat{r}_t^{\,2}} = \frac{1}{M} + \frac{1}{N} + \hat{var}\left(\frac{r_c}{r_t}\right) \text{ [2]} \quad (2.11)$$

$$\hat{var}(\hat{\pi}) = \hat{\pi}^2 \left[\frac{1}{K} + \frac{\hat{var}(\hat{r}_t)}{\hat{r}_t^{\,2}}\right] \text{ [2]} \quad (2.12)$$

Equations (2.7)-(2.12) serve as building blocks for the Four-Step process. From them we get the estimates of λ and π and their variance. To finish the comparison before-and-after study, we then need to follow the four steps listed in equations (2.1)-(2.6). The detailed proof for equations (2.7)-(2.12) can be found in book [2], from page 125 to page 127.

The before-and-after study with comparison group method is considered a better approach than the simple method because it accounts the effect of maturation. However, the accuracy of this method highly depends on the availability of comparison sites and the similarity between the comparison and the treatment site [3].

2.3 The before-and-after study with the Empirical Bayes (EB) method

First, we would like to introduce the fundamental concepts of the Bayes and Empirical Bayes Theorem. In probability theorem, Bayes Theorem shows the relationship between the conditional probability and its inverse. Bayesian analysis depends on a prior distribution. The Empirical Bayes approach uses the observed data to estimate the prior and then proceeds as though the prior was known [10]. Below is the definitions and Bayes theorem which serves as a core concept of the case study are listed [11].

Definition 1

If T, is a statistic, $T = t(x_1, x_2, \ldots x_n)$, is an estimator of $\tau(\theta)$, then the loss function $L(T;\theta) \geq 0$ for all t, and $L(T;\theta) = 0$ when $t = \tau(\theta)$.

Definition 2

The risk function is defined as the expected loss $R_T(\theta) = E_{x|\theta}[L(T;\theta)]$.

Definition 3

For a random sample from f(x; θ), the Bayes Risk of a estimator T related to a risk function $R_T(\theta)$ and pdf $p(\theta)$ is the average risk with respect to $p(\theta)$

$A_T = E[R_T(\theta)] = \int R_T(\theta)\, p(\theta) d\theta.$

Definition 4

For a random sample from f(x; θ), the Bayes Estimator T^* relative to the risk function $R_T(\theta)$ and pdf $p(\theta)$ is the estimator with respect to the minimum expect risk

$A_{T^*} = E[R_{T^*}(\theta)] \leq A_T.$

Definition 5

The conditional density θ given the sample observations $X= (x_1, x_2, \ldots x_n)$ is called the posterior density(pdf), is given by:

$$f(\theta|x_1, x_2, \ldots x_n) = \frac{f(\theta, x_1, x_2, \ldots x_n)}{f(x_1, x_2, \ldots x_n)} = \frac{f(x_1, x_2, \ldots x_n|\theta)p(\theta)}{\int f(x_1, x_2, \ldots x_n|\theta)p(\theta)d\theta}.$$

For a single observation

$$f(\theta|x) = \frac{f(x|\theta)p(\theta)}{\int f(x|\theta)p(\theta)d\theta}.$$

Bayes Theorem

The Bayes estimator T, of $\tau(\theta)$ under the squares error loss function, $L(t; \theta) = [t - \tau(\theta)]^2$ is the conditional mean of $\tau(\theta)$ relative to the posterior distribution.

$T^* = E_{\theta|x}[\tau(\theta)] = \int \tau(\theta) f(\theta|x) d\theta.$

Proof

$$A_T = \int R_T(\theta)p(\theta)d\theta = \iint [T - \tau(\theta)]^2 f(x|\theta)p(\theta)dxd\theta$$

$$= \int [T^2 \int f(\theta|x)p(\theta)d\theta - 2T \int \tau(\theta)\, f(x|\theta)p(\theta)d(\theta) + \int \tau^2(\theta) f(x|\theta)p(\theta)d\theta]dx$$

$$= \int \{ \int f(x|\theta)p(\theta)d\theta [T - \frac{\int \tau(\theta)f(x|\theta)p(\theta)d\theta}{\int f(x|\theta)p(\theta)d\theta}]^2 - \frac{(\int \tau(\theta)f(x|\theta)p(\theta)d\theta)^2}{\int \tau(\theta)f(x|\theta)p(\theta)d\theta} + \int \tau^2(\theta) f(x|\theta)p(\theta)d\theta \} dx$$

Then A_T is minimized when

$$T^* = \frac{\int \tau(\theta) f(x|\theta) p(\theta) d\theta}{\int f(x|\theta) p(\theta) d\theta} = \int \tau(\theta) f(\theta|x) d\theta.$$

Before and after study with Empirical Bayes method

The most critical part in the before-and-after study is to estimate what would have been the crash frequency if there were no treatments implemented in the after period, which is denoted by $\hat{\pi}$ in the Four Step. Both of the simple and the CG before and after approach are based on the assumption that for any treated entity, the before accident count K is a sensible estimate for the expected after accident count κ with no treatment. This is not necessarily the case. If an entity is treated for its unusually high accident count, then this accident count would not be a good estimate of its expected accident count κ in the after period. The reason is statistically straight forward, since an unusual accident count cannot be a good estimate for the usual case. In the transportation safety studies, a traffic entity is more likely to be treated due to unusually high accident count [2]. This causes the so called "selection bias" or "regression to the mean" bias.

The Empirical Bayes approach is designed to eliminate the regression to the mean bias. The essence of the EB approach is that it uses two different kinds of clues to estimate the safety of an entity [2]. Clues of the first kind contain in the traits of the safety entity. A few are the traffic flow, road condition, weather condition. Clues of the second kind are derived from the history of accident occurrence, including the number of accidents.

To use both clues to estimate κ for a certain entity by EB method, first identify which reference population that the entity belongs to—the entities that have expected number of accident count κ in the after period with mean $E(\kappa)$ and variance $Var(\kappa)$; Second, select the entities from the reference population that record K accidents in the before period. Let $E(\kappa|K)$ and $Var(\kappa|K)$ denote the mean and the variance in this "sub-population." [2]

The steps to apply EB method is listed as follow. Intuitively, $E(\kappa|K)$ will be decided by both $E(\kappa)$ and K. Actually, the value of $E(\kappa|K)$ is a combination of $E(\kappa)$ and K, which will be proved later that

$$E(\kappa|K) = \alpha E(\kappa) + (1-\alpha)K \text{ [2]} \qquad (2.13)$$

In this expression α is a number between 0 and 1. To estimate the κ of the entity with maximum precision

$$\alpha = \frac{1}{1+\frac{Var(\kappa)}{E(\kappa)}} \text{ [2]} \qquad (2.14)$$

Two assumptions are listed below in order to get the parameter α in our case.

1. 1. The expectation of crash counts is gamma distributed, which is $g(\kappa) = \frac{a^b}{\Gamma(b)} \kappa^{b-1} e^{-a\kappa}$

where $E(\kappa) = \frac{b}{a}$, $Var(\kappa) = \frac{b}{a^2}$ [2].

2. The observation of the crash counts given its expectation is Possion distributed, denote by $\pi(K|\kappa) = \frac{\kappa^K e^{-\kappa}}{K!}$ [2].

Hauer did not give a complete deduction in his book. A theoretical deduction about the formula $E(\kappa|K) = \alpha E(\kappa) + (1-\alpha)K$[2] is shown below:

By Definition 5,

$$f(\kappa|K) = \frac{P(K|\kappa)*P(\kappa)}{P(K)} = \frac{\pi(K|\kappa)*g(\kappa)}{\int \pi(K|u)*g(u)du}.$$

and according to the assumptions,

$$f(\kappa|K) = \frac{\pi(K|\kappa)*g(\kappa)}{\int \pi(K|u)*g(u)du} = \frac{\frac{\kappa^K*e^{-\kappa}}{K!}*\frac{a^b}{\Gamma(b)}\kappa^{b-1}e^{-a\kappa}}{\int_0^\infty \frac{u^K*e^{-u}}{K!}*\frac{a^b}{\Gamma(b)}u^{b-1}e^{-au}du} = \frac{\kappa^K*e^{-\kappa}*\kappa^{b-1}e^{-a\kappa}}{\int_0^\infty u^K e^{-u}*u^{b-1}e^{-au}du} = \frac{\kappa^{K+b-1}*e^{-\kappa(1+a)}*(1+a)^{K+b}}{\int_0^\infty [u(1+a)]^{K+b-1}*e^{-u(1+a)}du(1+a)} = \frac{(1+a)^{K+b}*\kappa^{K+b-1}*e^{-K(1+a)}}{\Gamma(K+b)}$$

Therefore, the expected accident count given its observation is $GAM(\frac{K+b}{1+a}, \frac{K+b}{(1+a)^2})$.

Let $\alpha = \frac{1}{1+\frac{Var(\kappa)}{E(\kappa)}} = \frac{a}{1+a}$, then $E(\kappa|K) = \frac{K+b}{1+a} = \alpha E(\kappa) + (1-\alpha)K$, and $Var(\kappa|K) = (1-\alpha)E(\kappa|K)$.

Had we estimated κ in usual way, using only the history of its accident occurrence K, the value of α should be 0. And by the formula $Var(\kappa|K) = (1-\alpha)E(\kappa|K)$, the variance should be $E(\kappa|K)$. If we use both clues of the safety, since $0 < \alpha = \frac{1}{1+\frac{Var(\kappa)}{E(\kappa)}} < 1$, variance $Var(\kappa|K)$ never exceed $E(\kappa|K)$ and is always smaller.

The same result can be produced by using the Bayes Theorem with single observation.

The Bayes Estimator $\hat{\kappa} = E(\kappa|K) = \frac{\int \kappa\pi(K|\kappa)g(\kappa)d\kappa}{\int \pi(K|\kappa)g(\kappa)d\kappa} = \frac{\int \kappa\frac{a^b}{\Gamma(b)}\kappa^{b-1}e^{-a\kappa}*\frac{\kappa^K e^{-\kappa}}{K!}d\kappa}{\int \frac{a^b}{\Gamma(b)}\kappa^{b-1}e^{-a\kappa}*\frac{\kappa^K e^{-\kappa}}{K!}d\kappa} = \frac{\Gamma(K+b+1)}{\Gamma(K+b)}\frac{1}{1+a} = \frac{K+b}{1+a}$, then $E(\kappa|K) = \frac{K+b}{1+a} = \alpha E(\kappa) + (1-\alpha)K$.

Example [2]: There are 2 accident counts in a 5 year period in a certain location. The multivariate method suggests that $\hat{E}(\kappa) = 0.0239$/year, $\hat{Var}(\kappa) = 0.0011$/year. What's the estimate of κ ?

For a 5 year period, $\widehat{E}(\kappa) = 0.0239 * 5 = 0.1195,\ \widehat{Var}(\kappa) = 0.0011 * 5 = 0.0275$

$\hat{\alpha} = \frac{1}{1+\frac{\widehat{Var}(\kappa)}{\widehat{E}(\kappa)}} = \frac{1}{1+\frac{0.0275}{0.1195}} = 0.81,$

$\hat{\kappa} = \widehat{E}(\kappa|K) = \hat{\alpha}\widehat{E}(\kappa) + (1-\hat{\alpha})K = 0.81 * 0.1195 + 0.19 * 2 = 0.48,$

$\widehat{Var}(\kappa|K) = (1-\hat{\alpha})\widehat{E}(\kappa|K) = 0.19 * 0.48 = 0.09.$

Notice that had only the accident count been used, the estimate of κ would have been 2 accidents in 5 years and the standard deviation of that estimate would be estimated at $\sqrt{2} = 1.4$ accidents in 5 years. The EB approach makes use of both of these clues to produce a more accurate, location-specific safety estimate.

Now the focus will on be how to estimate $E(\kappa)$ and $Var(\kappa)$. By Adam's formula in probability theory, $E(K) = E(E(K|\kappa))$. By definition $E(K|\kappa) = \kappa$, therefore

$$E(K) = E(\kappa) \tag{2.15}$$

By Eve's formula in probability theory, $Var(K) = E\big(Var(K|\kappa)\big) + Var(E(K|\kappa))$. Since observation of the crash counts given a its expectation is poison distributed, $Var(K|\kappa) = E(K|\kappa) = \kappa$, therefore

$$Var(K) = E(\kappa) + Var(\kappa) \tag{2.16}$$

We illustrate this with an example:

Consider a reference population of two lane highway in rural area with speed limits70 m/h. To do an accurate calculation, we need the sample size to be large.

Let $\overline{K}$ be the average accident count among the reference population. $\overline{K}$ is the sample mean and $\overline{K} = \sum K * n(K)/n$ where $n(K)$ is the number of locations that records K accidents in this year. n is the total number of accident among the reference group. The sample variance S^2 is defined as $S^2 = \sum(K-\overline{K})^2 * n(K)/n$. As n becomes large, $\overline{K} \longrightarrow E(K)$ and $S^2 \longrightarrow Var(K)$. Therefore, if we know the occurrence of the accident in a certain year we are able to do the estimation for $E(\kappa)$ and $Var(\kappa)$. Below is a table shows the observation and the calculation.

Table 2.3: Information to estimate $E(\kappa)$ and $Var(\kappa)$

	K	n(K)	$K * n(K)$	$(K-\overline{K})^2 * n(K)$
	0	5930	0	3.416
	1	120	120	114.309
	2	4	8	15.618
	3	5	15	44.283
Total	——	6059	143	177.626

$\overline{K} = \frac{143}{6059} = 0.024 = \hat{E}(K)$,

$S^2 = \frac{177.626}{6059} = 0.029 = \hat{Var}(K)$.

Therefore by using relations 2.15 and 2.16, $\hat{E}(\kappa) = \hat{E}(K) = 0.024$ and $\hat{Var}(\kappa) = \hat{Var}(K) - \hat{E}(\kappa) = 0.029 - 0.024 = 0.005$. Therefore $\hat{\alpha} = \frac{1}{1+\frac{\hat{Var}(\kappa)}{\hat{E}(\kappa)}} = \frac{1}{1+\frac{0.005}{0.024}} = 0.8276$. Now the estimated conditional mean can be calculated. For instance, for all the locations in the reference group which recorded $K = 2$ in a certain year, then we would estimate $\hat{E}(\kappa|K) = \hat{\alpha}\hat{E}(\kappa) + (1 - \hat{\alpha})K = 0.8276 * 0.024 + (1 - 0.8276) * 2 = 0.3647$ in that year. We should also pay attention to the variance of κ. It is estimated to be 0.005, which is relatively small when compare to $\hat{E}(\kappa)$. This is partly because that the size of the reference group is large. In some cases, the reference group is not large enough to make an accurate estimation by the method of the sample moments. To address this problem the multivariate regression method is introduced here [3].

Let $X_1, X_2, \dots, X_n$ be the independent variables of the reference sites, such as AADT, road section length, or number of lanes, which are believed to be the most important factors for the occurrence of accidents. Assume that K mostly depends on these independent variables, and their relationships with K is exponential.

$K = \beta_0 + {X_1}^{\beta_1} + {X_2}^{\beta_2} + \cdots + {X_n}^{\beta_n} + \varepsilon$ where $\beta_0, \beta_1, \dots, \beta_n$ are parameters of the independent variables, and $E(\varepsilon) = 0$

Therefore,

$$E(K) = \beta_0 + {X_1}^{\beta_1} + {X_2}^{\beta_2} + \cdots + {X_n}^{\beta_n} \text{ [1]} \qquad (2.17)$$

$Var(K)$ can be estimated by the maximum likelihood estimate

The multivariate method is better than the method of sample moments in the following two aspects. First, a large number of reference sites are not needed for any particular combination of characteristics. And second, it provides estimates of $Var(\kappa)$ as well as $E(\kappa)$ for the reference sites.

In summary, had we got the historical information from the treatment site and the reference site in the before period, we could do the EB estimate for value $\hat{\kappa} = \hat{E}(\kappa|K)$. Note that this $\hat{\kappa}$ is also for the before period. The next thing is to estimate what would happen if the treatment site remains untreated in the after period. Let κ_b be the expected accident count in the before year, usually a year before implementation take place. Also, let κ_a be the expected accident count in the after year without treatment. Since the observation in the after period with no treatment is no longer available, then there is no way to distinguish κ_a with κ_b, except the performance of their reference group during different periods. Then it is reasonable to introduce the formula here:

$$\hat{\kappa}_a = \frac{\hat{E}(\kappa_a)}{\hat{E}(\kappa_b)} * \hat{\kappa}_b \text{ [2]} \qquad (2.18)$$

The loop is now closed. Had we got the estimate of κ_a, the expected accident count in the after year without treatment for the treatment site, and the observation of accident frequency in the after year, we would be able to conduct a before-and-after study using the Four-Step to measure the effectiveness of the treatment.

The EB before-and-after study method cures the RTM problem. The reasons are as following. First, it is said that the conceptual frame of the EB method fits the reality of observational study[2]. We estimate κ by calculating $E(\kappa|K)$, the mean of the $\kappa's$ in the subpopulation. When using EB method, there is a two-stage selection process. The first stage is to identify the group of locations with similar traits and remain untreated throughout the study period- the reference population. Next from the reference population, select the subpopulation with K accident counts. This way of estimation is based on the belief that locations that record K accident counts have a mean that different from the locations with L accidents. $K \neq L$, then $E(\kappa|K) \neq E(\kappa|L)$. Second, since $E(\kappa|K) = \alpha E(\kappa) + (1-\alpha)K$ and $0 < \alpha = \frac{1}{1+\frac{Var(\kappa)}{E(\kappa)}} \leq 1$, then $E(\kappa|K)$ is always between $E(\kappa)$ and K. Thus at least qualitatively, $E(\kappa|K)$ does what the logic of RTM predicts. Namely it shifts the estimate of accident count in the direction of the population mean. Hauer used a real world problem to show that the reasons are logically sound. The example appears to be incomplete in the book. Below is a more complete explanation.

Table 2.4 is based on the accident report of 1139 intersections in San Francisco in year1974 and 1975 [2]. The total accident count in year 1974 is 1211, the same as in year 1975.

Table 2.4: [2] Juxtaposition of EB estimates for 1974 and observed average in 1975

n(K) of 1974	K of 1974	$\hat{\kappa}$ of 1974 $\triangleq$ $\hat{\kappa}$ of 1975	Avg(K) of 1975
553	0	0.48	0.54
296	1	1.03	0.97
144	2	1.57	1.53
65	3	2.11	1.97
31	4	2.65	2.10
21	5	3.19	3.24
9	6	3.73	5.67
13	7	4.27	4.69
5	8	4.81	3.80
2	9	5.35	6.50

The first column in Table 2.4 is the number of the reference locations that recorded K accidents in 1974. The last column is the average accident counts in year 1975 of these n(K) locations. The third column in the table lists the estimate of κ of year 1974 using EB method. To get the value in column 3, we need to do an EB estimate, as shown in Table 2.5.

Table 2.5: Juxtaposition of EB estimates for 1974

n(K) (1974)	K(1974)	K*n(K)	$(K-\overline{K})^2 * n(K)$
553	0	0	621.35
296	1	296	1.07
144	2	288	127.24
65	3	195	244.63
31	4	124	267.95
21	5	105	326.00
9	6	54	219.63
13	7	91	458.69
5	8	40	240.82
2	9	18	126.09
Total		1211	2633.64

$\widehat{E}(\kappa) = \widehat{E}(K) = \overline{K} = \frac{1211}{1139} = 1.06,$

$\hat{V}ar(K) = S^2 = \frac{2633.64}{1139} = 2.31,$

$\hat{V}ar(\kappa) = \hat{V}ar(K) - \widehat{E}(K) = S^2 - \overline{K} = 2.31 - 1.06 = 1.25,$

$\hat{\alpha} = \frac{1}{1+\frac{\hat{V}ar(\kappa)}{\widehat{E}(\kappa)}} = \frac{1}{1+\frac{1.25}{1.06}} = 0.46.$

Therefore, if K=4, then $\hat{\kappa} = \widehat{E}(\kappa|K) = \hat{\alpha}\widehat{E}(\kappa) + (1-\hat{\alpha})K = 0.46 * 1.06 + 0.54 * 4 = 2.65$. The other values can be calculated similarly.

The next step is to predict what would happen had these 1124 locations remained untreated. The formula (2.18) $\hat{\kappa}_a = \frac{\widehat{E}(\kappa_a)}{\widehat{E}(\kappa_b)} * \hat{\kappa}_b$ is used. In the formula, "a" refers to year 1974, "b" refers to year 1975. In this case, $\frac{\widehat{E}(\kappa_{1975})}{\widehat{E}(\kappa_{1974})} = \frac{\frac{1211}{1139}}{\frac{1211}{1139}} = 1$, therefore $\hat{\kappa}_{1975} = \hat{\kappa}_{1974}$.

The simple before-and-after study method would use only the first year accident count to predict the happen of accidents in the after period, while EB method uses both the accident history and the crash information from the reference population. It is necessary to compare these two types of estimates. Namely, column 2 compares to column 3 in Table 2.4. The idea is, check which estimate is closer to the average accidents frequency in 1975. The results are shown in Figure 2.2.

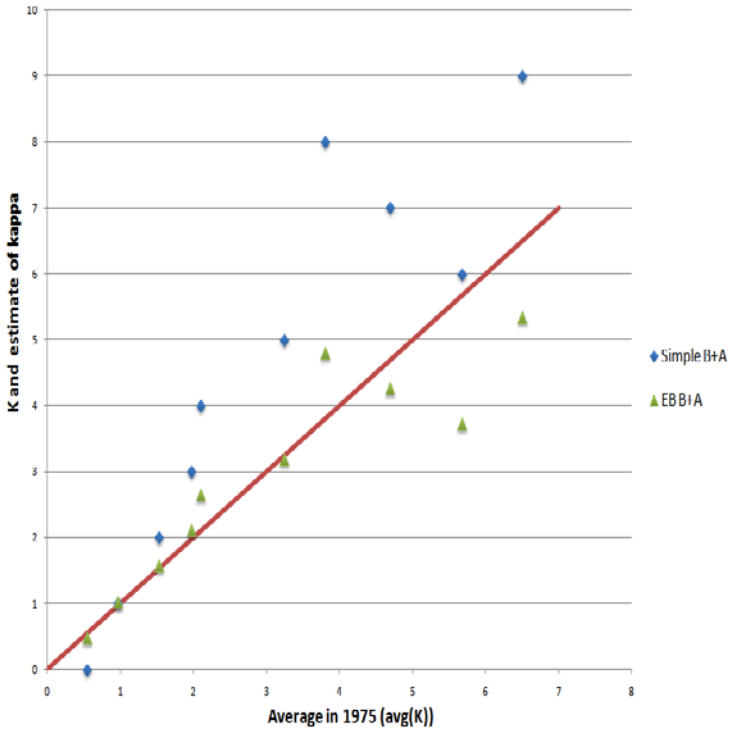

Figure 2.2: Comparing EB B+A Estimate with Simple B+A Estimate

As shown above, the ordinate of each diamond is the value of simple B+A estimate of what would happen in year 1975, the ordinate of triangle is the EB B+A estimate of what would happen in year 1975. Both these two estimates are plotted against avg(K) for 1975. The line is a standard line where K and estimate of kappa=Average of K. Except one point where K=6, all the other points of EB B+A estimates are shown to be closer to the standard line than the simple B+A estimates. That means in this example, EB method tends to be more accurate than simple before-and –after method.

2.4 Comparisons of the three methods

So far we have discussed all the three types of before and after study methods. They are all designed to estimate the effect of safety treatment. Putting aside their similarities, each method has its own way of managing data and has its advantages and downsides. We summarize the comparison among the three methods in Table 2.9 below.

Table 2.6: Comparisons of the three methods

Existing method	The Simple B+A method	The B+A with Comparison Group method	The B+A with Empirical Bayes method
Data collection	Crash history of only the treatment site	History of both the treatment site and the comparison group	History of both the treatment site and the reference group
Advantages	Only need the history data of the improvement location, calculation is straightforward	Eliminated the maturation.	Eliminated the RTM.
Shortcomings	Ignored the existence of regression to the mean, crash migration and maturation.	Usually requires a relatively large size of comparison groups. Still neglects RTM.	The calculation is relatively complicated. Still neglects maturation.

2.5 A more coherent method

This approach introduced by Hauer in [2] serves as an extension of the basic empirical before and after study method. It provides a possible way to estimate and predict the accident count all in one setting.

Let $K_{i,1} \dots K_{i,Y,}K_{i,Y+1} \dots K_{i,Y+Z}$ be the observed accident count in the ith location from year 1 through year Y+Z, in which year 1 to year Y are the Y years before treatment, and year Y+1 to year Y+Z are the Z years after treatment. Let $\kappa_{i,1} \dots \kappa_{i,Y}, \kappa_{i,Y+1} \dots \kappa_{i,Y+Z}$ be the expected accident count corresponding to the observations. The task is to estimate $\kappa_{i,1} \dots \kappa_{i,Y}$ and to predict $\kappa_{i,Y+1} \dots \kappa_{i,Y+Z}$ if the site remains untreated.

Model selection

Hauer emphases that the selection of the model is more influential in determining the quality of the product than the methodology used to estimate the parameter values. The choice of the model should reflect the prior knowledge of the relationship between accident count and the factors that potentially influence traffic safety. In a road section study, Hauer suggests to use model [2]:

$$\kappa_{i,y} = d_i \alpha_y {F_{i,y}}^{\beta} + \varepsilon_{i,y}$$

where:

d_i is the ith road section length;
$F_{i,y}$ is the annual average daily traffic(AADT);
α_y and β are parameters of the model;
$\varepsilon_{i,y}$ is the error of the model, where $E(\varepsilon_{i,y}) = 0, Var(\varepsilon_{i,y}) = \sigma^2$.

This model is based on the belief that the occurrence of accidents mainly depends on the traffic flow and the road section length. The use of α_y's in the model reflects that other than road section length and traffic flow, all the other factors that influence the road safety change from year to year, and these changes of each year affect the safety between locations in the same manner. The parameter β determines how the change of traffic flow (AADT) could affect the incidence of the accidents. This model also indicates the fact that when $d_i = 0$ or $F_{i,y} = 0$, $\kappa_{i,y} = 0$.

By the model,

$$E(\kappa_{i,y}) = d_i \alpha_y {F_{i,y}}^{\beta} \text{ [2]} \quad (2.19)$$

Likelihood function for parameter estimation

One of the most widely used methods of statistical estimation is the Maximum likelihood Estimation (MLE) method. We introduce it in this report in order to get a sensible estimate for the parameters α_y's, β, and b which will be introduced later in this section. All these parameters need to be estimated by MLE. Later they will be used to learn what would have been if there were no treatment in the treatment site. To do a MLE estimate we need the accident records of the reference group from the beginning of the before period though the after period.

Assume the occurrence of accidents at a certain entity and year is Poisson distributed. Then [2]:

$$P(K_{i,y}|\kappa_{i,y}) = {\kappa_{i,y}}^{K_{i,y}} e^{-\kappa_{i,y}} / K_{i,y}! \text{ [2]} \quad (2.20)$$

Then for R reference locations and Y+Z years,

$P(\text{accident count}\{K_{i,Y}\}|\text{parameters}\{\kappa_{i,Y}\})$ $\prod_{i=1}^{R} \prod_{y=1}^{Y+Z} {\kappa_{i,y}}^{K_{i,y}} e^{\kappa_{i,y}} / K_{i,y}!$ [2], since ${K_{i,y}}'s$ are independent.

There are R*(Y+Z) unknowns in the formula. The next task would be to replace many unknowns by the parameters. Let

$$\frac{E(\kappa_{i,y})}{E(\kappa_{i,1})} = C_{i,y} \text{ and } \frac{\kappa_{i,y}}{\kappa_{i,1}} \triangleq C_{i,y} \text{ [2]} \quad (2.21)$$

By doing this, the many ${\kappa_{i,y}}'s$ can be expressed as a function of $\kappa_{i,1}$ and $C_{i,y}$'s. Most of the time, the ${\kappa_{i,y}}'s$ in different years are not equal. The reason is as follows: in this equation, people assume that over the years the ${\kappa_{i,y}}'s$ will remain similar in some aspects, but there will also be some change from year to year. And this change should not be totally unpredictable. The author assumed that this change have something to do with the traffic flow (AADT) that it can be

captured by the model as well. Also $\kappa_{i,y}$ will be different from $E(\kappa_{i,y})$. For a certain year y, the expected accident counts of the reference sites are similar for they share similar traits, but will still be different because other factors that are not been captured could also influence the incidence of accident.

Now for a certain location i, the many unknowns $\kappa'_{i,y}s$ can be replaced by the combination of $\kappa_{i,1}$ and $C_{i,y}$'s,
then$P(K_{i1}, \dots, K_{i,Y,}K_{i,Y+1}, \dots, K_{i,Y+Z}|\kappa_{i,1}, \dots, \kappa_{i,Y}, \kappa_{i,Y+1}, \dots, \kappa_{i,Y+Z} =$
$P(K_{i1}, \dots, K_{i,Y,}K_{i,Y+1}, \dots, K_{i,Y+Z}|\kappa_{i,1}, \dots C_{i,Y}\kappa_{i,1,}C_{i,Y+1}\kappa_{i,1,} \dots C_{i,Y+Z}\kappa_{i,1}) =$

$$\left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y!}}\right)({\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}}e^{-\sum_{y=1}^{Y+Z}C_{i,y}\kappa_{i,1}})\ [2] \tag{2.22}$$

Now the likelihood function becomes:

$$P(\text{accident count}\{K_{i,Y}\}|\text{parameters}\{\kappa_{i,Y}\}) = \left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y!}}\right)({\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}}e^{-\sum_{y=1}^{Y+Z}C_{i,y}\kappa_{i,1}})$$
$$= P(K_{i1}, \dots, K_{i,Y,}K_{i,Y+1}, \dots, K_{i,Y+Z}|\kappa_{i,1}) = \prod_{i=1}^{R}\left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y!}}\right)({\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}}e^{-\sum_{y=1}^{Y+Z}C_{i,y}\kappa_{i,1}})\ [2].$$

By using the $C_{i,Y}'s$, the dimension of unknowns has been reduced a lot. However, the expected accident counts for different sites of year 1 $(\kappa_{1,1}, \kappa_{2,1}, \dots, \kappa_{R,1})$ still remain unknown. To solve this problem, assume that $\kappa_{1,1}, \kappa_{2,1}, \dots, \kappa_{R,1}$ are Gamma distributed with parameters a_i and b. Note that the reference locations have similar traits as location i, but their expected number of accident count are not necessary the same. This is because the reference locations were selected for they have similar traits, but the number of the traits we used to identify the reference location is limited. There still exist distinct characteristics among the reference locations, and this difference should not be unpredictable. Therefore we assume this change subject to the Gamma distribution

$$f(\kappa_{i,1}) = \frac{{a_i}^b}{\Gamma(b)}{\kappa_{i,1}}^{b-1}e^{-a_i\kappa_{i,1}} \text{ where } a_i = \frac{E(\kappa_{i,1})}{Var(\kappa_{i,1})}, b = \frac{E^2(\kappa_{i,1})}{Var(\kappa_{i,1})}, \text{ i=1,…,R} \tag{2.23}$$

$$\text{Therefore, } {\kappa_{i,1}}^{b-1}e^{-a_1\kappa_{i,1}} = f(\kappa_{i,1})\frac{\Gamma(b)}{{a_i}^b} \tag{2.24}$$

Also, following the same fashion,

$${\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}+b-1}e^{-(a_1+\sum_{y=1}^{Y+Z}C_{i,y})\kappa_{i,1}} = f(\kappa_{i,1})\frac{\Gamma(\sum_{y=1}^{Y+Z}K_{i,y}+b-1)}{(a_i+\sum_{y=1}^{Y+Z}C_{i,y})^{\sum_{y=1}^{Y+Z}K_{i,y}+b}} \tag{2.25}$$

Divide (2.25) by (2.24), then ${\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}}e^{-(\sum_{y=1}^{Y+Z}C_{i,y})*\kappa_{i,1}}$

Plugging the result into function (2.22), we get

$$P_T(K_{i1},\dots,K_{i,Y},K_{i,Y+1},\dots,K_{i,Y+Z}|\kappa_{i,1}) = \prod_{i=1}^{R}\left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y}!}\right)\left({\kappa_{i,1}}^{\sum_{y=1}^{Y+Z}K_{i,y}}e^{-\kappa_{i,1}\sum_{y=1}^{Y=Y+Z}C_{i,y}}\right)$$

$$= \prod_{i=1}^{R}\left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y}!}\right)\frac{(\frac{b}{E(\kappa_{i,1})})^{b}}{(\frac{b}{E(\kappa_{i,1})}+\sum_{y=1}^{Y+Z}C_{i,y})^{\sum_{y=1}^{Y+Z}K_{i,y}+b}}\frac{(\sum_{y=1}^{Y+Z}K_{i,y}+b-1)!}{(b-1)!}$$

The likelihood ℓ is given by the joint probability distribution evaluated at observed accident count $K_{i,y}$, hence

$$\ell = \prod_{i=1}^{R}\left(\prod_{y=1}^{Y+Z}\frac{{C_{i,y}}^{K_{i,y}}}{K_{i,y}!}\right)\frac{(\frac{b}{E(\kappa_{i,1})})^{b}}{(\frac{b}{E(\kappa_{i,1})}+\sum_{y=1}^{Y+Z}C_{i,y})^{\sum_{y=1}^{Y+Z}K_{i,y}+b}}\frac{(\sum_{y=1}^{Y+Z}K_{i,y}+b-1)!}{(b-1)!} \quad (2.26)$$

So far we have introduced the parameter b into the likelihood function.

Then

$$\ln(\ell) = \sum_{i=1}^{R}([\sum_{y=1}^{Y+Z}K_{i,y}\ln(C_{i,y})] + b*\ln(\frac{b}{E(\kappa_{i,1})})$$

$$-\left(\sum_{y=1}^{Y+Z}K_{i,y}+b\right)\ln\left(\frac{b}{E(\kappa_{i,1})}+\sum_{y=1}^{Y+Z}C_{i,y}\right)+\ln\frac{(\sum_{y=1}^{Y+Z}K_{i,y}+b-1)!}{(b-1)!} \quad (2.27)$$

Once the parameter values $\alpha_y's, \beta$ and b are chosen, the values of $E(\kappa_{i,y})'s$ can be calculated. Then by using equation (2.21), the parameter $C_{i,y}$ for each location (i=1,…,R) and year (y=1,…,Y+Z) can be calculated. A sensible estimate of those parameters would be the parameters that maximize the log-likelihood function. The values of the parameters that maximize the log-likelihood cannot be expressed in a nice closed form solution. (Normally, the method to solve for the values of the parameter is taking the partial derivative.) Instead they must be determined numerically by starting with a set of initial values and iterating to the maximum of the log-likelihood function. Technically, this procedure is called an iteratively re-weighted least squares method [12]. However, for this case, the form is complicated and it may be difficult to solve. It turns out that the Excel software provides a "Solver Function" that will do the job. More information about how to use solver function will be given in Section 3. From this step, we will get the estimate of $\alpha_y's, \beta$ and b.

Estimate $k_{i,1}k_{i,2}\dots k_{i,Y}$ for a certain entity

Let "i" be a treated entity. Suppose the treatment was taken at the end of year Y. The $K_{i1},\dots,K_{i,Y}$ for this treated site are available. Also to do the estimate people will need to know the value of independent variables of the model (road section length and traffic flows for Y year).

There are two kinds of estimation, namely Maximum Likelihood estimation and Empirical Bayes estimation.

(a) Maximum Likelihood Estimation [2]

From year 1 to year Y before treatment happens, the likelihood function can be written as:

$\ell(\kappa_{i,1}) = P_T(K_{i1}, \dots, K_{i,Y,}|\kappa_{i,1}) = \left(\prod_{y=1}^{Y} \frac{C_{i,y}{}^{K_{i,y}}}{K_{i,y!}}\right)\left(\kappa_{i,1}{}^{\sum_{y=1}^{Y} K_{i,y}} e^{-\kappa_{i,1}\sum_{y=1}^{Y} C_{i,y}}\right)$, this is the likelihood function for $\kappa_{i,y}$ and we wish to find the value of $\kappa_{i,1}$ that maximized the likelihood function. First take log on both sides,$\ln \ell(\kappa_{i,1}) = \ln\left[\left(\prod_{y=1}^{Y} \frac{C_{i,y}{}^{K_{i,y}}}{K_{i,y!}}\right)\right] + \sum_{y=1}^{Y} K_{i,y}\ln(\kappa_{i,1})$- $\kappa_{i,1}\sum_{y=1}^{Y} C_{i,y}$, then take derivative on both sides:

$$\frac{d[\ln \ell(\kappa_{i,1})\,]}{d\kappa_{i,1}} = \frac{\sum_{y=1}^{Y} K_{i,y}}{\kappa_{i,1}} - \sum_{y=1}^{Y} C_{i,y}$$

If $\hat{\kappa}_{i,1}$ is that value of $\kappa_{i,1}$ at which the derivative equals 0, then

$$\hat{\kappa}_{i,1} = \frac{\sum_{y=1}^{Y} K_{i,y}}{\sum_{y=1}^{Y} \hat{C}_{i,y}} \tag{2.28}$$

For the remaining $\hat{\kappa}_{i,y}$ where $y \neq 1$,

$$\hat{\kappa}_{i,y} = \hat{\kappa}_{i,1}\hat{C}_{i,y}. \tag{2.29}$$

(b) Empirical Bayes Estimation [2]

In the previous section we have demonstrated the existence of RTM and its influence on the estimation. The Maximum likelihood estimation for $\kappa_{i,y}$'s is straight forward but still subject to RTM. The EB estimation is a remedy to that. With EB approach, the estimation of $\kappa_{i,y}$ is based on the joint use of two clues: those contained in accident counts of the treated entity, and those contained in traits of this entity. $\kappa_{i,1}$ is the expected accident count for treated entity i in year 1, which had been treated at the end of year Y. Now introduce its reference population. We have discussed in the first step, that the reference locations have similar traits as location i, but their expected number of accident count are not necessary the same. And we assume this change subject to the Gamma distribution, then $f(\kappa_{i,1}) = \frac{a_i\kappa_{i,1}{}^{b-1}e^{a_i\kappa_{i,1}}}{\Gamma(b)}$ and $a_i = \frac{E(\kappa_{i,1})}{Var(\kappa_{i,1})}$, $b = \frac{[E(\kappa_{i,1})]^2}{Var(\kappa_{i,1})}$.

Under this condition, consider what is the probability density function of $f(\kappa_{i,1}, \dots, \kappa_{i,Y}|K_{i1}, \dots, K_{i,Y})$.

$$f(\kappa_{i,1}, \dots, \kappa_{i,Y}|K_{i1}, \dots, K_{i,Y}) = f(\kappa_{i,1}, C_{i,2}\kappa_{i,1}, \dots C_{i,Y}\kappa_{i,1}|K_{i1}, \dots, K_{i,Y}) = f_T(\kappa_{i,1}|K_{i1}, \dots, K_{i,Y})$$
$$= \frac{f_T(\kappa_{i,1}, K_{i1}, \dots, K_{i,Y})}{\int f_T(\kappa_{i,1}, K_{i1}, \dots, K_{i,Y})d\kappa_{i,1}} = \frac{f_T(K_{i1}, \dots, K_{i,Y}|\kappa_{i,1})f_T(\kappa_{i,1})}{\int f_T(K_{i1}, \dots, K_{i,Y}|\kappa_{i,1})\, f_T(\kappa_{i,1})d\kappa_{i,1}}$$

Let $\int f_T(K_{i1}, \dots, K_{i,Y}|\kappa_{i,1})\, f_T(\kappa_{i,1})d\kappa_{i,1} = \frac{1}{m_1}$ is a constant, then

$$f(\kappa_{i,1}, \dots, \kappa_{i,Y}|K_{i1}, \dots, K_{i,Y}) = m_1 * f_T(K_{i1}, \dots, K_{i,Y}|\kappa_{i,1}) f_T(\kappa_{i,1}) =$$
$$m_1 \left(\prod_{y=1}^{Y} \frac{C_{i,y}{}^{K_{i,y}}}{K_{i,y}!}\right)\left(\kappa_{i,1}{}^{\sum_{y=1}^{Y} K_{i,y}} e^{-\kappa_{i,1}\sum_{y=1}^{Y} C_{i,y}}\right)\left(\frac{a_i \kappa_{i.1}{}^{b-1} e^{a_i \kappa_{i,1}}}{\Gamma(b)}\right) =$$
$$m_2 \kappa_{i,1}{}^{b+\sum_{y=1}^{Y} K_{i,y}-1} e^{-\kappa_{i,1}(a_{i+}\sum_{y=1}^{Y} C_{i,y})} \text{ where } m_2 = \frac{(a_{i+}\sum_{y=1}^{Y} C_{i,y})^{b+\sum_{y=1}^{Y} K_{i,y}}}{\Gamma(b+\sum_{y=1}^{Y} K_{i,y})}.$$

At this moment we can see the conditional probability is also gamma distributed with mean $E(\kappa_{i,1}|K_{i1}, \dots, K_{i,Y}) = \frac{b+\sum_{y=1}^{Y} K_{i,y}}{a_{i+}\sum_{y=1}^{Y} C_{i,y}}$ and variance $Var(\kappa_{i,1}|K_{i1}, \dots, K_{i,Y}) = \frac{b+\sum_{y=1}^{Y} K_{i,y}}{(a_{i+}\sum_{y=1}^{Y} C_{i,y})^2}$

By the Bayes Theorem, the Bayes estimator

$$\hat{\kappa}_{i,1} = \hat{E}(\kappa_{i,1}|K_{i1}, \dots, K_{i,Y}) = \frac{\hat{b}+\sum_{y=1}^{Y} K_{i,y}}{\hat{a}_{i+}\sum_{y=1}^{Y} \hat{C}_{i,y}} = \frac{\hat{b}+\sum_{y=1}^{Y} K_{i,y}}{\frac{\hat{b}}{\hat{E}(\kappa_{i,1})}+\sum_{y=1}^{Y} \hat{C}_{i,y}} \quad (2.30)$$

$$\hat{Var}(\kappa_{i,1}) = \hat{Var}(\kappa_{i,1}|K_{i1}, \dots, K_{i,Y}) = \frac{\hat{b} + \sum_{y=1}^{Y} K_{i,y}}{(\hat{a}_{i+}\sum_{y=1}^{Y} \hat{C}_{i,y})^2} \quad (2.31)$$

For the remaining $\hat{\kappa}_{i,y}$ where $y \neq 1$, $\hat{\kappa}_{i,y} = \hat{\kappa}_{i,1}\hat{C}_{i,y}$.

(c) The relationship between Maximum Likelihood estimation and the EB estimation

When the distribution of $\kappa_{i,1}'s$ is such that the standard deviation $\sigma(\kappa_{i,1})$ is large compared with $E(\kappa_{i,1})$, then $b = [\frac{E(\kappa_{i,1})}{\sigma(\kappa_{i,1})}]^2$ is small and $a_i = \frac{b}{E(\kappa_{i,1})}$ is also a small number, then the EB estimate will coverage toward the maximum likelihood estimate. The Maximum likelihood estimator converges to the average accident counts (accident count divided by the number of years) when $\kappa_{i,1}'s$ does not change from year to year [2] ($C_{i,y} = 1$ for all i=1,…,Y). The advantage of EB method is that it provides not only the estimator, but also the estimate of the variance.

Estimate $k_{i,Y+1}k_{i,Y+2} \dots k_{i,Y+Z}$ for a certain entity

We have stated that the entity i had been treated at the end of year Y. Then the observation $K_{i,Y+1}K_{i,Y+2} \dots K_{i,Y+Z}$ could no longer represent the historical data under untreated situation. Our task is to predict what would have been the expected accident frequencies $\kappa_{i,Y+1}\kappa_{i,Y+2} \dots \kappa_{i,Y+Z}$ in the after year had the treatment not been applied. From the maximum likelihood function in (2.27) we get the estimation of necessary parameters $C_{i,Y+1}, \dots, C_{i,Y+Z}$, and for the treated entity we have the estimate of $\kappa_{i,1}$, using either Maximum Likelihood estimation or Empirical Bayes estimation. What remains to be done is pretty straight forward by following equation (2.32):

For y>Y, $\hat{\kappa}_{i,y} = \hat{C}_{i,y}\hat{\kappa}_{i,1}$ and $\hat{Var}(\kappa_{i,y}) = \hat{C}_{i,y}{}^2\hat{\kappa}_{i,1}$ (2.32)

Overall, this method performs better and is preferred when required data are available. In our decision support system to be developed, all four methods discussed in this chapter are used with preference given to this method first. It is also used in the example we are going to discuss in the next chapter.

Chapter 3. Systematic Optimization of Traffic Safety Strategies Implementation

3.1 The optimization model

As discussed in Chapter 2, the standard procedure for identifying and eliminating hazardous locations represents a reactive approach, and the optimization model used in the Safety Analyst software is not accurate. To help bridge the gap and provide a more accurate model for proactive implementation of countermeasures, we developed the following optimization model:

Decision variables:

Y_{Lp}^{Si}: whether or not to implement countermeasure scenario i at location p (0 = no, 1 = yes).

Objective function and constraints:

$$Maximize: (N_{L1}^{S1} - N_{L1}^{0})Y_{L1}^{S1} + (N_{L1}^{S2} - N_{L1}^{0})Y_{L1}^{S2} + \cdots (N_{L1}^{Si} - N_{L1}^{0})Y_{L1}^{Si} + \cdots (N_{L1}^{SI} - N_{L1}^{0})Y_{L1}^{SI}$$

$$+(N_{L2}^{S1} - N_{L2}^{0})Y_{L2}^{S1} + (N_{L2}^{S2} - N_{L2}^{0})Y_{L2}^{S2} + \cdots (N_{L2}^{Si} - N_{L2}^{0})Y_{L2}^{Si} + \cdots (N_{L2}^{SI} - N_{L2}^{0})Y_{L2}^{SI}$$

$$+ \ldots + (N_{Lp}^{S1} - N_{Lp}^{0})Y_{Lp}^{S1} + (N_{Lp}^{S2} - N_{Lp}^{0})Y_{Lp}^{S2} + \cdots (N_{Lp}^{Si} - N_{Lp}^{0})Y_{Lp}^{Si} + \cdots (N_{Lp}^{SI} - N_{Lp}^{0})Y_{Lp}^{SI}$$

$$+ \ldots + (N_{LP}^{S1} - N_{LP}^{0})Y_{LP}^{S1} + (N_{LP}^{S2} - N_{LP}^{0})Y_{LP}^{S2} + \cdots (N_{LP}^{Si} - N_{LP}^{0})Y_{LP}^{Si} + \cdots (N_{LP}^{SI} - N_{LP}^{0})Y_{LP}^{SI}$$

S.T.

$$\left.\begin{array}{l} \sum_{i=1}^{I} Y_{L1}^{Si} \le 1 \\ \sum_{i=1}^{I} Y_{L2}^{Si} \le 1 \\ \quad . \quad . \quad . \\ \sum_{i=1}^{I} Y_{Lp}^{Si} \le 1 \\ \quad . \quad . \quad . \end{array}\right\} \text{Constraint group 1}$$

$$\sum_{i=1}^{I} Y_{LP}^{Si} \le 1$$

$$\sum_{p=1}^{P}\sum_{i=1}^{I} CI_{Lp}^{Si} Y_{Lp}^{Si} + \sum_{p=1}^{P}\sum_{i=1}^{I} CM_{Lp}^{Si} Y_{Lp}^{Si} \le B_{total}$$

Y_{Lp}^{Si}'s are binary

Where N_{Lp}^{Si} is the predicted average number of crashes for location p after the implementation of countermeasures in scenario i, CI_{Ln}^{Si} is the implementation cost of the countermeasures in scenario i at location p, CM_{Lp}^{Si} is the maintenance cost of the countermeasures in scenario i at location p,and B_{total} is the total budget.

This model maximizes the expected crash reduction number in all the locations under consideration while satisfying the budget constraints. The first group of constraints makes sure that only one scenario of countermeasures is implemented in each location to avoid counting the implementation of a countermeasure more than once. For example, S_1 represents the scenario that implements countermeasure 1, S_2 represents the scenario that implements countermeasures 1 and 2, S_3 represents the scenario that implements countermeasures 1, 2, and 3, and so on. All the possible combinations of countermeasures will be included as different scenarios. The constraint after the first group makes sure that the total implementation cost and maintenance cost of a certain year does not exceed the total budget available. The last constraint is the binary constraint that limits the variable value to 1 or 0.

3.2 Predict the number of crashes

To get the input to the optimization model presented in section 3.1, the expected accident counts for a facility/site with or without treatment in the coming year need to be estimated. We first divide the study period into two parts: the before period is from the beginning of the study till the current year, and the after period is the coming year. The study takes into consideration the two concepts discussed in Chapter 2, which are:

1. The RTM problem is the main problem that could affect the accuracy of the estimate;
2. The road section length and the traffic flow are two main factors that will affect traffic safety.

It is also important to emphasize the use of the expectation value in this probability function. We use the expected traffic accident counts for two reasons. First, the accident count in the coming year could not be observed in the current year, so it is not available to use. Second, since the RTM problem is shown to be disturbing, which will often exaggerate the effect of the treatment, we use the expected value to remove the random effect, especially for the RTM. The expected value represents the actual accident frequency from implementing the countermeasures. This will make our inputs to the previous optimization model more accurate to use. Typically, the method of forecasting the accident frequencies consists of 3 steps. The detailed discussion is listed below.

Step 1: Data collection and preparation

An adequate set of data need to be prepared before applying the algorithm. Basically, the following data will be needed for the study:

1. The accident report of the treatment site from the beginning of the study period till the current year.
2. The accident report of the first reference group G1, from the beginning of the study period till current year. G1 is the group of sites that share similar traits with the treatment site. The sites in G1 were not been treated by S_p throughout the study period.
3. The accident report of the second reference group G2, from the beginning of the study period till current year. G2 is the group of sites that share similar traits with the treatment site and all the sites had been treated by S_p at least from the beginning of the study period.

For missing data in the data set, we will interpolate to predict the data value. Interpolation is a method of constructing new data points within the range of a discrete set of known data points. A brief introduction of different types of interpolation method is given below [13]:

Linear interpolation is one of the simplest methods. Basically it tells that if a point (x_i, y_i) is missing between two known points (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}). The point to be interpolated is given by fitting (x_i, y_i) into the line that created by (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1})

$y_i = y_{i-1} + \frac{x_i - x_{i-1}}{x_{i+1} - x_{i-1}}(y_{i+1} - y_{i-1})$ at the point (x_i, y_i). It is said that linear interpolation is easy to handle, but it is not very precise for data with random effects.

Polynomial interpolation is a generalization of linear interpolation. We replace the linear function with polynomial function when predict. Polynomial extrapolation can create a smoother curve.

The polynomial interpolation subjects to great error when its degree is large. Spline interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a spline.

In practice, the Mathematica software is available to use for interpolation. It uses minimum polynomial to fit the data. For example, suppose there is a sequence of accident counts from year 1 to 5, which the third year data is missing: 2,3,*,0,5. In Mathematica, use the Interpolate statement:

```
In[1]:= f = InterpolatingPolynomial[{{1, 2}, {2, 3}, {4, 0}, {5, 5}}, x]
Out[1]= 2 + (1 + (-5/6 + 3/4 (-4 + x)) (-2 + x)) (-1 + x)

In[3]:= f[x = 3]
Out[3]= 5/6 [3]

In[33]:= Show[Plot[f, {x, 1, 5}], ListPlot[{{2, 3, 5/6, 0, 5}}]]
```

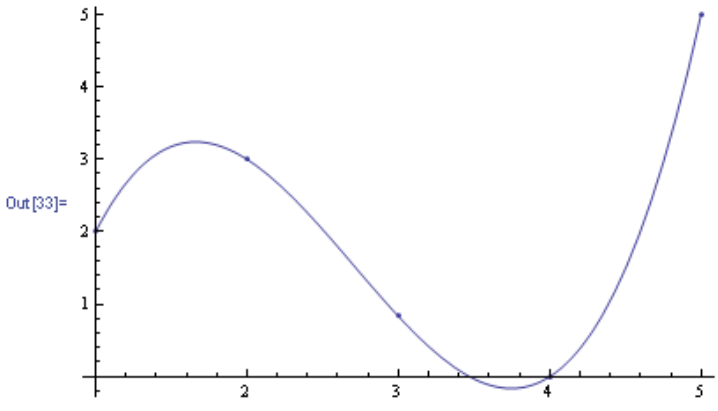

Figure 3.1: Example of Data Interpolation 1

It is shown in this example that when year is 3, the predict value is 5/6. The function it used to fit the data is also showed in the output.

In addition to interpolation for the missing values, it is required that we forecast the occurrence of the accidents of the treatment site and the two reference groups of the coming year to prepare for the next step. The extrapolation method will do the job. We choose the extrapolation method because of three reasons. First, notice that the occurrence of the accidents varies in certain pattern, and this pattern should not be unpredictable. Second, the extrapolation method offers several ways to extrapolate according to the pattern of variation. The last but not least, the extrapolation is a handy tool which is found to be useful in many transportation projects. For example, Hauer used the linear extrapolation method to simulate the missing data in the study of measure the effect of implementation in California.

Here we introduce four main types of extrapolation methods [14]. The choice of which type to use in practice depends on a prior knowledge of the data pattern.

1. Linear extrapolation means creating a tangent line using the two end points (or more than two points) and extending it beyond that limit. This is a sound choice when the data points are approximately linearly distributed. For example, suppose the two end points of the data are (x_{i-1}, y_{i-1}) and (x_i, y_i), then the point to be extrapolated is $y_{i+1} = y_{i-1} + \frac{x_{i+1}-x_{i-1}}{x_i-x_{i-1}}(y_i - y_{i-1})$.
2. Polynomial extrapolation is to generate a polynomial curve through the entire known data or just near the end. Polynomial extrapolation is typically done by means of Lagrange interpolation or using Newton's method of finite differences to create a Newton series that fits the data. The resulting polynomial may be used to extrapolate the data.
3. Conic extrapolation uses five points at the end of the data to create a conic section. If the section created is an ellipse or circle, it will loop back and rejoin itself. A parabolic or hyperbolic curve will not rejoin itself, but may curve back relative to the X-axis.
4. French curve extrapolation is suitable for any distribution with accelerating or decelerating factors.

It is suggested that we use the linear extrapolation method. Although the polynomial extrapolation can create a smooth result, it subjects to great uncertainty. The polynomial extrapolation provides a sound result only near the end point. The next example shows that polynomial is not suitable. Suppose the four year accident counts for a certain site is 5, 4, 9, 6. We want to forecast the accident frequency of the fifth year. By using Mathematica, the result is shown below:

```
In[1]:= f = Interpolation[{5, 4, 9, 6}]
Out[1]= InterpolatingFunction[{{1, 4}}, <>]

In[2]:= f[5]
InterpolatingFunction::dmval :
  Input value {5} lies outside the range of data in the interpolating function. Extrapolation will be used.
Out[2]= -19

In[3]:= Plot[f[x], {x, 1, 5}]
        Show[%, ListPlot[{5, 4, 9, 6, -19}]]
```

Out[4]=

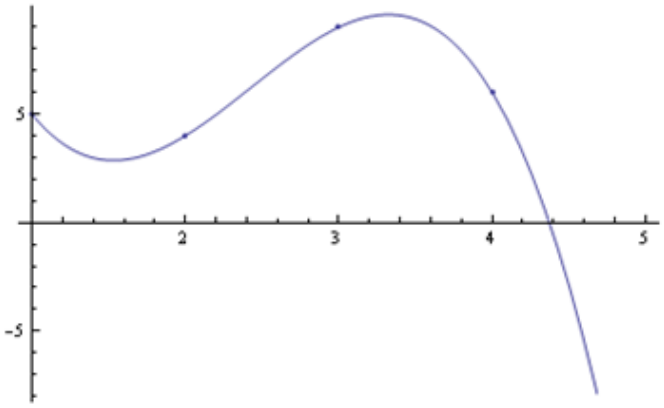

Figure 3.2: Example of Data Extrapolation 2

We can see year 5 is out of the domain. As a result, a warning appears claiming that extrapolation is used. When the point is far apart from the end point, the error became large. -19 definitely is not a nice result since we expect a result that is greater or equal to 0.

Moreover, conic extrapolation is not suitable for predicting accident count since the set of points does not have any trend to "loop back". French curve is good when the distribution has accelerating or decelerating factors. The occurrence of the accident reflects the combined effect of many sundry factors, such as road condition, traffic condition and drive's behavior. In practice the trend of accelerating or decelerating is not obvious.

Above are the reasons why the other extrapolations are not suitable and we may use linear extrapolation. There are different types of linear extrapolation, depend on how we choose to use the data. The extrapolation with two end points is easy to use, but it cannot reflect the trend of accident frequency over the years. Here we introduce linear regression model to fit the data. This model use least square estimate (LSE) to fit the data and provide a regression line through the two periods. The coming year accident frequency can be predicted by the linear regression model.

Model: $y = \beta_o + \beta_1 x + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$ (3.1)

In the model x represent the year, y represent the accident frequency. The statistical software SAS will provide a LSE estimate for the parameters β_o and β_1 along with the variance.

An example:

A report from Saint Louis county reveals that in year 2004-2008, the accident frequencies for the road section with number 0069009999 are 2, 3, 2, 4, 3. The data is missing in year 2009. We want to make a predictive analysis about how many accidents would have happened in 2009. Use simple SAS code will give the regression values.

SAS Code:

```
proc reg data=StLouis_Crash;
model Acc_Num =Year;
symbol value=circle INTERPOL=R;
proc gplot;
plot Acc_Num *Year;
Run;
```

Result is shown in Figure 3.3:

The REG Procedure
Model: MODEL1
Dependent Variable: accnum accnum

Number of Observations Read	5
Number of Observations Used	5

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	0.90000	0.90000	1.42	0.3189
Error	3	1.90000	0.63333		
Corrected Total	4	2.80000			

Root MSE	0.79582	R-Square	0.3214
Dependent Mean	2.80000	Adj R-Sq	0.0952
Coeff Var	28.42223		

Parameter Estimates						
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	1.90000	0.83467	2.28	0.1073
year	year	1	0.30000	0.25166	1.19	0.3189

Figure 3.3: SAS Results

The scatter plot with the regression line (2004-2008) is also given:

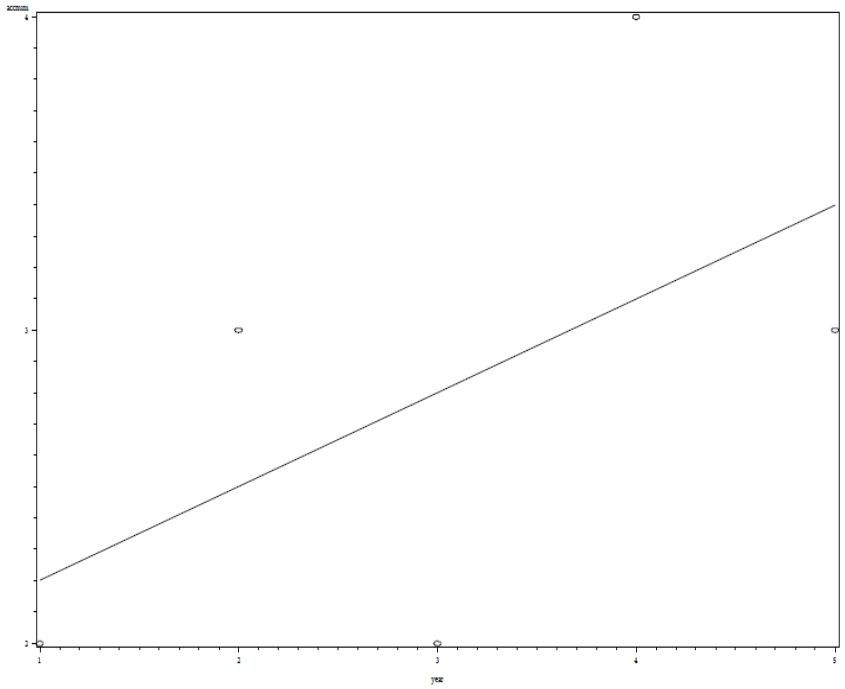

Figure 3.4: Scatter Plot and Regression Line for the Five-Year Data

We can see from the result the estimate of β_o is 1.9 and the estimate of β_1 is 0.3. Therefore for year 2009, the expected accident count is predicted as: $\beta_o + \beta_1 x = 1.9 + 0.3 * 6 = 3.7$. When necessary, the result can be round up to 4.

By using the interpolation and extrapolation method, the forecasting values are prepared and ready for use.

Step 2: Forecast the expected accident count of treatment site without treatment

We will be using Hauer's more coherent method to predict the expected accident count in the after period if no treatment has been made. The reason for choosing this method is because that it provide the estimate of the expected accident count for treatment site of each individual year. Also, the use of the Empirical Bayes estimation in Hauer's method mitigates the influence of the RTM problem. Moreover, the coherent approach takes into account the road characteristics that will relate to the number of accidents. This is achieved by selecting the adequate model $E(\kappa_{i,y}) = d_i \alpha_y F_{i,y}{}^{\beta}$. This model reflects the concern of road section length and traffic flow as the main external factors that will affect the accident frequency.

However, this method cannot be used directly to forecast the expected accident frequencies in the coming year. From the previous step, we have got the extrapolation value ready to use. Hereby we introduce the extrapolation method combining with the Hauer's more coherent method to do the forecast.

Suppose the before period of the treatment site t pertains L years and it's actually accident numbers are $K_{t,1}K_{t,2}$ ... $K_{t,L}$. Our goal is to forecast the expected accidents count for the next coming year—year L+1 under the condition that no improvement has been made. In this case, the after period is the year L+1. To start, we introduce the reference group G1. As mentioned before, G1 is a group with no treatment and has similar traits with the treatment site. Let's assume G1 consists of m sites and has L years of accident counts report. The accident counts of site i year j is denoted by $K_{i,j}$. Then for each site, the actual accident frequencies are:

$K_{1,1}, K_{1,2\ldots}K_{1,L}$

$\vdots$

$K_{m,1}, K_{m,2\ldots}K_{m,L}$

If we were given the accident frequencies of the untreated reference group G1 in the coming year, and then use the EB method to mitigate the regression to the mean phenomenon, we then able to conduct relatively accurate forecast estimation. The extrapolation will be use here. Let's denote the crash frequency of a certain site i in the coming year without treatment is $K_{i,L+1}$ for i=1,…, m then extrapolate the data we get the estimated accident frequency for the next year:

$$K_{1,1}, K_{1,2\ldots}K_{1,L} \xrightarrow{\text{extrapolation}} K_{1,L+1}$$

$$K_{m,1}, K_{m,2\ldots}K_{m,L} \xrightarrow{\text{extrapolation}} K_{m,L+1}$$

The next step is to apply the maximum likelihood estimate. The idea is that after extrapolation, we precede as the accident counts of the coming year were known. At this step, the problem becomes how to predict the expected accident counts of the treatment site in year L+1 without treatment. The model to be used is from Hauer's coherence method $E(\kappa_{i,y}) = d_i\alpha_y F_{i,y}{}^{\beta}$. Also, by equation (2.27), the log likelihood function for this problem is

$$\ln(\mathcal{L}) = \sum_{i=1}^{m}([\sum_{y=1}^{L+1} K_{i,y}\ln(C_{i,y})] + b * \ln(\frac{b}{E(\kappa_{i,1})}) - \left(\sum_{y=1}^{L+1} K_{i,y} + b\right)\ln\left(\frac{b}{E(\kappa_{i,1})} + \sum_{y=1}^{L+1} C_{i,y}\right) + \ln\frac{(\sum_{y=1}^{L+1} K_{i,y}+b-1)!}{(b-1)!}.$$

As introduced in Chapter 2, apply the solver function in Excel will give the estimate of $\alpha_1, \ldots \alpha_L, \alpha_{L+1,}\beta, b$. Below is an example about how to use Excel to estimate the parameters. The four years' data of G1 is from [2]. In our case, if the last year is the current year, then we will extrapolate to get the estimate accident number of the coming year, then proceeds as we have five year of data. Due to the availability of a practical data source, we will just use the four year data provided by Hauer. However, the way to process the data is the same no matter how we prepare the data in the previous step.

An example:

Suppose there is four year of data of six rural two lane road section, their information including road section length, accident counts and AADT are shown in table:

Table 3.1: Data for six road sections [2]

Road section	Year	AADT	Length (km)	Accident counts	Road section	Year	AADT	Length (km)	Accident counts
1	1	320	3.1	1	4	1	4100	5.6	9
1	2	400	3.1	0	4	2	4500	5.6	7
1	3	450	3.1	0	4	3	5000	5.6	6
1	4	500	3.1	0	4	4	5100	5.6	2
2	1	1220	4.2	5	5	1	7600	3.7	13
2	2	1200	4.2	4	5	2	7400	3.7	12
2	3	1300	4.2	9	5	3	7200	3.7	15
2	4	1250	4.2	6	5	4	6800	3.7	23
3	1	3300	2.7	2	6	1	9250	1.9	6
3	2	3000	2.7	6	6	2	9260	1.9	2
3	3	2700	2.7	0	6	3	8700	1.9	4
3	4	2700	2.7	3	6	4	8900	1.9	3

The model used is $E(\kappa_{i,y}) = d_i \alpha_y F_{i,y}{}^{\beta}$. We start to find the maximum likelihood value with assigning initial value to the parameters. These parameters are $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta, b$. $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta$ are from the model, b is parameter of the Gamma distribution. We start with setting $\hat{\beta} = 1$, that is, assume the traffic accident counts would be proportional to traffic flow AADT. Also, use $\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = 0.0002$. This comes from the observation of road section 4, year 3. It is known that 5.6 kilometer long record accident frequency of 5 with an AADT about 5000 and $\frac{6}{5.6*5000} \triangleq 0.0002$. Also, start with guessing $\hat{b} = 1$, this is equivalent to guessing $E^2(\kappa_{i,1}) = Var(\kappa_{i,1})$ since $b = \frac{E^2(\kappa_{i,1})}{Var(\kappa_{i,1})}$. With these initial values, we are able to calculate the likelihood value, for example in year 3 of road section 2, $\hat{E}(\kappa_{2,3}) = d_2 \hat{\alpha}_3 F_{2,3}{}^{\hat{\beta}} = 4.2 * 0.0002 * 1300^1 = 1.092$, and $\hat{C}_{2,3} = \frac{\hat{E}(\kappa_{2,3})}{\hat{E}(\kappa_{2,1})} = \frac{1.0920}{1.0248} = 1.0656$. The value of $\ln(\mathcal{L})$ turns out to be 137.48 in this case.

Table 3.2 shows the estimated values of $E(K_{i,y})$ and $C_{i,y}$.

Table 3.2: Starting values of $\hat{E}(K_{i,y})$ and $\hat{C}_{i,y}$

Road Section	Year	$\hat{E}(K_{i,y})$	$\hat{C}_{i,y}$	$\sum_{y=1}^{4}\hat{C}_{i,y}$	Road Section	Year	$\hat{E}(K_{i,y})$	$\hat{C}_{i,y}$	$\sum_{y=1}^{4}\hat{C}_{i,y}$
1	1	0.1984	1.0000		4	1	4.5920	1.0000	
1	2	0.2480	1.2500		4	2	5.0400	1.0976	
1	3	0.2790	1.4063		4	3	5.6000	1.2195	
1	4	0.3100	1.5625	5.2188	4	4	5.7120	1.2439	4.5610
2	1	1.0248	1.0000		5	1	5.6240	1.0000	
2	2	1.0080	0.9836		5	2	5.4760	0.9737	
2	3	1.0920	1.0656		5	3	5.3280	0.9474	
2	4	1.0500	1.0246	4.0738	5	4	5.0320	0.8947	3.8158
3	1	1.7820	1.0000		6	1	3.5150	1.0000	
3	2	1.6200	0.9091		6	2	3.5188	1.0011	
3	3	1.4580	0.8182		6	3	3.3060	0.9405	
3	4	1.4580	0.8182	3.5455	6	4	3.3820	0.9622	3.9038

Table 3.3 shows the value of the likelihood function. It is divided into four parts:

$\ln(\mathcal{L})$ =Part1+Part 2+Part 3+Part 4,

Part 1=$\sum_{i=1}^{6}([\sum_{y=1}^{4} K_{i,y}\ln(C_{i,y})]$,

Part 2= $b * \ln(\frac{b}{E(\kappa_{i,1})})$,

Part 3=$-\left(\sum_{y=1}^{4} K_{i,y} + b\right)\ln\left(\frac{b}{E(\kappa_{i,1})} + \sum_{y=1}^{4} C_{i,y}\right)$,

Part 4= $\ln\frac{(\sum_{y=1}^{4} K_{i,y}+b-1)!}{(b-1)!}$.

The values in Table 3.3 can be calculated by plugging the values from Tables 3.1 and 3.2 into each part.

Table 3.3: Values for the four components of the log likelihood function

Road section	Part 1	Part 2	Part3	Part 4	Row sum
1	0.0000	1.6175	-4.6563	0.0000	-3.0389
2	0.6513	-0.0245	-40.4826	54.7847	14.9289
3	-1.1739	-0.5777	-16.9512	17.5023	-1.2005
4	2.2788	-1.5243	-39.1045	54.7847	16.4348
5	-3.6892	-1.7270	-88.6203	201.0093	106.9727
6	-0.3360	-1.2570	-22.9166	27.8993	3.3896
Total Sum					137.4866

The value of the likelihood varies along with the changing of the parameter values $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta, b$. The question is when the likelihood function reaches the maximum. The solver function in excel is a tool to find the maximum. The way to apply is shown here:

1. Choose the value that needs to be maximized. Select the Solver function in the data column. A window will pump out by pressing the "solver" bottom. In this window, one can set the target cell. In this spread sheet, the target cell is P17, the corresponding value is the maximum likelihood function value.
2. Set the values that will be changed to achieve the maximum or (minimum). In this example, the values are $\hat{\alpha}_1, \hat{\alpha}_2, \hat{\alpha}_3, \hat{\alpha}_4, \hat{\beta}, \hat{b}$ which correspond to the cells L5, M5, N5, O5, P5, and Q5 respectively. The "Solver Parameters" window also provides space for you to add the constraints for the parameters.
3. After setting the objective values and the dependent values in the window, press the "Solve" bottom and Excel will iterate until reach the maximum. Figures 3.5 and 3.6 show the spreadsheets of all the corresponding values.

	A	B	C	D	E	F	G	H	I	J
3	Total # of road sections in the reference		6							
4										
5	Total # of years:		4							
6										
7										
8	Input Data for Six Road Sections					E(Kiy) and Ciy			Calculating:	
9	Road Sectior	Year	AADT	Length in KM	Accidents count	E(Kiy)	Ciy	sum(Ciy)	$K_{i,y}\ln(C_{i,y})$	$-\left(\sum_{y=1}^{4} K_{i,y} + b\right)$
10	1	1	320	3.1	1	1.05	1.00		0.00	
11	1	2	400	3.1	0	1.04	1.00		0.00	
12	1	3	450	3.1	0	1.20	1.14		0.00	-4.00
13	1	4	500	3.1	0	1.50	1.43	4.57	0.00	-27.00
14	2	1	1220	4.2	5	3.38	1.00		0.00	-14.00
15	2	2	1200	4.2	4	2.88	0.85		-0.63	-27.00
16	2	3	1300	4.2	9	3.22	0.95		-0.43	-66.00
17	2	4	1250	4.2	6	3.67	1.09	3.89	0.50	-18.00
18	3	1	3300	2.7	2	4.14	1.00		0.00	
19	3	2	3000	2.7	6	3.36	0.81		-1.26	
20	3	3	2700	2.7	0	3.32	0.80		0.00	
21	3	4	2700	2.7	3	3.88	0.94	3.55	-0.19	
22	4	1	4100	5.6	9	9.88	1.00		0.00	
23	4	2	4500	5.6	7	9.05	0.92		-0.61	
24	4	3	5000	5.6	6	10.28	1.04		0.24	
25	4	4	5100	5.6	2	12.17	1.23	4.19	0.42	
26	5	1	7600	3.7	13	9.74	1.00		0.00	
27	5	2	7400	3.7	12	8.26	0.85		-1.98	
28	5	3	7200	3.7	15	8.60	0.88		-1.86	
29	5	4	6800	3.7	23	9.68	0.99	3.73	-0.12	
30	6	1	9250	1.9	6	5.68	1.00		0.00	
31	6	2	9260	1.9	2	4.90	0.86		-0.29	
32	6	3	8700	1.9	3	4.99	0.88		-0.39	
33	6	4	8900	1.9	4	5.92	1.04	3.79	0.17	

Sheet1

Figure 3.5: Spreadsheet1—Road Section Data and Estimate of $E(K_{i,y})$ and $C_{i,y}$

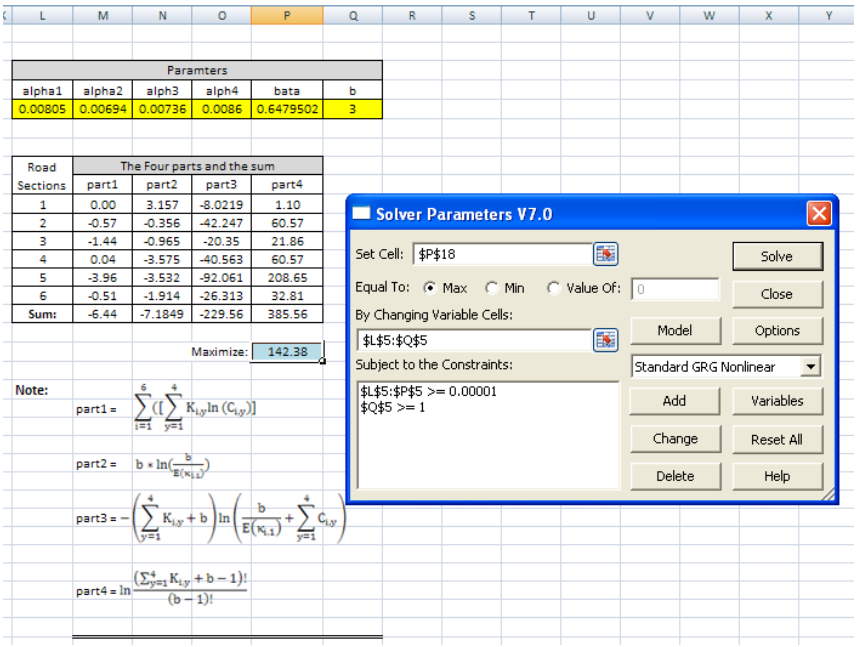

Figure 3.6: Spreadsheet 2—Output of the Maximum Likelihood Estimate

The result is $\ln(\mathcal{L})$=142.3781, when

$$\hat{\alpha}_1 = 0.0080, \hat{\alpha}_2 = 0.0069, \hat{\alpha}_3 = 0.0074, \hat{\alpha}_4 = 0.0086, \hat{\beta} = 0.6480, \hat{b} = 3$$

It is necessary to emphasize the role of the initial value. In this example, the likelihood function is a nonstandard type of function. Local maximum values might be reached with different set of initial values. It is better to have the initial values as close to the true values as possible.

With the estimated parameters, the next thing is to estimate $\kappa_{t,1}\kappa_{t,2} \ldots \kappa_{t,L}$ in the before period and predict $\kappa_{t,L+1}$ in the after period for the treatment site. From the last step, we now have obtained $\hat{\alpha}_1, \ldots, \hat{\alpha}_L, \hat{\alpha}_{L+1}, \hat{\beta}, \hat{b}$. The model $E(\kappa_{t,y}) = d_t \alpha_y {F_{t,y}}^{\beta}$ gives the estimate of $E(\kappa_{t,y})$ of year 1…L+1. By equation (2.21) $\frac{E(\kappa_{t,y})}{E(\kappa_{t,1})} = C_{t,y}$, the estimate of $C_{t,1}, \ldots, C_{t,L+1}$ can be calculated. Chapter 2 provides two types of estimation methods: Maximum Likelihood estimate and Empirical Bayes estimate. It is suggested to use the Empirical Bayes estimation method since it provides the variance of the estimation. The estimator is $\hat{\kappa}_{t,1} = \frac{\hat{b}+\sum_{y=1}^{L+1} K_{t,y}}{\hat{a}_{t+}\sum_{y=1}^{L+1} \hat{C}_{t,y}}$, where $\hat{a}_t = \frac{\hat{b}}{\hat{E}(\hat{\kappa}_{t,1})}$. This estimator estimates the expected accident number of the treatment site t of year 1. It's variance is given by equation (2.31), $\hat{V}ar(\kappa_{t,1}) = \frac{\hat{b}+\sum_{y=1}^{L+1} K_{L,y}}{(\hat{a}_t+\sum_{y=1}^{L+1} \hat{C}_{L,y})^2}$ For the coming year L+1, by the assumption $\frac{\kappa_{i,y}}{\kappa_{i,1}} \triangleq C_{i,y}$ which appears in equation (2.21), then

$$\hat{\kappa}_{t,L+1} = \hat{\kappa}_{t,1}\hat{C}_{t,L+1}, \qquad (3.2)$$

$$\widehat{\mathrm{Var}}\left(\kappa_{t,L+1}\right) = \hat{C}_{t,L+1}^{2}\widehat{\mathrm{Var}}(\hat{\kappa}_{t,1}). \quad (3.3)$$

We can also use the same algorithm to get the estimate of $\hat{\kappa}_{i,L+1}$ for each reference site where i=1,…,n. But so far at this step, what we care about is the estimate of $\kappa_{t,L+1}$.

At this moment, $\hat{\kappa}_{t,L+1}$ — the forecast value of expected accident count in the after period with no treatments is available to use.

Step 3: Forecast the expected accident count of treatment site after treatment

Different from the previous step, the accident counts of the treatment site after treatment can neither be observed nor be extrapolated. The critical point is that we do not have any information about what will happen in the post-treatment site. However, similar information can be gained from the sites with the same treatment. Therefore, it is necessary to introduce another reference group (G2). G2 is a group of sites that have similar traits with treatment site and have been treated at least from the beginning of the study period with certain improvement(s). G1 different from G2 in the sense that treatment has been applied to G2 before study begins while G1 are not been treated throughout the study period. Suppose there are n reference sites in G2, their accident counts are listed below:

$K_{1,1}^{*}, K_{1,2}^{*}, \ldots, K_{1,L}^{*}$

$K_{n,1}^{*}, K_{n,2}^{*}, \ldots, K_{n,L}^{*}$

Where '*' is a special sign to differentiate second step from first step. At this point, according to different situation, we have three recommended methods.

1. The ideal case. Suppose there exists a site that is similar enough to the treatment site in many aspects such as road type, road section length, traffic flow and other sundry factors. The only difference is that it has been treated before the study period. In this case it is reasonable to assume that the observed accident frequency between this site and the treatment site are close enough. By introducing this site as a reference site, and then extrapolate to predict accident count for next year. This predicted value would be the estimate of accident count for the treatment site of the next year.

$$K_{1,1}^{*}, K_{1,2}^{*}, \ldots, K_{1,L}^{*} \xrightarrow{\text{extrapolation}} K_{1,L+1}^{*},$$

$$\hat{\kappa}_{t,L+1}^{*} = K_{1,L+1}^{*}. \quad (3.4)$$

2. However, in most of the cases, it is hard to find such a site that similar enough to the target site. In this case, a large sample size of the second reference group would be a better choice. "Large" means sufficient enough to get rid of the random effect and other sundry effects. And also the mean of the road section length and the mean of the traffic flow of those reference sites are close to the treatment site ($E(d_i) \triangleq d_t, E\left(F_{i,y}\right) = F_{t,y}$). To estimate the expected accident count of treatment site after treatment, first extrapolate

each reference site to get the predicted accident count. And then take the average of those predicted accident counts. This average value will serve as the estimated expected accident count of the treatment group in the after period.

$$K^*_{1,1}, K^*_{1,2}, \dots, K^*_{1,L} \xrightarrow{\text{extrapolation}} K^*_{1,L+1},$$

$$K^*_{n,1}, K^*_{n,2}, \dots, K^*_{n,L} \xrightarrow{\text{extrapolation}} K^*_{n,L+1},$$

$$\hat{\kappa}^*_{t,L+1} = \frac{\sum_{i=1}^{n} K^*_{i,L}}{n} \quad \hat{\text{Var}}\left(\hat{\kappa}^*_{t,L+1}\right) = \frac{\sum_{i=1}^{n} (K^*_{i,L} - \overline{K^*_{i,L}})^2}{n-1} \tag{3.5}$$

3. Sometimes the restrictions are tight thus the number of the qualified reference sites is limited. In this case the size of the reference group is not large enough to remove the random effect. However, if the reference sites all have similar road section length and traffic flows and close to the treatment group, we could first extrapolate to forecast the occurrence of crash and then apply Hauer's method to calculate the expected accident count for each reference group in order to adjust the regression to the mean effect. The averages of the estimated expected accident count of the reference sites would be the wanted value. The brief algorithm is listed below:

$K^*_{i,1}, K^*_{i,2}, \dots, K^*_{i,L} \xrightarrow{\text{extrapolation}} K^*_{i,L+1}$ where i=1,…,n

$\hat{\alpha}_1, \dots, \hat{\alpha}_L, \hat{\alpha}_{L+1}, \hat{\beta}, \hat{b}$ and $\hat{\kappa}^*_{i,L+1}$ for i=1,…,n

The detailed algorithm to calculate $\hat{\alpha}_1, \dots, \hat{\alpha}_L, \hat{\alpha}_{L+1}, \hat{\beta}, \hat{b}$ and $\hat{\kappa}^*_{i,L+1}$ is the similar to the one showed in step 2.

$$\hat{\kappa}^*_{t,L+1} = \frac{\sum_{i=1}^{n} \hat{\kappa}_{i,L+1}}{n} \tag{3.6}$$

If the road section length and the traffic flow are different among the reference sites, then similar to the previous situation, after the extrapolation, we need to use Hauer's method to adjust the regression to the mean. However, the difference in road section length and traffic flow is still a disturbing factor. The best way to solve this problem is to use the road section length and the traffic flow of the treatment site to calculate its own expected accident count in the coming year. The algorithm is listed below:

$K^*_{i,1}, K^*_{i,2}, \dots, K^*_{i,L} \xrightarrow{\text{extrapolation}} K^*_{i,L+1}$ (where i=1,…,n)

$\hat{\alpha}_1, \dots, \hat{\alpha}_L, \hat{\alpha}_{L+1}, \hat{\beta}, \hat{b}$

$$\hat{\kappa}^*_{t,L+1} = \hat{E}\left(\kappa_{t,L+1}\right) = d_t \hat{\alpha}_{L+1} F_{t,L+1}^{\hat{\beta}} \tag{3.7}$$

The detailed algorithm to calculate $\hat{\alpha}_1, \dots, \hat{\alpha}_L, \hat{\alpha}_{L+1}, \hat{\beta}, \hat{b}$ is the similar to the one showed in step 2.

In the previous section 2.2.2 we have discussed the relationship between $\hat{\kappa}_{i,y}$ and $\widehat{E}(\kappa_{i,y})$, normally they are not equal to each other. However, since we could not get enough information of the post-treatment site, there is no way to distinct $\hat{\kappa}_{i,y}$ from $\widehat{E}(\kappa_{i,y})$. Therefore, we will assume that they are equal. That is

$$\hat{\kappa}_{i,y} = \widehat{E}(\kappa_{i,y}) \tag{3.8}$$

3.3 Summary

In this chapter, we propose the systematic method to proactively To provide an accurate input for the optimization model, a combined method with Hauer's coherent method (reviewed and discussed in Chapter 2) and extrapolation was proposed and illustrated. Limitations of our method include that: First, though the RTM problem is properly settled by the EB method, the potential existence of crash migration problems still need to be discussed and solved. Second, the Hauer's coherence method imposes some assumptions about the probability distribution of crash occurrences. The estimate tends to be inaccurate when the assumption is not met. The method proposed in section 3.2 to forecast road section crash number in the coming year can be refined too by exploring more extrapolation methods. Algorithms discussed in Chapter 2 and developed in this Chapter are used in cooperation in the decision support system we proposed in this project.

Next, we will discuss the design and development of our software system.

Chapter 4. GIS-Based Decision Supporting Tool

During the performance of real world case studies including the recent highway 169 projects, it was identified that the lack of unified data sources and formats made the data collection a very time consuming and burdensome task. To ease the traffic engineers' job in data collection, it will be helpful to link the various data sources together and transform them into a format that can be used as input directly to our model and the CMF's in the *Highway Safety Manual* [1]. Therefore, we propose the use of a GIS based decision supporting tool to merge the cartography, database, and statistical analysis together. (A geographic information system is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographical data.) Next, we illustrate the design of the software.

4.1 Overall design of the GIS based decision supporting tool

In our proposed decision supporting tool, the GIS user interface allows the users to select the sites under consideration for treatment directly from the map. Once a map is open, the user can click the icons on the tool bar to zoom in and zoom out the map. By clicking on a road segment or intersection, a drop down list will appear that shows the details of the selected road/intersection and enable the users to add it to the selection. Once a road segment or intersection is selected, it will be highlighted in different colors, as the ones shown in Figure 4.1. The interface also enables the traffic engineers to add new road(s)/intersection(s) to be constructed by adding a new layer of shape file to the map. Layers of the shape files can be turned on and off to have them shown or hidden on the map interface. For example, in the left column of the user interface shown in Figure 4.1, the three items under "Map Contents – StLouis" represent three layers of shape files that contained the map, the AADT data of the road, and the crash data. As they are all checked on, the roads are shown on the map, the AADT data can be viewed if one clicks on a road segment, and information on the crashes, which are shown as dark red dots along the roads, can be viewed too. If, for example, the first item is checked off, then the dark red dots representing the individual crashes will not be shown on the map.

Once the selection of treatment sites being considered is done and the proposed new roads/intersections are added, the user can click "Next" button to go to the next step, where potential treatment including many ITS and LCPSI strategies can be pre-selected, as shown in Figure 4.2. The purpose of this second step is to add the ITS and/or LCPSI strategies that the traffic engineers feel would be helpful to treat the selected sites. For each road segment and intersection, if new constructions are proposed and added to the map, the system will compare if treating the existing site using those strategies represents a better option than the new constructions, and if yes, which strategies should be used. After a total budget in unit of $1000 is input, clicking the "Suggest" button will lead the users to the result page, as shown in Figure 4.3, where the road segments and intersections to be treated as well as the treatments for each will be suggested by the software. Total costs associated with the treatments will also be shown.

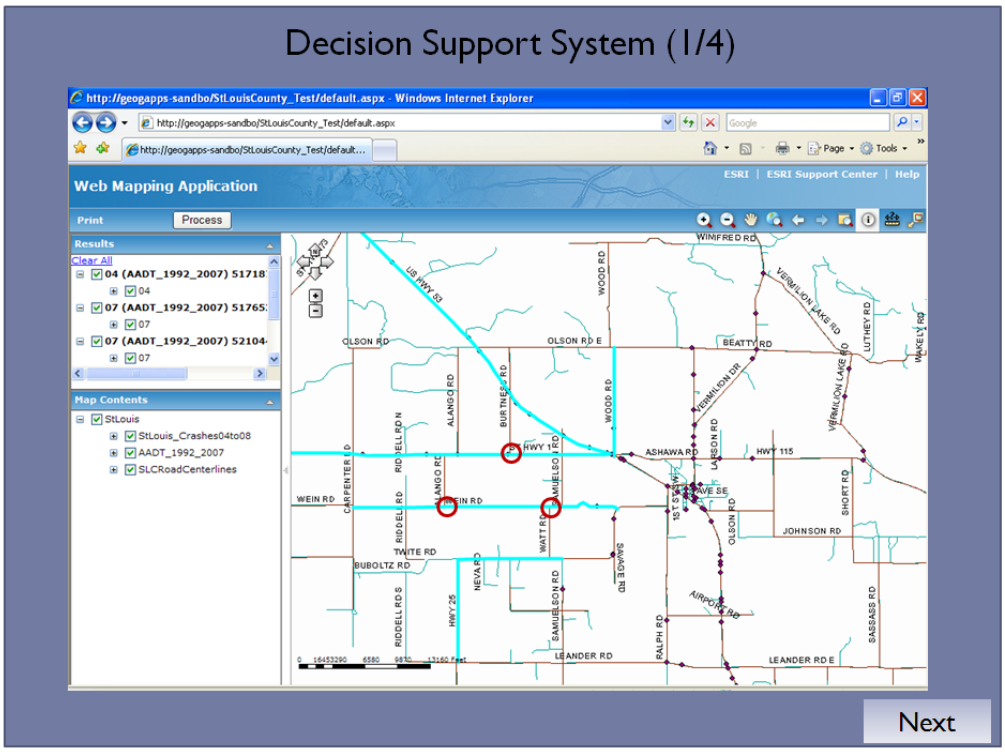

Figure 4.1: Decision Support System Design 1

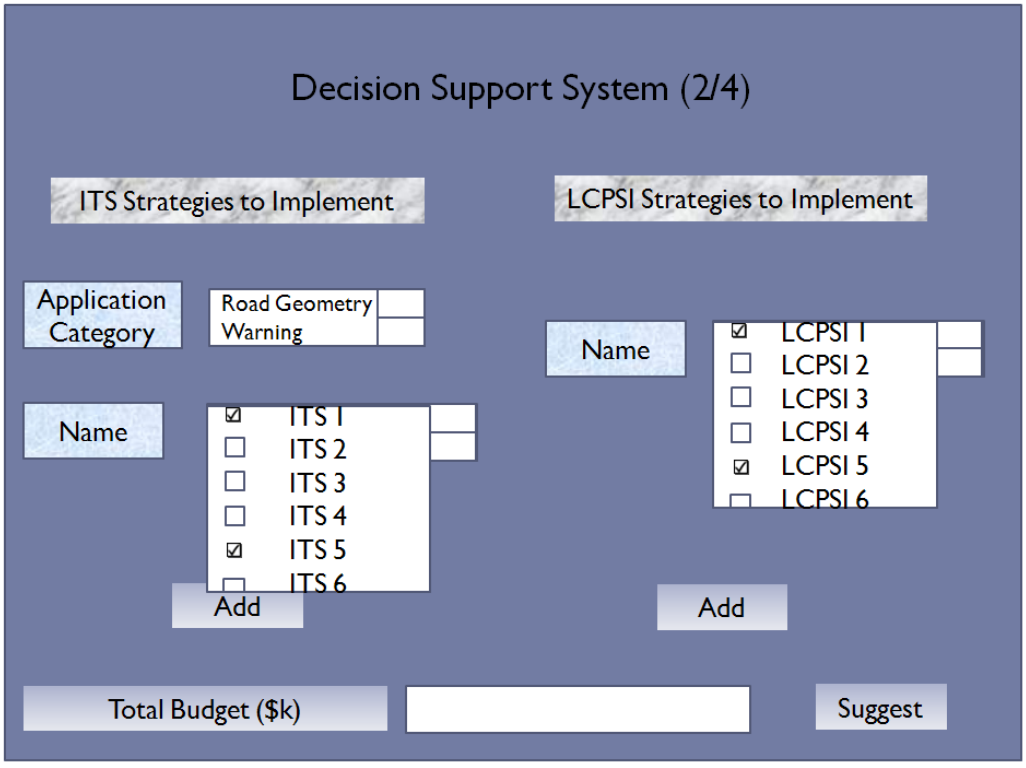

Figure 4.2: Decision Support System Design 2

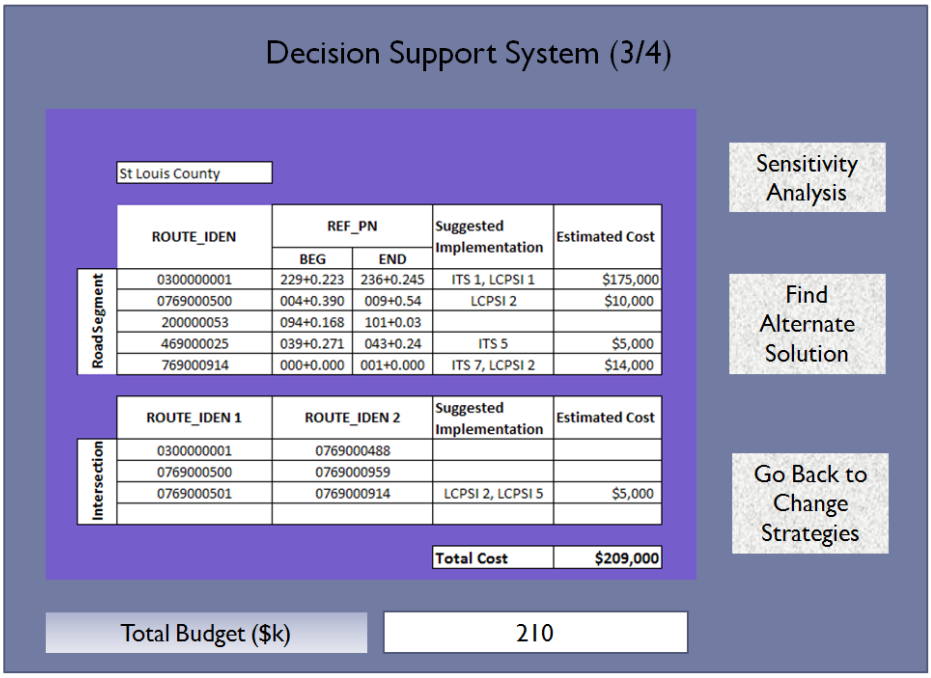

Figure 4.3: Decision Support System Design 3

The software should also have a supporting database where the ITS/LCPSI strategies implemented and their implementation locations can be added to the decision support system, so that their effectiveness in reducing specific crash types under different conditions can be assessed using the method discussed in Chapter 2. Figure 4.4 shows the preliminary design of the input page of the data based.

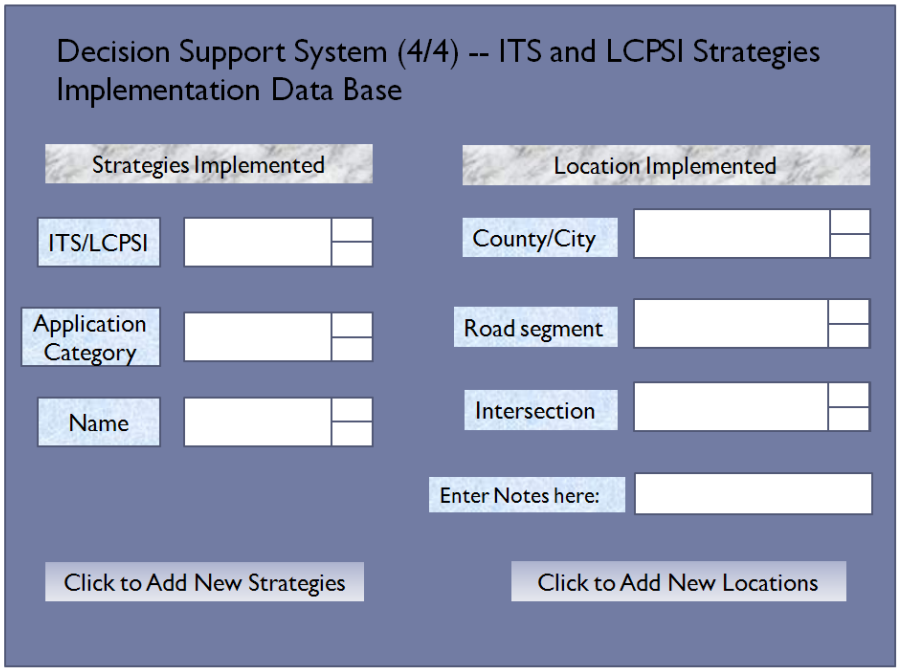

Figure 4.4: Decision Support System Design—Database Input Page

4.2 Detailed design of the GIS based decision supporting tool

Besides using the methods proposed in Chapter 3 to predict the expected average number of crashes for a facility/site, we also adopt the predicted average crash frequency function and crash modification functions provided in the *Highway Safety Manual* [1] for certain specific facility/site types. In general, the predicted average crash frequency is given by the following equation.

$$N_{predicted} = N_{spf\,x} \times \left(CMF_{1x} \times CMF_{2x} \times \ldots \times CMF_{yx} \times \ldots \times CMF_{Yx}\right) * C_x \quad (4.1)$$

Where N_{spf} is the predicted average crash frequency for base condition of site type x.

For rural two-way two-lane roads, $N_{Rural\ 2-lane} = AADT \times L \times (365 \times 10^{-6}) \times e^{-0.312}$

CMF_1 through CMF_Y are the crash modification functions to this site type with y geometric design and control features.

For example, $CMF_{Horizontal\ curve} = CMF_{3r} \times CMF_{4r}$

$$CMF_{3r} = \frac{(1.55 \times L_c) + \frac{80.2}{R} - (0.012 \times S)}{(1.55 \times L_c)}$$

$$CMF_{4r} = \begin{cases} 1, if\ SV < 0.01 \\ 1 + 6 \times (SV - 0.01), if\ 0.01 \leq SV < 0.02 \\ 1.06 + 3 \times (SV - 0.02), if\ SV \geq 0.02 \end{cases}$$

C_x is the calibration factor to adjust SPF for local conditions for site type x.

$$C = \frac{\sum_{all\ sites} Observed\ Crashes}{\sum_{all\ sites} Predicted\ Crashes}$$

After studying the current shapefiles available from MnDOT, we summarized the availability of data needed as input to analyze rural two-way two-lane roads. We separated the data into three general groups: basic information, geometric design features, and traffic control features and site characteristics. We found that all the data that were not readily available could be either calculated from or added to the shapefiles to the GIS system. Table 4.1 shows the summary of the data in these different groups.

Table 4.1: Data needs summary for the GIS tool

Basic Information	**Readily available**	**Can be calculated**	**Can be added to the shape files**
AADT	X		
Geometric Design Features	**Readily available**	**Can be calculated**	**Can be added to the shape files**
Length of horizontal curve (mile)		X	
Radius of horizontal curve (feet)		X	
Presence of spiral transition curve		X	
Super elevation of horizontal curve and the maximum super elevation		X	
Grade (percent)	X		
Roadside hazard rating			X
Traffic control features and site characteristics	**Readily available**	**Can be calculated**	**Can be added to the shape files**
Length of segment (miles)	X	X	
Lane width (feet)			X
Shoulder width (feet)			X
Shoulder type (paved / gravel / composite / turf / combined)			X
Driveway density (driveways per mile)		X	X
Presence of centerline rumble strips			X
Presence of a passing lane			X
Presence of a short four-lane section			X
Presence of a two-way left-turn lane			X
Presence of roadway segment lighting			X
Presence of automated speed enforcement			X

To calculate the calibration factor, C, the following values are also needed.

- Number of single-vehicle run-off-the-road crashes
- Number of multiple-vehicle head-on crashes
- Number of opposite-direction sideswipe crashes
- Number of same-direction sideswipe crashes
- Number of driveway-related crashes
- Number of nighttime fatality or injury crashes for unlighted roadway
- Number of nighttime property damage only crashes for unlighted roadway
- Total number of nighttime crashes for unlighted roadway
- Total number of nighttime crashes for unlighted intersections

Since all of these data can be easily calculated from the crash data collected by the state patrol, we developed software to help the traffic engineers to easily store and transfer those collected data in shape files and use GIS as the interface of the decision support system. As a result, software interfaces were designed, and algorithms and software code were developed to calculate

and store the needed data in the shape files for the GIS based decision support system. While the GIS coding part of the project is on-going, we report the parts that have been completed so far, which are the identification of horizontal curves on a road segment and the calculation of the curve radius.

In the following example, we illustrate how a road segment is evaluated and compared to a proposed new road using the crash modification factors calculated by the decision support system based on the *Highway Safety Manual* [1]. Figure 4.5 shows the starting user interface before any selection is made. The menu on the right hand side includes parameters related to the geometric design features, traffic control features and site characteristics.

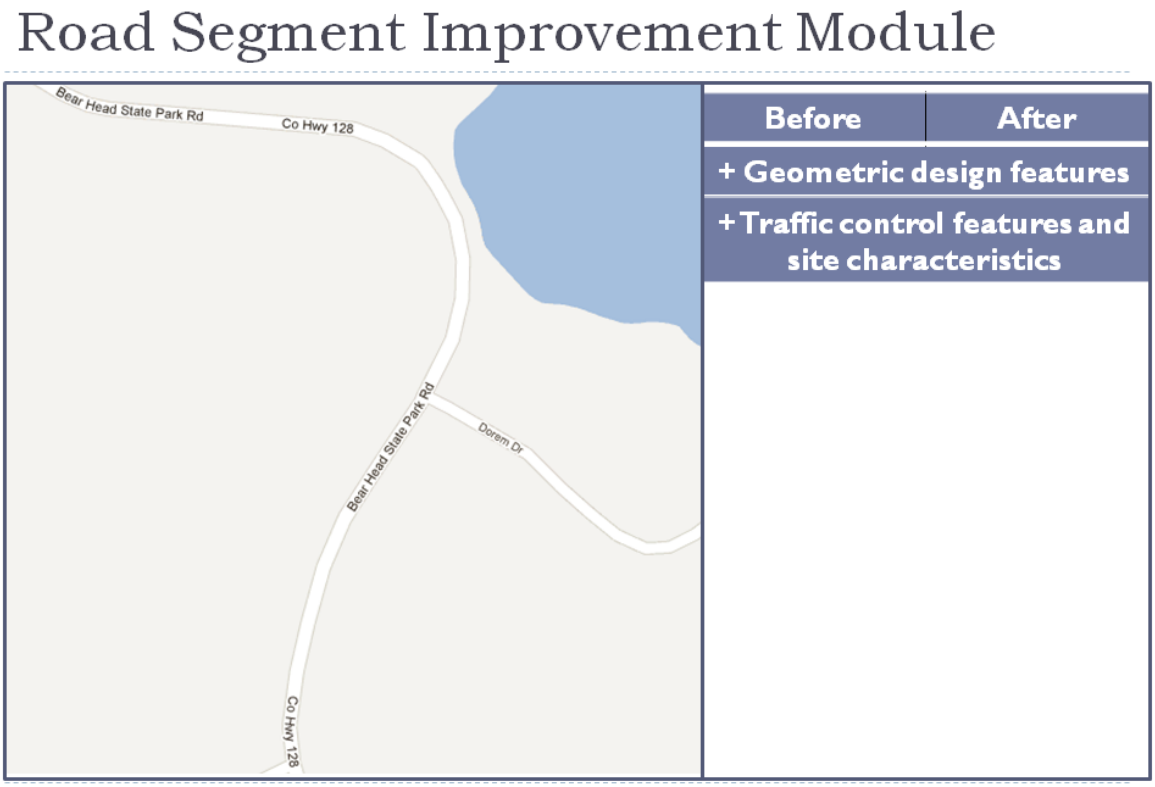

Figure 4.5: User Interface 1

Once a road segment is selected and the proposed new road is added, we can view and edit the related parameters and calculate the crash modification factors on those road segments in the right hand side menu, as shown by Figure 4.6. Parameters of the existing road segments are displayed under the "Before" tab while those of the proposed change are displayed under the "After" tab. We will start from the geometric design feature related parameters. Clicking on the "+" sign on the "Geometric design feature" tab will activate the full down list where the geometric design feature related parameters of the road segments will be shown. For the road segment to be treated, if horizontal curves are involved, the software automatically identifies the curves and shades them in different colors based on the degree of curvature. It also differentiates the different curves on each road segment, such as the #1, #2, and #3 curves shown in Figure 4.6, if there are multiple ones on a road. Note that, the parameters for the proposed new road can be

calculated automatically by the software once the shapefile is added to the system. Users can also type in the values and even try different numbers for analysis purpose.

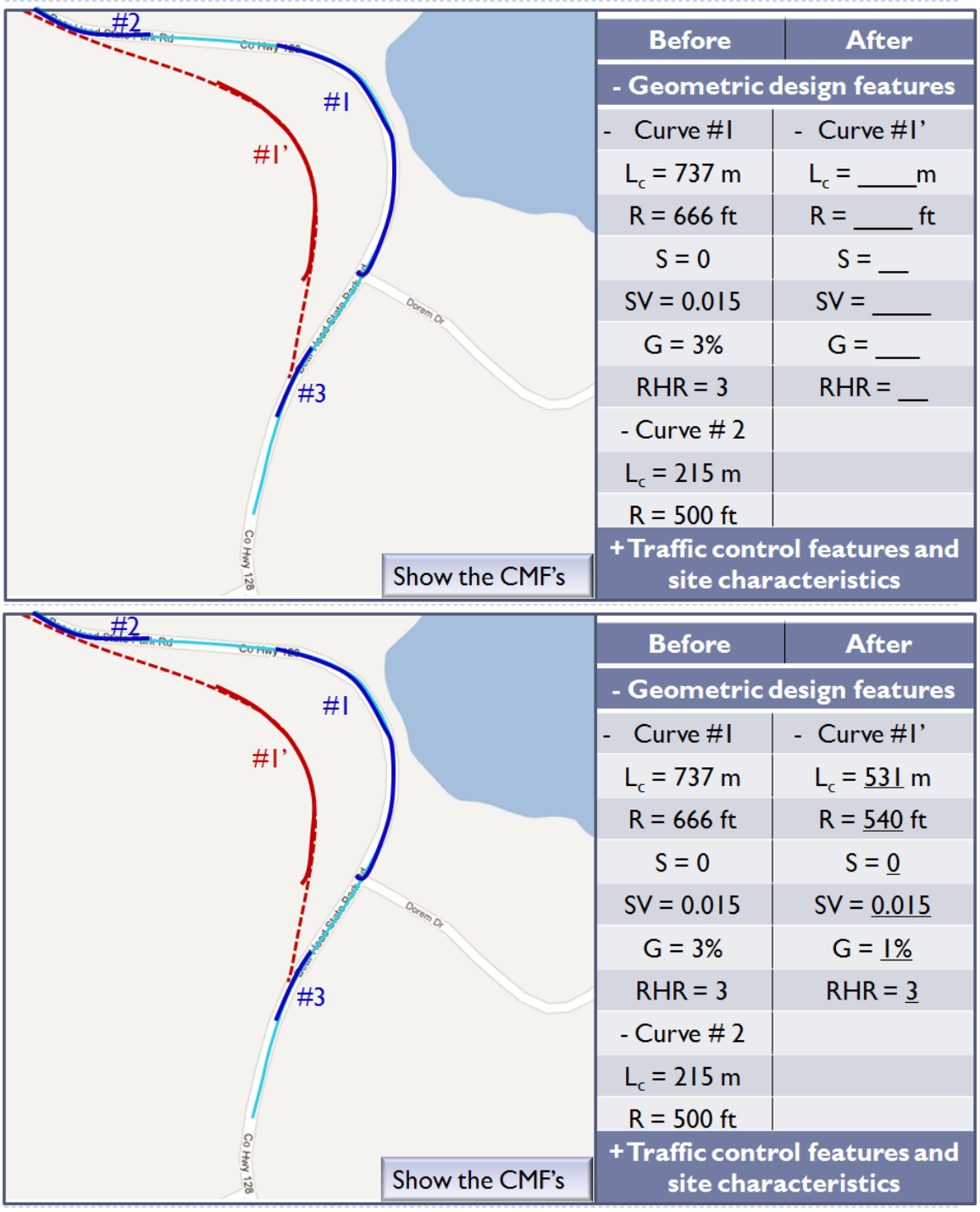

Figure 4.6: User Interface 2

Once the geometric design feature related parameters are input/calculated by the system and reviewed by the user, the user can click on the button "Show the CMF's" to see the resulting

crash modification factors as a result of these features. In our example, the curve related crash modification factors are calculated by the software and displayed in Figure 4.7.

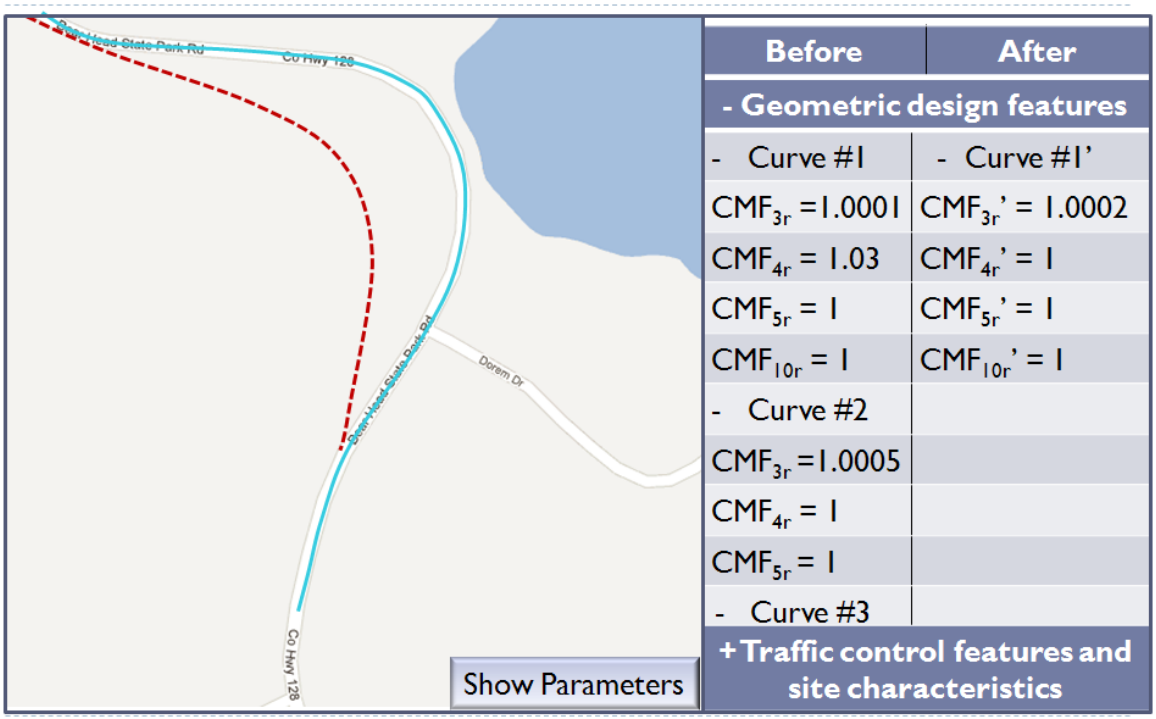

Figure 4.7: User Interface 3

Next, we can close the geometric design feature related calculations by clicking on the "-" sign on the "Geometrid design feature" tab and open the traffic control feature and site characteristics related parameters by clicking the "+" sign on its tab. Information imbedded in the shape files will be displayed automatically, such as the lane widths and shoulder widths on both directions of the road, as shown in Figure 4.8. Again, for the proposed new road, if such information is attached to the shape file uploaded to the system, the values will be displayed automatically; otherwise, they need to be typed.

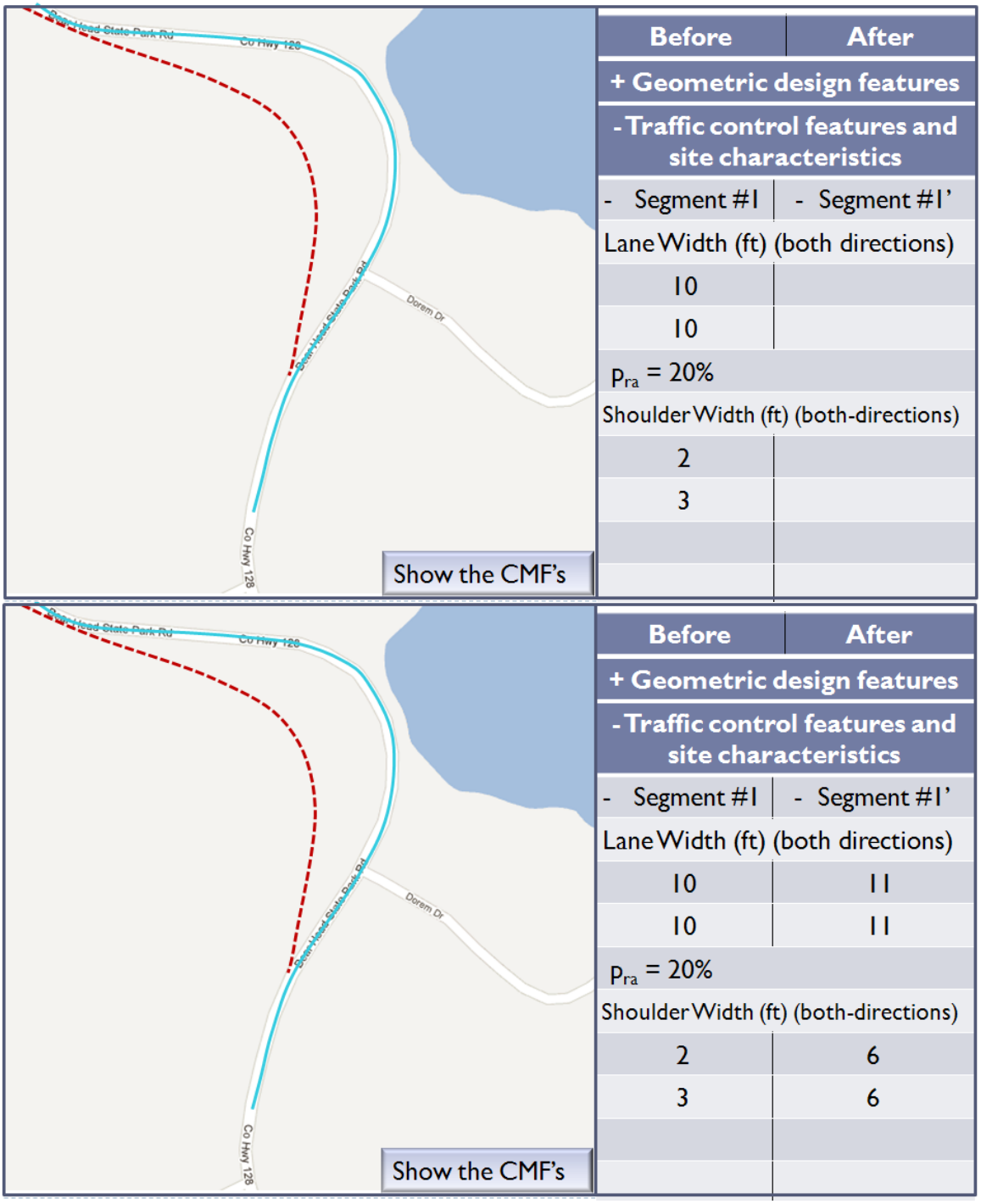

Figure 4.8: User Interface 4

Following the same step, if we click "Show the CMF's" button, we can see the crash modification factors calculated by the system using these parameters, as shown in Figure 4.9.

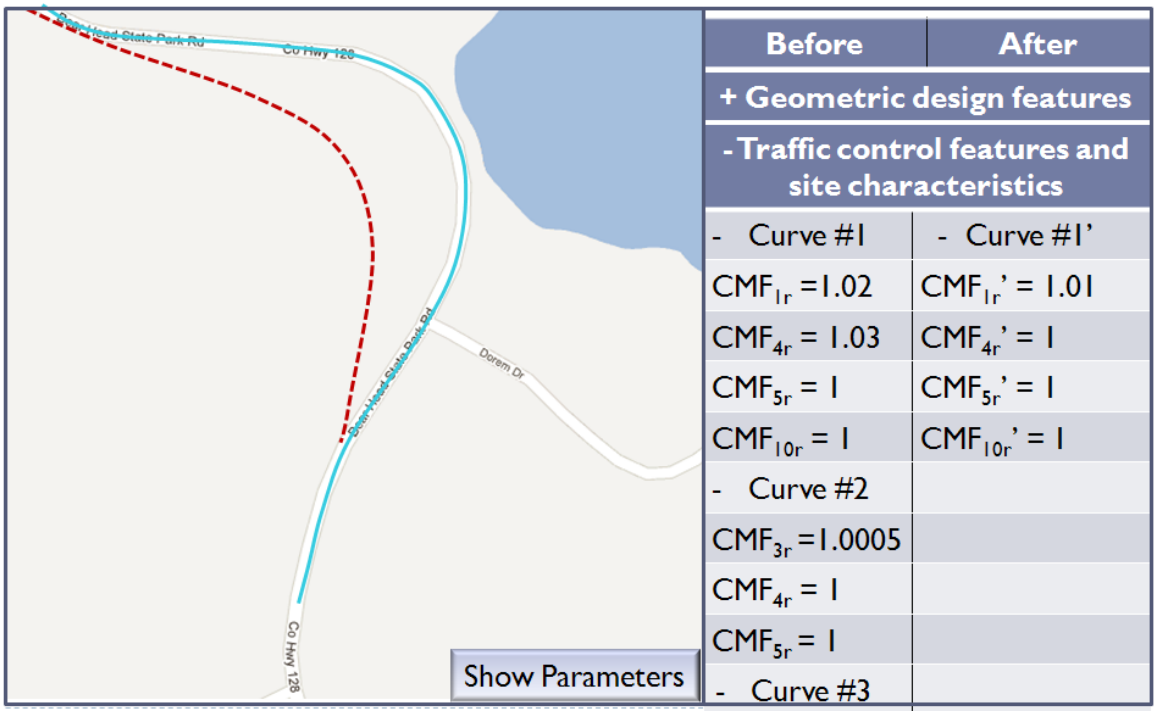

Figure 4.9: User Interface 5

It should be noted that the calculation of all the CMF's follows the formulas giving in the *Highway Safety Manual* [1]. While all the parameters shown in Table 4.1 as "Can be calculated" are coded or being coded in our decision support system, we illustrate the calculation of horizontal curve's radius in the following section.

4.3 Polyline curve analysis algorithm

The code used to calculate the radius of horizontal road curves in its entirety is listed in Appendix A. Here we break it up and explain the sections consecutively in the way that the computer executes it. First, it is worth noting that python is an untyped programming language that uses intentation in place of brackets {} for defining its structure.

The program starts by linking in the software libraries that it will be using further in, using the following commands:

```
import sys
import os
import arcpy
import math
import arcgisscripting
```

It then declares the location of the shapefile containing the highway shapefile and checks, using the arcpy library, to see if it is present before continuing. Here we used the highway 35 E as an example :

```
infc = "K:/MNDOT_Project_GIS/mnDOT/roads/highway35eSolid.shp"
print arcpy.Exists(infc)
```

“RoadPtClass” is the definition for a class, a data object that can be created and deleted from memory with specific ways to get and set its data members. It was created so that all the newly calculated data from the radius equation can be stored in a single array. No object is created in this reading but the recipe, as it were, is stored in memory so that objects of this type can be created later.

```
class roadPtClass:
    radius = 0
    avgRadiusPerCurve = 0
    x = 0
    y = 0
    rPtX = 0
    rPtX = 0

    def __init__(self, nX, nY):
        self.x = nX
        self.y = nY
        print "object instantiated"
        print self.x
        print self.y
    def __init__(self, nX, nY, nR):
        self.x = nX
        self.y = nY
        self.radius = nR
        print "object instantiated"
        print self.x
        print self.y
    def __init__(self, nX, nY, nR, nRx, nRy):
        self.x = nX
        self.y = nY
        self.radius = nR
        self.rPtX = nRx
        self.rPtY = nRy
        print "object instantiated"
        print self.x
        print self.y
    def setXY(self, nX, nY):
        self.x = nX
        self.y = nY
    def getX(self):
        return self.x
    def getY(self):
        return self.y
    def getR(self):
```

```
    return self.radius
def getRX(self):
    return self.rPtX
def getRY(self):
    return self.rPtY
def setRadius(self, nR):
    radius = nR
def setAvgRadius(self, nAvgR):
    radius = nAvgR
def clear(self):
    radius = 0
    avgRadiusPerCurve = 0
    x = 0
    y = 0
```

The following codes create the SearchCursor which allows you to go through the data in the shapefile row by row, accessed by the row's object. The "for row in rows" does just that. Then for each row it stores the information from that row in "feat".

```
# Identify the geometry field
#
desc = arcpy.Describe(infc)
shapefieldname = desc.ShapeFieldName
# Create search cursor
#
rows = arcpy.SearchCursor(infc)

# Enter for loop for each feature/row
#
for row in rows:
    # Create the geometry object 'feat'
    #
    feat = row.getValue(shapefieldname)
    print feat.type
    print feat.length
    print feat.pointCount
```

Next, an array is created to store the points that when connected make up the line data that is the highway. And loops through the line "feat" storing the X and Y data in the ptArray array. The "else:" clause is to handle the case that there is a road completely circling another road, although unlikely in a highway scenario it is common when dealing soley with polyline data in other fields.

```
    partnum = 0
    ptArray = arcpy.Array()

    for part in feat:
```

```
print "Part %i:" % partnum #part num 0 for the one highway polyline

# Step through each vertex in the feature
for pnt in feat.getPart(partnum):
   if pnt:
      ptArray.append(arcpy.Point(pnt.X, pnt.Y))
      print pnt.X
      print pnt.Y
   else:
      # If pnt is Null, it's from an interior polyline
      print "Interior Ring:"
partnum += 1
```

The following portion of code calculates the radius values for each point along the road. It starts by instantiating a new array to store the roadPtClass objects which will be created for each radius calculated. After that, it stores the number of points that make up the line. It then starts with the second point, which is point one since in computer science you start counting from point zero. Then you enter the loop which will execute for the number of points minus one.

To calculate the radius of curvature at a given point along the line, the X and Y coordinates of the points immediately in front of and behind it are stored for all the Cartesian coordinate math to follow. The coordinate values must be stored in float to maintain the decimal values and high precision throughout the calculation. If the data were stored as integers, the decimal values would not be preserved between computations.

The slopes between each pair of points and the midpoints are calculated and then using the pythagorean theorem,the y intercepts are calculated. Then to get the lines that intersect the midpoints of those lines perpendicularly, we take the inverse of that original slope (*1 / abM*) and multiply it by negative one (*(-1) * (1 / abM)*). Then the y intercepts for the new slopes are calculated.

Next, to find the intercept point of the two new lines, the y values are set to be equal, in order to get the y value of the intercept point. That point is plugged into one of the interceping lines to get the X coordinate. The distance equation is used to get the radius by calculating the difference between the intercept point and the middle point of the three points used to calculate the curve.

The data is then used to create a roadPtClass object and stored in the pointArray Array object. Which stores the geographic coordinates of the point on the line in which the radius is associated, the radius, and the intercept point coordinates which may be useful if further computation is required.

The division by zero cases are handled by simply testing the output of the offending calculation and manually changing the values since division by zero will crash most computer programs.

The codes are listed as follows:

```
#calculate the radii

    #frame ptSlots a,b,c, r
    #slopes ab,bc, abt, bct
    #radius radius
    pointArray = []
    u = 1
    ptCount = ptArray.count

    while (u < ( ptCount - 1)):
        a = ptArray[(u - 1)]
        b = ptArray[u]
        c = ptArray[(u + 1)]

        aX = float(a.X)
        aY = float(a.Y)
        bX = float(b.X)
        bY = float(b.Y)
        cX = float(c.X)
        cY = float(c.Y)

        abM = (bY - aY)/(bX - aX)
        #solve y = mx + b for the b
        abB = aY - ( aX * abM)

        print "slope from a to b"
        print abM
        print "y intercept for slope from a to b"
        print abB

        bcM = (cY - bY)/(cX - bX)
        #solve y = mx + b for the b
        bcB = bY - ( bX * bcM)

        print "slope from b to c"
        print bcM
        print "y intercept for slope from b to c"
        print bcB

        if (abM == 0.0):
            abtM = float(1)
        else:
            abtM = ( (-1) * ( 1 / abM) )
        print "inverse slope from a to b"
```

```
    print abtM

    #solve y = mx + b for the b
    #using a-b midpoint as the (x,y)
    abx = (( aX + bX ) / 2 )
    aby = (( aY + bY ) / 2 )
    print "mid point"
    print abx
    print aby
    print "reverse B"
    abtB = aby - ( abx * abtM)
    print abtB

    if (bcM == 0.0):
       bctM = float(1)
    else:
       bctM = ( (-1) * ( 1 / bcM) ) #div by zero possible
    #solve y = mx + b for the b
    #using a-b midpoint as the (x,y)
    bcx = (( cX + bX ) / 2 )
    bcy = (( cY + bY ) / 2 )
    bctB = bcy - ( bcx * bctM)

    print "inverse slope from b to c"
    print bctM
    print "mid point"
    print bcx
    print bcy
    print "reverse B"
    print bctB


    #solve for the intercept point (ry)
    rX = (bctB - abtB) / (abtM - bctM)
    rY = (bctM * rX) + bctB

    print "mid point is:
==============================================="
    print rX
    print rY

    #math.pow(x,y) x raised to the power y
    #radius is the difference between b and r
    diffX = abs(abs(rX - abx) + abs(rX - bcx))/2
    diffY = abs(abs(rY - aby) + abs(rY - bcy))/2
    print diffX
```

```
        print diffY
        radius = math.sqrt((math.pow(diffX,2))+(math.pow(diffY,2)))

        print "and Radius is:"
        print radius

        pointArray.append(roadPtClass(bX, bY, radius, rX, rY))

        u += 1
```

Next, the information to create the new shapefile which will contain the radius data is created by the following code.

```
    print "And now the Radii
==================================================="

    print "Now Creating new Shapeile"

    outShape = "K:\\MNDOT_Project_GIS\\mnDOT\\output\\newRoad.shp"
    outShapePath = os.path.dirname(outShape)
    outShapeName = os.path.basename(outShape)
```

This code deleted the previous file of the same name if it exists.

```
    try:
        arcpy.Delete_management(outShape)
        print "existing shapefile deleted"
    except:
        pass
```

This code creates the new shapefile and the attributes that will be filled with the data stored in the roadPtClass objects stored in the pointArray. The DefineProjection is necessary so that the mapping software knows how to project it.

```
    arcpy.CreateFeatureclass_management(outShapePath, outShapeName, "POLYLINE")
    arcpy.AddField_management(outShape, "RADIUS", "LONG", "10")
    print "radius field created"
    arcpy.AddField_management(outShape, "RAD_PT_X", "LONG", "10")
    arcpy.AddField_management(outShape, "RAD_PT_Y", "LONG", "10")
    print "radius x & y fields created"
    insertCur = arcpy.InsertCursor(outShape)
    arcpy.DefineProjection_management(outShape,
"PROJCS['NAD_1983_UTM_Zone_15N',GEOGCS['GCS_North_American_1983',DAT
UM['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],P
RIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transv
erse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northin
g',0.0],PARAMETER['Central_Meridian',-
```

```
93.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UN
IT['Meter',1.0]]")
```

The following portion of code is for the first road segment which only has a radius value from the second point so it is a special case. It creates an array of points made up of the first and second points and assigns it the radius value from the second point. This new line "feat" is then inserted into the new shapefile "insertCur.insertRow(feat)" then the array is emptied and the point objects are zeroed.

```
#segment should be an average of the two radii
 # pt 0 - pt 1 = radius of pt 1
 # pt 1 - pt 2 = raius of pt1 + radius of pt2 / 2
 ptObj = arcpy.Point()
 arObj = arcpy.Array()
 myCounter = 0

 #for a in pointArray:
 #manually do 0-1
 #
 a = pointArray[0]
 ptObj.X = a.getX()
 ptObj.Y = a.getY()
 arObj.add(ptObj)

 ptObj.X = 0
 ptObj.Y = 0

 a = pointArray[1]
 ptObj.X = a.getX()
 ptObj.Y = a.getY()
 arObj.add(ptObj)

 feat = insertCur.newRow()
 feat.Shape = arObj
 feat.RADIUS = a.getR()
 feat.RAD_PT_X = a.getRX()
 feat.RAD_PT_Y = a.getRY()

 insertCur.insertRow(feat)

 arObj.removeAll()
 ptObj.X = 0
 ptObj.Y = 0

 #pntCount = pointArray.count
 #Since its actually a list and not an array you must use len(list)
 pntCount = len(pointArray)
```

```
print "pointArray Count is:"
print ptCount
```

Finally, the following code loops through the pointArray creating line segments from the data stored in the roadPtClass objects and inserting each segment into the new shapefile. The radius values for these segments is the average of the two points making up the line segments. And lastly the insertCur is deleted.

```
u = 1
#while(u < 4):
while (u < ( pntCount - 1)):
   a = pointArray[u]
   print a.getR()
   ptObj.X = a.getX()
   ptObj.Y = a.getY()
   print "1st NEW LINE POINT X and Y"
   print ptObj.X
   print ptObj.Y


   arObj.add(ptObj)
   ptObj.X = 0
   ptObj.Y = 0
   a = pointArray[u + 1]
   print a.getR()
   ptObj.X = a.getX()
   ptObj.Y = a.getY()
   print "2nd NEW LINE POINT X and Y"
   print ptObj.X
   print ptObj.Y
   arObj.add(ptObj)
   #do the same thing as above but for plus one place in array

            feat = insertCur.newRow()
   feat.Shape = arObj
   feat.RADIUS = a.getR()
   feat.RAD_PT_X = a.getRX()
   feat.RAD_PT_Y = a.getRY()

   insertCur.insertRow(feat)

   arObj.removeAll()
   ptObj.X = 0
   ptObj.Y = 0
```

u += 1
del insertCur

Using highway 35-E from near Esko to the intersection of 35-E and London road as an example, we show the software results that separate the road segment into different sections by their curve radius. The Highway 35-E data is polyline data from the Minnesota Department of Natural Resources GIS Data Deli web service. The software libraries used are those from ESRI's arcpy Python scripting libraries and the data was viewed in ESRI's arcMap software. Figure 4.10 shows this road segment on the GIS map.

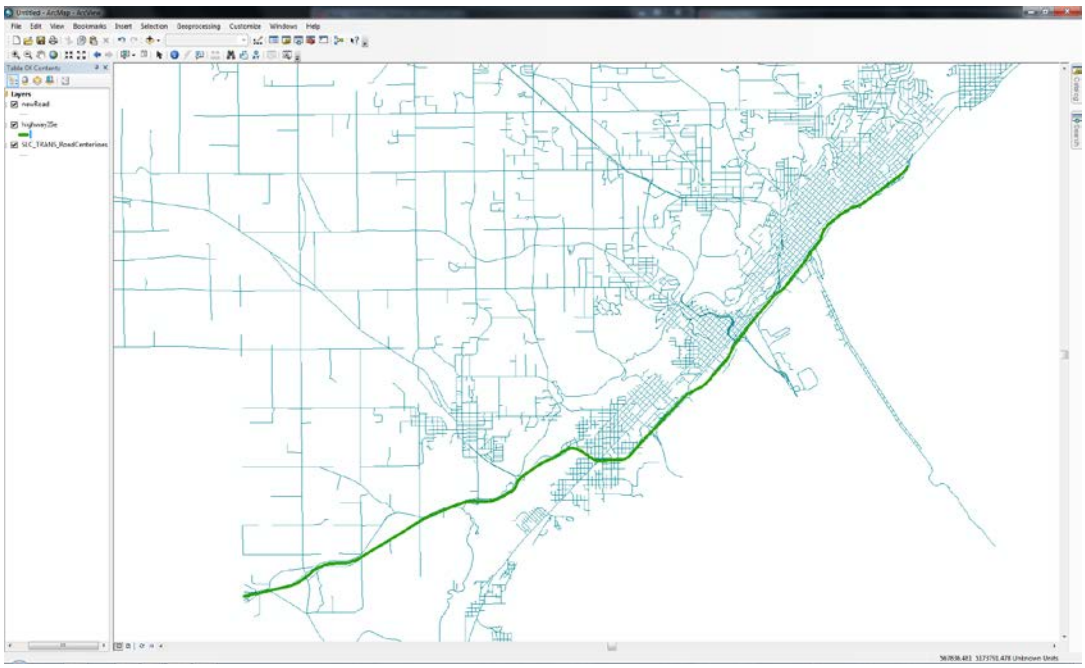

Figure 4.10: Highway 35-E on the User Interface

Figure 4.11 shows the road segment separated into different road sections based on the radius. As indicated by the legend, the darker the red color of a road section is, the smaller the curve radius is; or the lighter the road section's color is, the straighter that road section is.

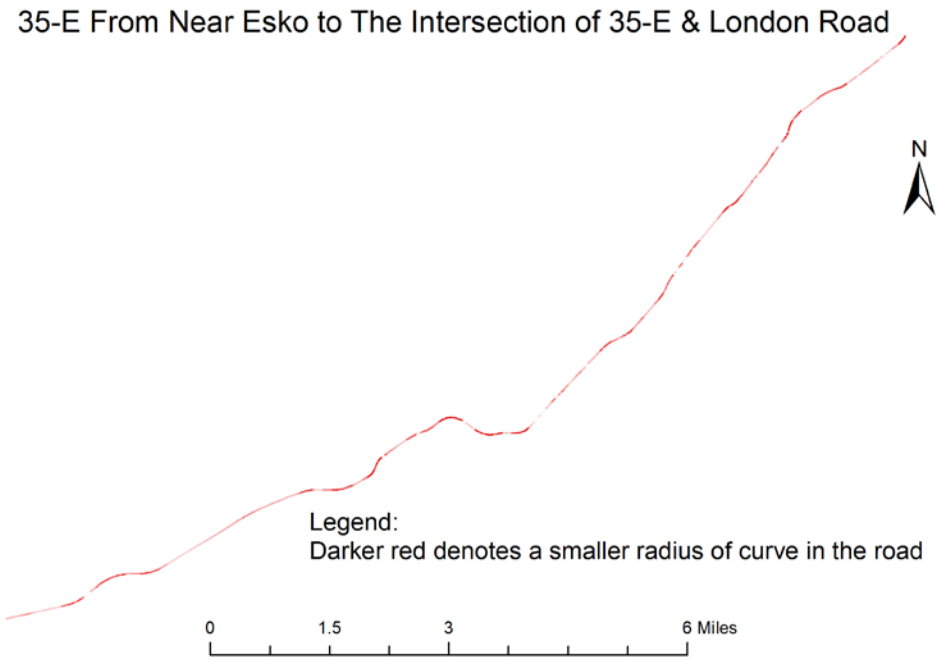

Figure 4.11: Curvature Degree of Road Sections on Highway 35-E

Actual radius data for all road sections were stored in the system and later used in the crash modification function as input to calculate the crash modification factors for each section in regard to horizontal curve. Calculation of the other crash modification factors follow similar suite, and we will not list all of them here.

Chapter 5. Conclusion and Future Work

In this project, we first reviewed and compared the traditional methods used in site selection and countermeasure evaluations. For those methods that lack complete or correct mathematical proofs in the original source, we provided the mathematical proofs to help verify the methods. Advantages and disadvantages of using each method were discussed and examples of using them were given. Based on the review, the research gap was identified. To better assist the traffic engineers in their resource allocation decisions, we proposed a decision support system that systematically optimizes the allocation of resources to treat the most needed sites with the most effective countermeasures. Underlying algorithm of this decision support system includes the methods developed in this project as discussed in Chapter 3 as well as methods evaluated in Chapter 2. Crash modification functions adopted in the *Highway Safety Manual* [1] were also incorporated into the system. As illustrated in Figure 5.1, we first provide a tool with GIS interface to extract road information from the shape files that MnDOT or other agents provide. Then the extracted information is used in the calculation to provide input to our optimization model. The decision support system intelligently selects the algorithms to be used in the calculation based on given data and preference. Requiring very limited manual input, the system automatically generates optimized scenarios for traffic safety strategy implementation.

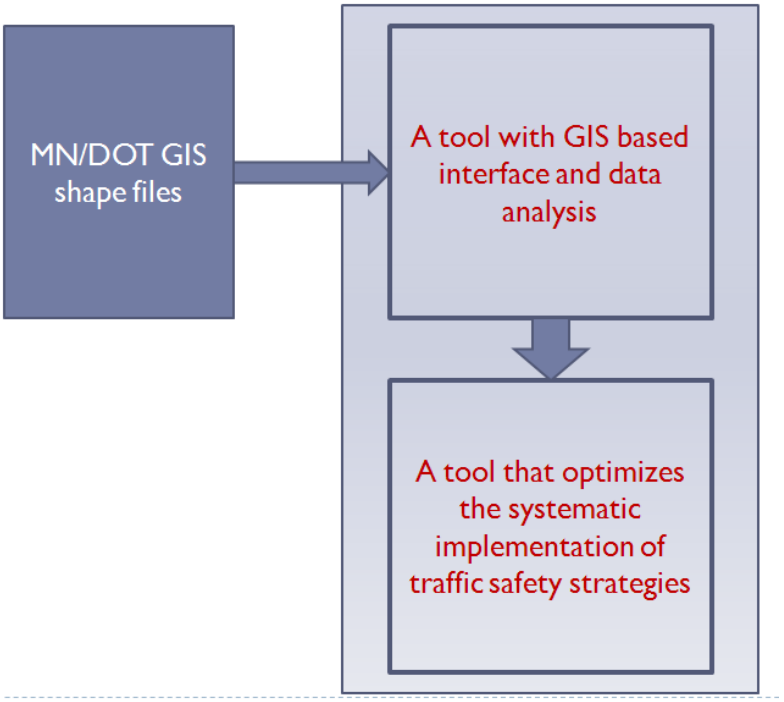

Figure 5.1: Components of the Decision Support System

Besides the limitations of the algorithm discussed at the end of Chapter 3, the software is also limited in its functions as a university research project prototype. Much more work is needed to refine the software to be bug free and user friendly. Future work can connect the current tool to the commercialized software, Safefy Analyst, so that they complement each other, as illustrated in Figure 5.2. By doing so, we expect to provide the Safety Anlyst users with better optimization

methods and a GIS interface that not only makes the site selection more visual but also saves tremendous data collection and preparation work.

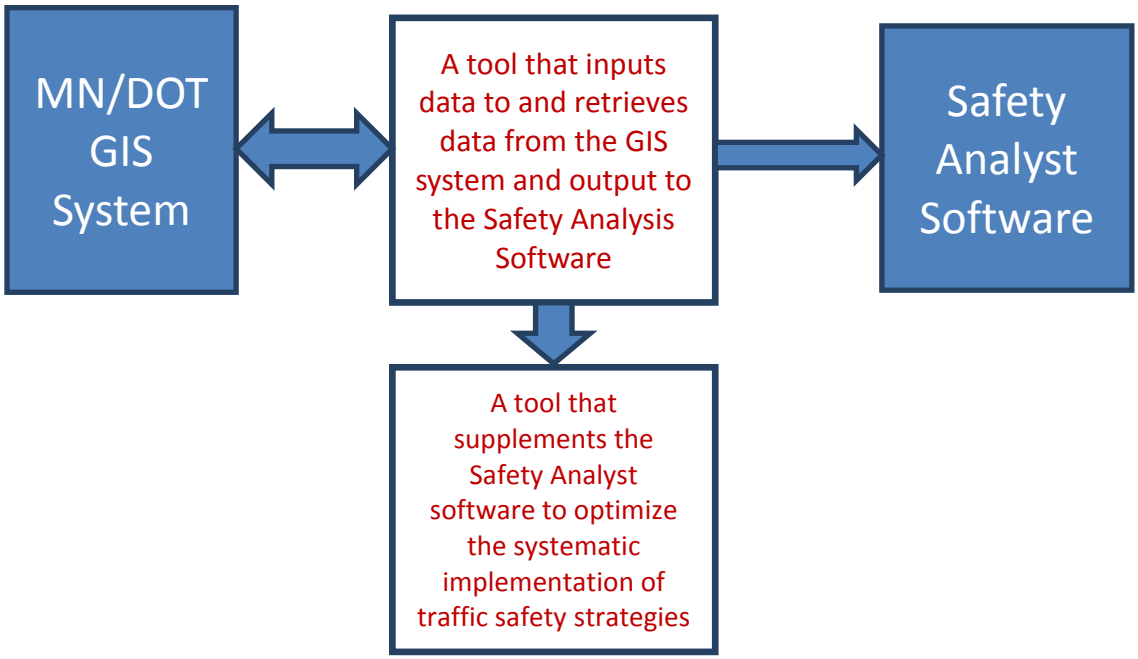

Figure 5.2: Future Work to Connect with Safety Analyst Software

References

1) *Highway Safety Manual*, 1st Ed., American Association of State Highway and Transportation Officials, 2011.
2) E. Hauer, *Observational Before-After Study's in Road Safety*. Pergamon, Oxford, U.K., 1997.
3) J. Shen and A. Gan, "Development of Crash Reduction Factors: Methods, Problems and Research Needs," *Transportation Research Record* 1840, 2003, pp. 50-56.
4) E. Hauer, D.W. Harwood, F. M. Council, and M.S. Griffith, "Estimating safety by the empirical Bayes method: A tutorial," *Transportation Research Record*, 2002, pp. 126-131.
5) G. Barnett, J. C. van der Pols, and A. J. Dobson, "Regression to the mean: what is it and how to deal with it," *Int J Epidemiol*, 2005, Vol. 34, pp. 215-220.
6) J. Boyle and C.C. Wright, "Accident 'migration' after remedial treatment at accident blakspot," *Traffic Engineering and Control*, 1984, Vol. 25, No.5.
7) R. Elvik, "Evaluations of road accident blackspot treatment: a case of the iron law of evaluation studies?" *Accident Analysis and Prevention*, Vol. 29, No 2, pp 191-199.
8) F. M. Council, D. W. Reifurt, B.J. Campbell, F. l. Roediger, C. L. Carroll, A.K. Dutt, and J. R. Dunham, *Accident research Manual*, Highway Safety Research Center, University of North Carolina, 1980.
9) W. Trochim and J. P. Donnelly, *Research Methods Knowledge Bases*, 3rd Ed., Atomic Dog Publishing, 2006 .
10) B.P. Carlin and T.A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*, 2nd Ed., Louis Boca Raton, FL: Chapman and Hall/CRC Press, 2000.
11) L. J. Bain and M. Engelhardt, *Introduction to Probability and Mathematical Statistics*, 2nd Ed., Duxbury Press, 1987.
12) R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, 6th Ed., Englewood Cliffs, NJ: Prentice-Hall, 1982.
13) R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
14) Brezinski and M. Redivo Zaglia, *Extrapolation Methods: Theory and Practice*, North-Holland, 1991.

Appendix A: Python Codes for the Polyline Curve Analysis

```
import sys
import os
import arcpy
import math
import arcgisscripting

infc = "K:/MNDOT_Project_GIS/mnDOT/roads/highway35eSolid.shp"

print arcpy.Exists(infc)

class roadPtClass:
    radius = 0
    avgRadiusPerCurve = 0
    x = 0
    y = 0
    rPtX = 0
    rPtX = 0

    def __init__(self, nX, nY):
        self.x = nX
        self.y = nY
        print "object instantiated"
        print self.x
        print self.y
    def __init__(self, nX, nY, nR):
        self.x = nX
        self.y = nY
        self.radius = nR
        print "object instantiated"
        print self.x
        print self.y
    def __init__(self, nX, nY, nR, nRx, nRy):
        self.x = nX
        self.y = nY
        self.radius = nR
        self.rPtX = nRx
        self.rPtY = nRy
        print "object instantiated"
        print self.x
        print self.y
    def setXY(self, nX, nY):
        self.x = nX
        self.y = nY
    def getX(self):
        return self.x
    def getY(self):
```

```
        return self.y
    def getR(self):
        return self.radius
    def getRX(self):
        return self.rPtX
    def getRY(self):
        return self.rPtY
    def setRadius(self, nR):
        radius = nR
    def setAvgRadius(self, nAvgR):
        radius = nAvgR
    def clear(self):
        radius = 0
        avgRadiusPerCurve = 0
        x = 0
        y = 0


# Identify the geometry field
#
desc = arcpy.Describe(infc)
shapefieldname = desc.ShapeFieldName
# Create search cursor
#
rows = arcpy.SearchCursor(infc)

# Enter for loop for each feature/row
#
for row in rows:
    # Create the geometry object 'feat'
    #
    feat = row.getValue(shapefieldname)
    print feat.type
    print feat.length
    print feat.pointCount
    partnum = 0
    ptArray = arcpy.Array()

    for part in feat:
        print "Part %i:" % partnum #part num 0 for the one highway polyline

        # Step through each vertex in the feature
        for pnt in feat.getPart(partnum):
            if pnt:
                ptArray.append(arcpy.Point(pnt.X, pnt.Y))
                print pnt.X
```

```
            print pnt.Y
        else:
            # If pnt is Null, it's from an interior polyline
            print "Interior Ring:"
    partnum += 1


#calculate the radii

#frame ptSlots a,b,c, r
#slopes ab,bc, abt, bct
#radius radius
pointArray = []
u = 1
ptCount = ptArray.count

while (u < ( ptCount - 1)):
    a = ptArray[(u - 1)]
    b = ptArray[u]
    c = ptArray[(u + 1)]

    aX = float(a.X)
    aY = float(a.Y)
    bX = float(b.X)
    bY = float(b.Y)
    cX = float(c.X)
    cY = float(c.Y)

    abM = (bY - aY)/(bX - aX)
    #solve y = mx + b for the b
    abB = aY - ( aX * abM)

    print "slope from a to b"
    print abM
    print "y intercept for slope from a to b"
    print abB

    bcM = (cY - bY)/(cX - bX)
    #solve y = mx + b for the b
    bcB = bY - ( bX * bcM)

    print "slope from b to c"
    print bcM
    print "y intercept for slope from b to c"
    print bcB
```

```
    if (abM == 0.0):
       abtM = float(1)
    else:
       abtM = ( (-1) * ( 1 / abM) )
    print "inverse slope from a to b"
    print abtM

    #solve y = mx + b for the b
    #using a-b midpoint as the (x,y)
    abx = (( aX + bX ) / 2 )
    aby = (( aY + bY ) / 2 )
    print "mid point"
    print abx
    print aby
    print "reverse B"
    abtB = aby - ( abx * abtM)
    print abtB

    if (bcM == 0.0):
       bctM = float(1)
    else:
       bctM = ( (-1) * ( 1 / bcM) ) #div by zero possible
    #solve y = mx + b for the b
    #using a-b midpoint as the (x,y)
    bcx = (( cX + bX ) / 2 )
    bcy = (( cY + bY ) / 2 )
    bctB = bcy - ( bcx * bctM)

    print "inverse slope from b to c"
    print bctM
    print "mid point"
    print bcx
    print bcy
    print "reverse B"
    print bctB


    #solve for the intercept point (ry)
    rX = (bctB - abtB) / (abtM - bctM)
    rY = (bctM * rX) + bctB

    print "mid point is:
==========================================="
    print rX
    print rY
```

```
        #math.pow(x,y) x raised to the power y
        #radius is the difference between b and r
        diffX = abs(abs(rX - abx) + abs(rX - bcx))/2
        diffY = abs(abs(rY - aby) + abs(rY - bcy))/2
        print diffX
        print diffY
        radius = math.sqrt((math.pow(diffX,2))+(math.pow(diffY,2)))

        print "and Radius is:"
        print radius

        pointArray.append(roadPtClass(bX, bY, radius, rX, rY))

        u += 1

    print "And now the Radii
==================================================="

    print "Now Creating new Shapeile"

    outShape = "K:\\MNDOT_Project_GIS\\mnDOT\\output\\newRoad.shp"
    outShapePath = os.path.dirname(outShape)
    outShapeName = os.path.basename(outShape)

    try:
        arcpy.Delete_management(outShape)
        print "existing shapefile deleted"
    except:
        pass

    arcpy.CreateFeatureclass_management(outShapePath, outShapeName, "POLYLINE")
    arcpy.AddField_management(outShape, "RADIUS", "LONG", "10")
    print "radius field created"
    arcpy.AddField_management(outShape, "RAD_PT_X", "LONG", "10")
    arcpy.AddField_management(outShape, "RAD_PT_Y", "LONG", "10")
    print "radius x & y fields created"

    insertCur = arcpy.InsertCursor(outShape)
    arcpy.DefineProjection_management(outShape,
"PROJCS['NAD_1983_UTM_Zone_15N',GEOGCS['GCS_North_American_1983',DAT
UM['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],P
RIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transv
erse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northin
g',0.0],PARAMETER['Central_Meridian',-
93.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UN
IT['Meter',1.0]]")
```

```
#segment should be an average of the two radii
# pt 0 - pt 1 = radius of pt 1
# pt 1 - pt 2 = raius of pt1 + radius of pt2 / 2
ptObj = arcpy.Point()
arObj = arcpy.Array()
myCounter = 0

#for a in pointArray:
#manually do 0-1
#
a = pointArray[0]
ptObj.X = a.getX()
ptObj.Y = a.getY()
arObj.add(ptObj)

ptObj.X = 0
ptObj.Y = 0

a = pointArray[1]
ptObj.X = a.getX()
ptObj.Y = a.getY()
arObj.add(ptObj)

feat = insertCur.newRow()
feat.Shape = arObj
feat.RADIUS = a.getR()
feat.RAD_PT_X = a.getRX()
feat.RAD_PT_Y = a.getRY()

insertCur.insertRow(feat)

arObj.removeAll()
ptObj.X = 0
ptObj.Y = 0

#pntCount = pointArray.count
#Since its actually a list and not an array you must use len(list)
pntCount = len(pointArray)

print "pointArray Count is:"
print ptCount

u = 1
#while(u < 4):
while (u < ( pntCount - 1)):
```

```
    a = pointArray[u]
    print a.getR()
    ptObj.X = a.getX()
    ptObj.Y = a.getY()
    print "1st NEW LINE POINT X and Y"
    print ptObj.X
    print ptObj.Y


    arObj.add(ptObj)
    ptObj.X = 0
    ptObj.Y = 0
    a = pointArray[u + 1]
    print a.getR()
    ptObj.X = a.getX()
    ptObj.Y = a.getY()
    print "2nd NEW LINE POINT X and Y"
    print ptObj.X
    print ptObj.Y
    arObj.add(ptObj)
    #do the same thing as above but for plus one place in array

                feat = insertCur.newRow()
    feat.Shape = arObj
    feat.RADIUS = a.getR()
    feat.RAD_PT_X = a.getRX()
    feat.RAD_PT_Y = a.getRY()

    insertCur.insertRow(feat)

    arObj.removeAll()
    ptObj.X = 0
    ptObj.Y = 0

    u += 1
del insertCur
```