In this section, a numerical example is provided. Exercise Three sensors were deployed in an environment to measure the concentration of ethanol in the environment with the following results: S1 max 0. Example In this section, how to compute covariance and covariance matrices are introduced first, followed by their use in Kalman filtering. The following definitions apply for the rest of the discussions in this chapter.

The manual method is tedious and prone to errors. The following matrices are therefore defined in this section. It is the error in prediction or estimate Q is the process noise covariance matrix R is the measurement error covariance matrix K is the Kalman gain. These matrices change at each time epoch and therefore will be subscripted in the analysis to follow.

Observe from these equations, that the Kalman gain K is a ratio of the prediction error to the sum of the prediction error and measurement noise error. K. H is a transformation matrix. The Kalman gain is nearly one when the measurement error is nearly zero. The measurement error also occurs due to deficiencies in the sensor or measuring equipment. These two noise sources are accounted for in the general Kalman filter models using covariance matrices. In this section, the words noise and error will be used to mean the same thing. The following matrices are therefore defined in this section. It is the error in prediction or estimate Q is the process noise covariance matrix R is the measurement error covariance matrix K is the Kalman gain. These matrices change at each time epoch and therefore will be subscripted in the analysis to follow.

To avoid confusion in subscripts and difficulties in learning Kalman filtering, double subscripting of variables between previous state and current state is avoided. Subscripting of variables is limited to single subscripts. It is hoped that by doing so, it will become easier to gain traction in learning. The same diagram is retained in Figure 4. They are used to compare statistical data to establish the levels of similarity between them. In this section, how to compute covariance and covariance matrices are introduced first, followed by their use in Kalman filtering. The following definitions apply for the rest of the discussions in this chapter.

The variance is a positive number. Computer their means and variances. Temperature The manual method is tedious and prone to errors. The deviation matrix method is more tractable and amenable for software or programming approach. To illustrate the manual approach, the following exercise is given. Exercise Three sensors were deployed in an environment to measure the concentration of ethanol in the environment with the following results: S1 max 0. Example In this section, a numerical example is provided.
The following table contains readings from three sensors S7, S8 and S9. The covariance of each record is computed S7 0. The deviation of a matrix A is defined by the expression $|A|^2$. In other words, there is no dependence between stock B and stock C and G. Also, there is no dependence between stocks B and G. Those error variables are independent of each other. Also, there is negative covariance between stock C and G, which means when one goes up in price, the other goes down. The covariance matrices are highly informative. Entries in the matrices are products of standard deviations that allows each entry to be composed as products of two or more standard deviations.

Thus, for example, a process of the covariance matrix can be created rather quickly for a process of 4. If the velocity has standard deviation 1. The initial state and measurements are as provided. The objective of this section is to use the given data to provide a realistic example of how Kalman filter works. In this section, we describe the eight steps. All the error matrices are for simplicity set to zero. The control variable is however used in the processing. The eight steps are as follows: 1. State Prediction From Figure 4. The expression to use in creating the predicted covariance matrix is given in Figure 4. This is because they do not really affect the distance and velocity covariance values. If we set them to zero initially, we can speed up the computation of the predicted covariance matrix. We have chosen not to eliminate them to help the reader go through the overall computation of the predicted covariance matrix. H is called the observation matrix.

It converts the predicted covariance matrix into the correct form. To calculate the Kalman gain, the observation covariance error matrix is required. It is 4. The H matrix is an identity matrix. Many businesses have huge data archives from which decision could be reached on issues like which product to push to the market, which market and when. In logistics and transportation, routing of traffic to minimize delays remains an area of great interest. Taking a route among many routes could result in savings of fuel, reduce costs and save time. Genetic algorithms belong to a class of search algorithms by mimicking how biological processes evolve and by extension help to unravel how natural and commercial processes adapt to changing conditions. They can be used to design software tools for decision making and for designing robust systems based on the interrelationships between system parameters. In doing so, they belong to a class of optimization algorithms, the so-called evolutionary computing algorithms.

They are a class of optimization problems. For that reason, GAs are used to find maxima or minima of functions. They are typically sets of binary bits, which represent each member of the population. Therefore, it is necessary to create a fitness function which is used for optimization. It is used to produce the next generation of chromosomes. In this process, offspring are selected to reproduce; crossover is a concept that is borrowed from genetics. Normally, it is really a sexual reproduction process in which two parents mate and exchange genetic materials to create superior offspring. In GA, single-point or two-point crossovers are popular. Mutation introduces diversity into the population. These six steps will be described in detail in this chapter. Genetic algorithms depend on mimicking biological evolution of species. It relies on survival of the fittest.

The fittest are selected to reproduce while the weakest are mostly ignored. In other words, the best solutions survive while bad solutions are left to die. In this section, this model is used to describe the basic terminology of genetic algorithms. Population: The population in a GA is defined as a subset of all possible solutions to the problem in hand. In other words, there is a bigger set 5. Chromosome: A chromosome in the sense of genetic algorithm is one of the possible solutions in the population for the given problem. Chromosomes consist of elements and their positions are called genes. Gene: A gene is an element position within the chromosome. A gene in a chromosome takes a value and the value is called allele. Genotype: This is a population within the computation space. There is a second definition of population in the actual real-world solution space called phenotype. Phenotype: From the above definition, we therefore define a phenotype as the population in the actual real-world solution space.

In practice, a mathematical transformation function is used to provide the connection between the genotype and phenotype. This transformation is called decoding. To model the GA, the chromosomes need to be fully and efficiently determined to ensure that during each generation, the iteration does not descend into a locked position to hinder variation. Each chromosome is described mathematically with binary numbers or bits. Encoding is the transformation of the parameters of the problem in hand into a chromosome. The choice of parameters is a design exercise in the use of GA. Decoding: This is a transformation between the phenotype and genotype spaces.

These GA terms are illustrated in Figure 5. As shown in Figure 5. An individual is a set of chromosomes with genes. Inside each gene is an allele. The table shows a population of 10 distinct entities of different fitness attributes, which we are yet to define. The next section describes how fitness functions are modelled. In genetic algorithms, they are used for ranking of populations as functions of the fitness of population members. In other words, a fitness function assesses how close to optimal solution a given solution is. The fitness function is normally the function to be optimized. Each solution of the GA can therefore be ranked. This leads to selection and the survival of the fitness in the population. Solutions to problems are usually groups of chromosomes, which makes it difficult to rank them without a criterion that is fair on all possible solutions.

A fitness function therefore is used to give a score to each solution and to rank them. How therefore should fitness functions be defined or created? Several criteria have been suggested, which include the following. Thus, it should lead to a clear distinction between the best and worst solutions. It should therefore not pose a bottleneck to solving the problem. It should be able to distinguish between members of the population.

How quickly this is achieved is essential. Example 1: Finding the best three stocks to buy to maximize profit within a set of options. Let the three stocks have yield values x, y and z. The problem in hand is to find the best set of returns to maximize profit p. Therefore, when the loss is very small, the inverse of the loss should be very high. A group of students were chosen at random to represent a University at a technical competition. The selectors are not yet sure if they have made the right choice and wants to run an algorithm to pick the best set of students to represent the university. The selectors thought it out that they will base their choice on how they performed in 10 subjects. Since there are many students who satisfy the criteria, it decided to use GA for the selection of the best fit students. They chose 6 initial students as the starting population.

Whenever a student passes a subject, a 1 is recorded, and whenever the student fails a subject, a zero is recorded. Students with the highest sum of ones are deemed to be the fittest initially. This is irrespective of the subject. Are they correct? They chose to use GA to maximize their decision. This is shown in Figure 5. The normalized input is used for ranking the population. Typically, Equation 5. Having a high fitness is a sign of better fitness and low fitness is a less desirable attribute for selection of a chromosome or member of the population. A good fitness function should be able to classify a population efficiently. Selection is based on the fitness function defined by the
A random selection may also be used, but the method could take a longer time to go through the generations. Usually it is prudent to also include the less fit solutions in the selection. This introduces diversity into the population and prevents premature convergence of the algorithm. Two selection methods are popular and include the roulette wheel and tournament selection methods. Individuals in the population are assigned probabilities proportional to their fitness, as a ratio of the total fitness of the population. Based on the probabilities, two individuals are chosen in a random manner by spinning the roulette wheel and where it lands, the individual at the stop is chosen. The roulette wheel is given a second spin, and where it lands, a second person is chosen to reproduce with the first chosen individual.

At this stage, chromosomes from two parents are shared through the process of crossover. Crossover operations are described in this section first with expressions and then an example. Crossover may occur at only one position or many positions in the chromosome. Each of the parts are transferred to each other as in Figure 5. The most common type of crossover is a single crossover, a common point within the chromosome.

Some of the alleles from two parents are crossed over to create new offspring. In the binary example, consider the chromosomes given in Figure 5. The mates exchange alleles starting from given positions. Figure 5. In each case, equal sets of alleles are used in the crossovers to create two new children, Child 1 and Child 2, respectively. Mutation is necessary to ensure that there is diversity in the population. Mutation is the process whereby a chromosome undergoes a change in which one or more genes is replaced with new gene.

This is illustrated in the next equation for a single-position mutation. The chromosome in the given example has one of its genes replaced with q. In the equation, the gene at position i is replaced with a gene q. Their roles are however to produce new chromosomes. The new chromosomes are created to avoid the process descending to local minima without using fitness functions. The genes involved in mutation and inversions are chosen using very small probabilities.

Therefore, algorithmically, the basic structure of a genetic algorithm is presented in the following flowchart Figure 5. Each iteration in the figure produces a new generation. New generations are affected by crossover and mutation. Each generation has at least a highly fit member of the generation. There remains the problem of which criteria to use in terminating the iterations. In the rest of the chapter, the concepts developed in the previous sections are applied to different problems. Each problem reveals a method of solution and choice of fitness function and crossover. The variable x lies between 0 and 31 as shown in Figure 5. To encode all values of the variable, five-bit chromosomes are required. This is because there are 32 values represented as binary numbers. The range of numbers is therefore from to Since x has a limit 10 placed on it, it is reasonable to pick 10 numbers randomly out of the 32 as the initial population.

The cumulative sum of the fitness function x is The minimum and maximum values are 8. This is shown in Table 5. With 10 chromosomes and pairs to mate, five mating groups are shown. The new population is derived from the parent population by mating pairs of chromosomes. This is followed with mutation of two offspring. The choices demarcate where crossovers have taken place and the red bits where Table 5. Using a fairly high mutation probability of 0. The maximum fitness of the new population is The average fitness for the population is Two chromosomes have mutations, and the positions of the mutations are marked with red bits. The bits are flipped in value from zero to one. Compared to their parents, this population has higher fitness sum and bigger average fitness. The above algorithm is repeated until there is a stopping point. The stopping point could be that there is no more significant changes in fitness of new generations and then stopped.

All the previous examples given so far relate to integers and indeed binary numbers. In this section, we provide examples on how to use genetic algorithms when the data is floating point. For such cases, the chromosome is an array of floating point numbers instead of integers. Therefore, the precision of the solution is only a function of the computing device not the algorithm. The dimensionality of the problem defines the size of the array. In the next sections, we provide two examples in the use of continuous genetic algorithms following the examples in [2]. The first example is to find the point of lowest elevation in a topographical map. The second example is to find the temperature distribution in an agricultural field as given by temperature and location sensors. The fitness function x, y is clearly the fitness function. The system chromosome needs to be defined. The chromosome contains two variables x and y defined by the longitude and latitude. Since the variables are real numbers, there is no clear size of the chromosome. One possible choice is to use all the discrete values of the longitude and latitude, which means the size of the chromosome could be up to This is too large a size for the chromosome. In [2], the size is limited to 12 and we use the same here. The mutation rate is 0. The speed of convergence of the algorithm depends on these values. Tables 5. The first table has a population of 12 and the second 6. The second table is a selection from Table 5. In Table 5. Note that the fitness values for the populations have negative numbers.

This therefore precludes the use of probabilities. Probabilities depend on summations of fitness to define them. How should the fitness function be defined? Table 5. This ranking leads to the following probability function for the n’th chromosome. The variable Nkeep Table 5. It was easy to implement crossover from bit strings. Unfortunately, in this example, we do not have bit strings anymore. How should we implement crossover in continuous genetic algorithm? The approach adopted was initially given by Haupt [3]. From Equation 5. In general, mutation is undertaken also using many different methods. We have shown this example for one iteration. Normally, repeat the procedure many times to find the new generations. Observe that continuous genetic algorithm does not need the decoding step and hence it is a lot faster to implement.

Other conditions soil conditions recorded with sensors include soil water content, nitrogen, potassium and sodium chloride. In this section, we demonstrate how to use genetic algorithm for predicting temperature distribution in a farm. In other words, g is an organism temperature and position set. By converting the vector to binary sets, we obtain the system chromosomes or DNA. Five values therefore need to be encoded for each value of temperature at each location. The fitness function is related to the summation expression above. The objective is to minimize the error between References 91 measured values at a location referenced to a temperature value Ti. In general, the genetic algorithm becomes useful when the search space is large with a large number of parameters.
This situation occurs in Big Data analysis. Genetic algorithm is applicable over integer and floating-point values, which is a huge benefit. Generally, good solutions are found using genetic algorithms, and it is faster and efficient as well. The mathematics is simple and applies to real-world problems. Thus, the field of application is diverse. References [1] David E. Haupt and S. One of the techniques for achieving this is the so-called computational graph. Computational graphs divide down a complex computation into small and executable steps which could be performed quickly with pencil and paper and better still with computers. In most cases, loops that require repeating of the same algorithm but wastes time computationally due to processing of loop times become a lot easier to handle.

Computational graphs ease the training of neural networks with gradient descent algorithm making them many times faster than traditional implementation of neural networks. Computational graphs have also found applications in weather forecasting by reducing the associated computation time. Its strength is fast computation of derivatives. In many ways, computational graph theory is similar with logic gate operations in digital circuits where dedicated logic operations are undertaken with logic gates such as AND, OR, NOR and NAND operations in the implementation of many binary operations. While the use of logic gates lead to complex systems such as multiplexers, adders, multipliers and more complex digital circuits, computational graphs have found their way into deep learning operations involving derivatives of real numbers, additions, scaling and multiplications of real numbers by simplifying the operations.

They also make it easier to track computations and to understand where solutions break down. A computational graph is a connection of links and nodes at which operations take place. Nodes represent variables and links are functions and operations. The range of operations include addition, multiplication, subtraction, exponentiation and a lot more operations herein not mentioned. Consider Figure 6. The variable c is the result of the operation of the function f on a and b. Consider the following nesting of operations with this computational graph in Equation 6. This is followed by the second operation, which is g. However, from the point of view of Equation 6. The arrows originate from the terms used to build the unit term where the arrow ends as in Figure 6. This form of abstraction is extremely useful in building neural networks and deep learning frameworks. In fact, they are also useful in programming of expressions for example in operations involving parallel support vector machines PSVM.

In Figure 6. The evaluation of the compound expression is Figure 6. For derivatives, it simplifies the use of the chain rule. The partial computational graph covering Equation 6. The partial derivative is given in Equation 6. From Equation 6. Two cases are of interest the linear case Figure 6. These two cases are illustrated in this section. The output depends recursively on both y and respect to the input x or dx and hence it is expected that the partial derivative of z will also depend on these two variables. The partial derivative of y with respect to x therefore needs to address this. This is shown in Figure 6. Each loop is treated with a linear chain rule. Consider the following loop diagrams Figure 6. The objective is to find the derivative of the output z using the linear chain rule along the two arms of the loop and sum them.

In the lower 6. Therefore, there will also be a sum of partial derivatives coming from the two branches. Therefore, the total derivative of z to the input is a chain of N partial derivatives. The general partial derivative expression shown as Figure 6. This area has two dimensions: the discrete step in sampling of the function multiplied by the amplitude of the function at that discrete step. N is the number of discrete sampling steps. This sets the bound on the error in the integration value obtained by using the trapezoidal rule for the function. Thus, once the choice of N is made, an error bound has been set for the result of the integration. This error may be reduced by changing the value of N, the number of terms in the summation.

The summation expression is different. The number of terms N is even. This sets the bound on the error in the integration value obtained by using the Simpson rule for the function. Once the choice of N is made, an error bound has been Calculus on Computational Graphs set for the result of the integration. Exercise: Draw the computational graph for the Simpson Rule for integrating a function. Results from previous steps affect the derivative of the current node. Take for example in Figure 6. In the reverse path, the derivative of Z affects the derivative of Y. Let us look at these two cases involving multipath differentiation. Multipath Forward Differentiation In the discussion, we limit the number of paths to three, but with the understanding that the number of paths is limited and depends on the application. Observe the dependence of the partial integrals on the weights from the integrals from the previous node.

Notice the starting point has the derivative of a variable X to X. Y X Figure 6. Multipath Reverse Differentiation In the reverse backward path dependence, Figure 6. SVM combines the best features of linear classification methods with optimization techniques to identify which data belongs to one class or the other. This form of supervised learning algorithm was introduced by Vladimir Vapnik in Since introduction, it has become more and more popular and ranks in line with other learning algorithms like neural networks. Comparatively, it is a lot easier to train an SVM than neural networks. It also does not possess local minima unlike in NN where in gradient descent local minima could result and impact on the convergence performance of NN.

For these reasons and others, it has found application in hand-writing digit recognition. Although NN may be used for digit recognition as well, it requires elaborate algorithmic overhead. Other areas where SVM has been widely applied include data mining, hand-writing recognition, bioinformatics, proteomics, medicine, image classification and biosequence analysis. Classification of data using SVM requires two stages of operation. The first stage is learning.

During this stage, labelled data is analyzed to learn a mapping from x to y, where x is the set of data and y the class set. The aim of this stage is to build a classifier. The second stage is the prediction stage using the classifier obtained from the first stage to predict which class the inputs belong to. To determine the classifier or model, which is a convex optimization problem, a local minimum is sought. The spread in data in an R2 space leads to two popular methods of analysis, the linearly separable case and the nonlinear case. Of these, the linearly separable case is the easiest and usually SVM analysis using this method is rare. For the linearly separable case, classification of data pairs is almost perfect with little or no errors. During the training stage, the classifier learns the parameters of the system, which are w and b. Once the SVM has learned the weights and b from the training points, they are ready to be used to produce outputs corresponding to unknown inputs.

These outputs indicate which class the input data belongs to. This is based on the decision function. This is basically saying that we want to find the best or optimal weights for classifying all the data in the data sets. When we have a parallel SVM, each parallel set needs to find its own optimal weights as well. This chapter is only on the traditional single SVM. Its expressions will be derived in this chapter. In the tutorial-type
discussions in this chapter, we cover the basis of SVM and linear separable types of support vector algorithms and nonlinear types.

Kernel algorithms are included with various kernel functions discussed. Some applications of support vector machines are discussed towards the end of the chapter. This chapter also uses vector theory heavily. Therefore, it is advised that a reader not familiar with vectors take time to first study vectors and then to apply vector calculus. Specifically, vector concepts including length of vectors, vector projections, scalars, dot products, cross products and norms should be reviewed. A patient in a clinic is given attributes that define the patient. The attributes are related to the type of symptoms to an illness the patient has. Please note: BCIT does not provide technical support for student hardware or operating systems. Labs and exercises show how to use SSRS to extract the data from its multiple collections of applications and data sources, how to deliver and manage reports, to integrate SSRS reports.

Participants explore business intelligence platforms and apply best practices to author, deploy and manage reports with SSRS. Participation online via the BCIT Learning Hub and attendance are mandatory during scheduled class hours plus online activities for a minimum of 4 hours per week. Labs and exercises show how to use SSAS to design, create and manage multidimensional structures containing data from other sources. Participants are shown how to create and maintain an Analysis Services database. Upon successful completion of COMP students will be able to use SSAS to deploy an Analysis Services database with multiple levels of security for data mining, they will be able to extract data from collections of multiple data sources and applications.

This online course will provide a comprehensive introduction to the range of accountability and performance measurements approaches used in the Canadian health sector. This course will include the integration of leadership ethics. The course will focus on the challenge of providing health services based on unlimited demand but the harsh reality of a fixed capacity of resource. The LEADS Framework will inform the course delivery and learners will be provided with a solid understanding of health leadership application of performance measurement, performance management and governance issues that emerging leaders need to know and apply.

This online course will provide a comprehensive introduction to evidence-based decision making in the health sector. The course will examine current practices and protocols, standards of evidence, sources of data, and the application of new knowledge to practice in order to foster change through health sector initiatives and projects. The course will include reference to the underlying ethics of decision making evidence. Critical thinking and analysis activities will be applied throughout the course and the LEADS Framework will inform the course delivery. Do you want to know if they transfer to courses here at BCIT? Students must provide a current model PC desktop or laptop with a webcam, microphone and have high-speed internet access. COMP courses may use two-way audio and video as well as group work outside of class. Participants must be computer literate in order to participate and complete each COMP course.

Please contact the department via email: cstpts.bcit. ADAC may be completed with as few as 17 courses, however delivery was designed for courses per term. Due to prerequisites and scheduling it is typically not possible to complete this part-time program in any less than 5 terms, at night and on weekends. Note: Part-time Studies was not designed for full-time delivery or for those on funding timelines. ADAC includes group work, written communications of technical system specifications and oral presentations in a team environment. This program must be completed within 5 years. Students who only complete one course per term will take more than 5 years to finish this certificate. Those students who complete two 2 courses per term may typically complete the ADAC program in around 3 years.

Please contact the department for course planning: cstpts.bcit. Note: Not all courses are offered every term and each course has specific prerequisites. These exams may be purchased for additional costs, from various third parties. Students will also require additional preparation and studying in order to pass these industry certification exams. Students who only complete one course per term will take 5 years to finish this certificate. Those students who complete two courses per term may complete the program in less than 3 years.

Students approved for a part-time studies program are expected to register and complete courses on a continuous basis. A student that has not completed a course from their approved program of study over a period of three consecutive academic terms will be considered to have abandoned the program. Throughout the duration of the program, a student is permitted a maximum of five academic terms where they are not required to register and complete a course from their approved program of study.

Students who exceed the five term maximum will be considered to have abandoned the program. Students who have abandoned their approved program of study and wish to be reinstated will be required to apply for program approval for a current credential. BCIT cannot guarantee that courses taken prior to this reapplication will be credited towards the current credential. The BCIT student outcomes report presents summary findings from the annual survey of former students administered by BC Stats one to two years after graduation.

These reports combine the last three years of available results for the BCIT Outcomes Surveys of graduates and for Degree graduates. More detailed information can be accessed at the BC Student Outcomes website. To view these results, you may need to have the Adobe Acrobat Reader installed in your Web browser. In order to qualify, students must attend an online Computing PTS program information session, and agree to follow a specific term by term course plan for ADAC, as provided by the Computing Department.

Priority approval for limited seats will be based on the timestamp of your email to: cstpts.bcit. Students who require a minimum course load to retain eligibility for financial aid or international student status have no guarantees that all or any Part-time Studies PTS courses will be available in any given term. Please understand there are no guarantees that all or any Part-time Studies PTS courses will be available in any given term. There is also no guarantee that the minimum required course credits you require can be sustained from term to term in PTS. Most PTS courses fill quickly so space is not guaranteed and courses may be cancelled before term start if there is insufficient interest. This may impact both the number of courses a student is able to take in any term, and the time it takes to complete a PTS credential. As a result, students may not be able to obtain the minimum credits they need or want in each term to continue in the program. ADAC was not designed for those attempting to complete this certificate program in a full-time delivery model of less than 2 years or 6x PTS terms.
Most Computing Part-time Studies credentials cannot be fast-tracked in less than 2 years due to prerequisites and scheduling. Please make allowances for this 2 year minimum time requirement in your funding or student visa applications. Before you fill out the form, check the information in all the pages for this program. Declare Now Contact Us. Fall delivery mode: varies by course Please check the individual COMP course schedule pages for delivery and location details. Other courses will be physically face to face and may include some blended online activities. Program declaration Declaring your Part-time Studies program ensures that BCIT is aware of your intent to complete a program as it is currently outlined and provides you the opportunity to apply for transfer credit. To submit your declaration: Answer all questions completely. If required, convert transcripts and documents to PDF files. Have a credit card ready to pay the application fee. Declare Now Upon approval, a program plan letter will be sent to you confirming your program of study.

Some summer sections may also be available. Students should have a personal computer with internet access capable of running MS Windows. There are typically 16 courses required to complete the CSC. This short-term program declaration suspension will provide better course availability for current declared students. Courses Class hours Part-time studies delivery on evenings and weekends. Program matrix Check current availability of courses for this program. Tier 1 Required Courses Program Details Computer hardware and Internet requirements Students must provide a current model PC desktop or laptop with a webcam, microphone and have high-speed internet access. Important Notice: Computing Part-time Studies is course by course registration and was not designed for students on limited timelines. Students on funding timelines should instead apply to Full-time programs. Please read the details on the program entry page.

Program length This program must be completed within 5 years. NOTE: Not all courses are offered every term and each course has specific prerequisites. Program abandonment Students approved for a part-time studies program are expected to register and complete courses on a continuous basis. Graduate employment outcomes The BCIT student outcomes report presents summary findings from the annual survey of former students administered by BC Stats one to two years after graduation.

Applied Data Analytics. Allow approximately six to eight weeks for processing. All financial obligations to the Institute must be met prior to issuance of any credential. Administrative support is provided by Joanne Atha. Students on Limited Timelines Important notice for International students.

~[DOWNLOAD_EBOOK]~ Applied Data Analytics Principles and Application…

Analysis of x-ray images in medical applications, cyber security data, crime data, telecommunications and stock market data, health records and business analytics data are but a few areas of interest. Applications and platforms including R, RapidMiner and Weka provide the basis for analysis, often used by practitioners who pay little to no attention to the underlying mathematics and processes impacting the data. This often leads to an inability to explain results or correct mistakes, or to spot errors. Applied Data Analytics - Principles and Applications seeks to bridge this missing gap by providing some of the most sought after techniques in big data analytics. Establishing strong foundations in these topics provides practical ease when big data analyses are undertaken using the widely available open source and commercially orientated computation platforms, languages and visualisation systems.

The book, when combined with such platforms, provides a complete set of tools required to handle big data and can lead to fast implementations and applications. The book contains a mixture of machine learning foundations, deep learning, artificial intelligence, statistics and evolutionary learning mathematics written from the usage point of view with rich explanations on what the concepts mean. The author has thus avoided the complexities often associated with these concepts when found in research papers. The tutorial nature of the book and the applications provided are some of the reasons why the book is suitable for undergraduate, postgraduate and big data analytics enthusiasts. This text should ease the fear of mathematics often associated with practical data analytics and support rapid applications in artificial intelligence, environmental sensor data modelling and analysis, health informatics, business data analytics, data from Internet of Things and deep learning applications.

Business Analytics, or BA, is neither a product nor a system. Business Analytics refers to a dynamically evolving strategy, vision, architecture, technologies, applications, processes and practices for the collection, integration, analysis and presentation of data with analytics to generate information and knowledge for efficient and effective evidence base management. Setting up a business intelligence program with analytics takes more than just installing the technology. A successful BA program involves a set of concepts and methods designed to make informed business decisions that execute corporate strategy, improve performance and ultimately produce the best possible results by putting targeted information into the hands of those who need it most and empowering people, at whatever level they occupy, from strategic to tactical and then operational.

Email Address. Facebook Twitter Instagram. Applied Data Analytics for Beginners.

Learn how to leverage data and analytics for insights and decision-making. Duration weeks. Mode Online, Zoom meetings. Level Beginner. Turn your knowledge into results! Recommended Courses for This Specialization. Beginner R1. Learn more Specific Objectives After completing this module, participants should: Learn to work with R using RStudio. Learn to install packages and work will libraries in R. Learn the basic of programming in R. Learn tidyverse approaches to data manipulation. Beginner R2. Specific Objectives After completing this module, participants should: Learn to summarize aggregate and reshape data. Learn to join data frames tables. Learn to create visualizations graphs. Key Topics Visualization with ggplot2. Summarizing and joining with dplyr. Reshaping with tidyr. Beginner Python 1. Specific Objectives After completing this module, participants should: Learn the how to code in Python Work with key python data structures. Learn to code functions in Python Work with a python library.

Beginner Python 2. Specific Objectives After completing this module, participants should: Learn which are fundamental python libraries. Learn to code control flow and set up logical statements. Learn to use graphlab create — a machine learning environment. Beginner Math, Probability and Statistics 1. Specific Objectives After completing this module, participants should: Prepare and interpret visual data representations. Define and interpret different data summarization techniques. Prepare and interpret a data summarization report in R. Prepare and interpret data visualization in R. Beginner Math, Probability and Statistics 2. This course presents elementary topics in data analysis through regression. Specific Objectives
After completing this module, participants should: Explain the difference between different types of probability distributions. Calculate and interpret the confidence interval of different data types. Create and interpret different types of hypothesis tests. Prepare data for OLS regression analysis in R. Describe the different assumptions of OLS regression. Perform a simple OLS regression in R. Interpret and present the results of simple OLS regression.

Beginner Business Foundations. Specific Objectives After completing this module, participants should: Be able to understand the basic economic theories of supply and demand and their applications to business. Describe different types of marketing theories and their translation to strategy. Understand the six fundamental principles of finance. Understand the purpose for accounting laws. Be able to interpret and use basic accounting tools such as income statements and balance sheets. Describe the role of management in business.

Key Topics Service oriented enterprise, architecture, Information technology, Management information systems, Cloud computing. Specific Objectives After completing this module, participants should: Understand the fundamentals of transactional OLTP and decision support systems OLAP. Learn the relationship between data, information, and knowledge. Learn the fundamentals of data and database management. Learn how to create a database, and how to access it. Learn how database, data warehousing, data lakes, big data, and business intelligence are connected, and how they are used to support smart decision making. Learn the fundamentals of Cloud Technology and cloud services used for Data Management.

Beginner Data Modeling. Specific Objectives After completing this module, participants should: Understand what is data modeling. Create a conceptual, logical and physical database model. Understand fundamentals of relational vs.

Beginner SQL 1. This course will cover structured query language SQL. Specific Objectives After completing this module, participants should: Learn fundamentals of structured query language. Learn how to obtain information from a database with SQL. Update database content with SQL. and transaction handling. Retrieve data with filter conditions and from multiple tables using various types of join. Beginner Excel for Data Analytics 1. Specific Objectives After completing this module, participants should: Utilize data visualization principles and avoid common mistakes. Be aware of different graph types and identify the most effective one given a business context. Create and publish Power BI dashboards. Customize dashboards in Power BI. Sample list of software that will be used: Power BI, Microsoft Excel.

Art And Engagement
Eyewitness DVD: Dinosaur
Your Dinners Poured Out: Memoirs of a Dublin that has Disappeared
The Uses of Art in Public Space
Nginx HTTP Server - Third Edition
The Year in Television, 2009