DOWNLOAD PDF GENTLE INTRODUCTION TO THE VAX SYSTEM

Chapter 1 : "A Gentle Introduction to the VAX System" by John R. Hubbard

Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

Items with a larger number of purchases by a customer can carry more weight in our ratings matrix of purchases. Our last step is to create the sparse ratings matrix of users and items utilizing the code below: In terms of sparsity of the matrix, that makes: For collaborative filtering to work, the maximum sparsity you could get away with would probably be about We are well below this, so we should be able to get decent results. We do this by creating a test set completely separate from the training set. Usually this is fairly simple: That normally looks like this: Then, check during the test phase how many of the items that were recommended the user actually ended up purchasing in the end. This will be a better test. You can see an example here: Our test set is an exact copy of our original data. We then check in the test set which items were recommended to the user that they ended up actually purchasing. If the users frequently ended up purchasing the items most recommended to them by the system, we can conclude the system seems to be working. As an additional check, we can compare our system to simply recommending the most popular items to every user beating popularity is a bit difficult. This will be our baseline. However, it is a practical way of testing performance I will use for this example. We will also import the random library and set a seed so that you will see the same results as I did. The test set will contain all of the original ratings, while the training set replaces the specified percentage of them with a zero in the original ratings matrix. Test is just a complete copy of the original set. This will be necessary later when evaluating the performance via AUC. We will test the performance of the recommender system on these users only. Implementing ALS for Implicit Feedback Now that we have our training and test sets finished, we can move on to implementing the algorithm. If you look at the paper previously linked above Hu, Koren, and Volinsky you can see the key equations will we need to implement into the algorithm. We need to turn this into a confidence matrix from page 4: The paper suggests 40 as a good starting point. We can derive a similar equation for our items: Now that the math part is out of the way, we can turn this into code! I have altered his to make things easier to understand. Designed for alternating least squares and implicit feedback based collaborative filtering. Should be a sparse csr matrix to save space. Increasing this value may increase bias but decrease variance. The paper found a default of 40 most effective. Decreasing this will decrease the variability in confidence between various ratings. More iterations will allow better convergence at the cost of increased computation. The authors found 10 iterations was sufficient, but more may be required to converge. The paper recommends varying this between Increasing the number of features may overfit but could reduce bias. The feature vectors for users and items. The dot product of these feature vectors should give you the expected "rating" at each point in your original matrix. Makes calculation more simple. We can compute this before iteration starts. T Transpose at the end to make up for not being transposed at the beginning. Y needs to be rank x n. Keep these as separate matrices for scale reasons. Hopefully the comments are enough to see how the code was structured. You want to keep the matrices sparse where possible to avoid memory issues! I will choose 20 latent factors as my rank matrix size along with an alpha of 15 and regularization of 0. This takes about 90 seconds to run on my MacBook Pro. The first user in our matrix has the fifth item with the greatest recommendation out of the first five items. However, notice we only did one iteration because our algorithm was so slow! We could wait 15 minutes to let this run, or. Fortunately, as I was still finishing this up, Ben Frederickson at Flipboard had perfect timing and came out with a version of ALS for Python utilizing Cython and parallelizing the code among threads. You can read his blog post about using it for finding similar music artists using matrix factorization here and his implicit library here. All I can tell you is that it is over times faster than this bare bones pure Python version when I tested it. Install this library before you continue and follow the instructions. If you have conda installed, just do pip install implicit and you should be good to go. First, import his library so we can utilize it for our matrix factorization. I did some testing and found the following settings to work the best. Also make sure that we set the type of our matrix to double for the ALS function to run properly. We now have

recommendations for all of our users and items. However, how do we know if these are any good? This will allow us to evaluate the performance of our recommender system. Essentially, we need to see if the order of recommendations given for each user matches the items they ended up purchasing. A greater area under the curve means we are recommending items that end up being purchased near the top of the list of recommended items. Usually this metric is used in more typical binary classification problems to identify how well a model can predict a positive example vs. It will also work well for our purposes of ranking recommendations. In order to do that, we need to write a function that can calculate a mean area under the curve AUC for any user that had at least one masked item. As a benchmark, we will also calculate what the mean AUC would have been if we had simply recommended the most popular items. Popularity tends to be hard to beat in most recommender system problems, so it makes a good comparison. Scikit-learn has one we can alter a bit. It should also calculate AUC for the most popular items for our users to compare. These should be stored in a list, with user vectors as item zero and item vectors as item one. The mean AUC area under the Receiver Operator Characteristic curve of the test set only on user-item interactions there were originally zero to test ranking ability in addition to the most popular items as a benchmark. The original pure Python version output the user and item vectors into the correct format already. Our system had a mean AUC of 0. You can go back and tune the hyperparameters if you wish to see if you can get a higher AUC score. A Recommendation Example We now have our recommender system trained and have proven it beats the benchmark of popularity. An AUC of 0. First, however, we need to find a way of retrieving the items already purchased by a user in the training set. Initially, we will create an array of our customers and items we made earlier. Looking at the list of customers:

Chapter 2 : Holdings : The VAX book : | York University Libraries

The purpose of this book is to help the novice become comfortable using any of the Digital Equipment Corporations VAX computers, from the Micro-VAX to the powerful VAX system. The book is meant to be used as a tutorial.

Clustering[edit] OpenVMS supports clustering first called VAXcluster and later VMScluster , where multiple systems share disk storage, processing, job queues and print queues, and are connected either by proprietary specialized hardware Cluster Interconnect or an industry-standard LAN usually Ethernet. This version only supported clustering over CI. OpenVMS supports up to 96 nodes in a single cluster, and allows mixed-architecture clusters, where VAX and Alpha systems, or Alpha and Itanium systems can co-exist in a single cluster Various organizations have demonstrated triple-architecture clusters and cluster configurations with up to nodes, but these configurations are not supported by HP. Unlike many other clustering solutions, VMScluster offers transparent and fully distributed read-write with record-level locking , which means that the same disk and even the same file can be accessed by several cluster nodes at once; the locking occurs only at the level of a single record of a file, which would usually be one line of text or a single record in a database. This allows the construction of high-availability multiply redundant database servers. Cluster connections can span upwards of miles, allowing member nodes to be located in different buildings on an office campus, or in different cities. Host-based volume shadowing allows volumes of the same or of different sizes to be shadowed mirrored across multiple controllers and multiple hosts, allowing the construction of disaster-tolerant environments. Full access into the distributed lock manager DLM is available to application programmers, and this allows applications to coordinate arbitrary resources and activities across all cluster nodes. This includes file-level coordination, but the resources and activities and operations that can be coordinated with the DLM are completely arbitrary. With the supported capability of rolling upgrades and multiple system disks, cluster configurations can be maintained on-line and upgraded incrementally. This allows cluster configurations to continue to provide application and data access while a subset of the member nodes are upgraded to newer software versions. The operating system includes a mechanism to adjust for hardware timekeeping drift; when calibrated against a known time standard, it easily achieves an accuracy better than 0. All OpenVMS hardware platforms derive timekeeping from an internal clock not associated with the AC supply power frequency. This clock keeps time to a lower resolution perhaps 1 second and generally, a lower accuracy often 0. When the system is restarted, the VMS bit time value is recomputed based on the time kept by the TOY clock and the last recorded year stored on the system disk. The nanosecond granularity implemented within OpenVMS and the bit absolute time representation the sign bit indicates absolute time when clear and relative time when set should allow OpenVMS trouble-free time computations up to JUL At this instant, all clocks and time-keeping operations in OpenVMS will suddenly fail, since the counter will overflow and start from zero again. Though the native OpenVMS time format can range far into the future, applications based on the C runtime library will likely encounter timekeeping problems beyond January 19, due to the Year problem. Many components and applications may also encounter field-length-related date problems at year see the Year 10, problem. Because of this, it is possible and straightforward to call a routine written in one language Fortran from another COBOL , without needing to know the implementation details of the target language. This provides mixed-language calls, and a set of language-specific, run-time library RTL , and system service routines. The language calls and the RTLs are implemented in user-mode shareable images, while the system services calls are generally part of the operating system, or part of privileged-mode code. This distinction between languages and RTLs and system services was once fairly clean and clear, but the implementations and specifics have become rather more murky over the years. Various utilities and tools are integrated, as are various add-on languages and tools. It allows breakpoints, watchpoints and interactive runtime program debugging either using a command line or graphical user interface. Used interactively, this represents the terminal keyboard. Used interactively, this is the terminal display. Used in a batch file, it outputs to the screen if run interactively or to the log file when run noninteractively. Used interactively, it will read from the terminal. Used in a batch file when run interactively, it will read from the terminal.

Chapter 3 : The OpenVMS Frequently Asked Questions (FAQ)

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

How does this work? This post is intended as a primer about payment systems and explains correspondent banking, nostros, real time gross settlement RTGS systems and deferred net settlement DNS systems. It supports other posts where I discuss decentralisation of these systems using distributed ledgers. What do the customers see? The more you have in your account, the more the bank owes you. So, after this transfer: Bank A now owes Annie less money! So this payment instruction must be balanced by a bank to bank transfer: So how do banks pay each other? How does it work digitally? There are two main ways: In this way, neither bank is better or worse off: How might Bank A fund its account at Bank C in the first place? Perhaps the banks opened accounts with each other at the same time, funded with the same amounts: If you felt like it, you could refer to your bank accounts as your nostros, but you would lose your friends. However, if you were a bank, maintaining accounts at every single other bank that your customers might want to transfer money to would be quite a pain, and expensive they have to have money sitting there doing nothing, in anticipation of payment instructions, and as we all know, current accounts pay very low interest. What if the other bank went bankrupt? There is another wayâ€¦ Central bank payment systems One bank to bank them all. So there is a more efficient way. So the final settlement where the banks pay each other and everyone is happy, is done at the central bank. NB Just to confuse everyone, the word clearing in payments means something different to the word clearing in securities trading. If both customers bank with the same bank, then that bank clears the transaction. If there is a correspondent banking relationship, then the receiving bank clears the transaction. The accounts that each bank holds with the central bank for this purpose is sometimes called their clearing account. In reality banks hold multiple accounts at the central bank, for different purposes. This is much more efficient than maintaining lots of nostros. However, it only works within one country and in one currency. So most advanced countries will have a centrally cleared RTGS or DNS system for clearing interbank payments within that country for the domestic currency. International payments International currency accounts. For international payments of one currency â€” ie not foreign exchange! For example if a Thai banking customer wants to pay GBP to a Singapore banking customer, this will be cleared somewhere in the UK, either by the Bank of England, or, if by chance they use the same correspondent bank, by that correspondent bank: Sometimes smaller banks or banks in poorer countries might not be able to establish banking relationships with big foreign banks, as I have written about. I hope it has been helpful!

Chapter 4 : Introduction to VAX/VMS

Description. This book is an expansion of the book, A Gentle Introduction to the Vax calendrierdelascience.com purpose of the book is to guide the novice, step-by-step, through the initial stages of learning to use the Digital Equipment Corporation's Vax computers, running under the VMS operating system (Version or later).

There are a few different models of VAX, but we will not cover that. VAX is able to run several versions of operating systems. On the machine we discuss, the operating system is VMS. The operating system is what allows you to run commands and software application. No two operating systems have the exact same command set. The terminals will use a direct line to the system. On the PCs or Macs, you would use the Telnet function. You even may log on to the VAX from home. To log on to the computer, you need to input some basic information to access your account. You first must know what your account number is. To find out what your account number is, you can go to the information center on the second floor of the I Building. Against the wall is two terminals that are dedicated to providing the account numbers of students. To use the system just type in your last name and hit the enter key. The display will show your name and what your account number is. Once you have this information, you can go to one of the computer terminals and log on. If you have a problem getting your account number, ask for help at the information center. When you get to a terminal, turn it on. You would respond VMS. Once you have entered that information, you will be asked for your User ID account number. Enter your account number. Once you have entered your account number, the system will prompt you to enter your password. This tells you that VMS is ready for your commands. Change Your Password The first operation that you should perform after you access your account is changing your password. The command to do this is "password". After you enter your command, the system will ask you for your current password. After you enter the password you will then be asked to enter the new password that you wish to have. You can make your password any combination of numbers or letters. You must have a minimum of six characters and a maximum of 31 characters. After entering your new password, the system will then ask you to enter it for a second time to verify that it is correct. After you enter it a second time, the system will either accept it or reject it because it is listed as already being used by someone else. You are required to change your password frequently. As often as 45 days. It is good security to change your password frequently and not to choose obvious passwords. To call it up at the command prompt type in "teach". After typing in the command, you will be prompted to either type in the name of the tutorial you wish to take or a question mark. Typing in the question mark will allow you to view a listing of your choices and then to make a selection. If you wish to take one of the lessons, type in the name of the lesson you desire and hit enter. If you just hit the return key, you will be returned to the command prompt. Here is a listing of the available tutorials for you. It is time well spent when you are a novice on the system. Basic VMS Operations For the most part, the majority of your computer usage will involve text editing and file management operations. You also may perform some software development and Internet functions, but text editing and file management will be most of your basic usage of the VAX. Otherwise you will be using communication aspects of the VAX Our goal is to get you started in the default VAX editor and be able to perform basic file management routines. Other topics will consider communications and other functions. In lieu of that the following basic instructions should be of assistance to you. You can just call the editor by entering the line "edit" at the command prompt. Once that is done, the editor is active and will prompt you for a file name. If you give it an active file name, it will call up the latest revision of that file name VAX will create multiple revisions of files. If you give it the name of a file that does not exist, it will create that file. The other way is to enter the line "edit filename. The editor will either come up in line mode or keypad mode. When edit is initially started it comes up in line mode. Line mode is just that. The editor works on one line at a time. It is visually difficult to work with. You can change to keypad mode by typing in "change" Once you have done that, you are able to work on a full screen seeing what ever of your text will fit. This is visually simpler to work with. When you are creating a file, there is initially a string at the top of the screen "[EOB]". This stands for End Of Buffer. It signifies the end of the file that you are working on. When you start typing in text, the input will appear above this

indicator. To enter text to the buffer, simply type whatever you want. To move to certain sections of your work, use the up, down, left and right arrow keys. To remove what you have typed, set the cursor to the right of the character you wish to delete. When you have gotten to that point, use the delete key to remove the undesired characters. The delete key will remove any character to the left of the cursor. Be aware of what you are doing. You just need to perform several keystrokes to do so. While you are in the keypad mode, the command input line is no longer active. At this point you can enter two commands. To save your work, you would enter the command "exit". This would save the work you just performed and exit the editor. If you were working on a new file, the file will appear in your directory. The files are numbered after the file extension. The higher the number, the more recent the revision. VMS will save the three most recent copies. This is a safety measure, but can be confusing and harmful if you do not pay attention to this. To eliminate older versions of files in your directory, use the "purge" command at the command line prompt. This will discard all older copies of files and save only the most recent revision. This will result in the editor shutting down and your work being discarded. There will be no changes to existing files. If it was a new file that was being created, the creation process will be terminated. There will be no sign of any such file. A simpler editor to use is PICO. The PICO editor though does not create any system problems and works quite well. Just type in "pico filename. We will not supply you with other operating instructions. There is no need for them. PICO is a menu driven editor. On the bottom of your screen, you will see all the operating commands and how to carry them out. Whatever the menu system gives you is all the assistance you are going to get on the editor. The help center in either the pc labs or in the I Building will not give you assistance either. If this is undesirable to you, then do not use the PICO editor. It is the activity of managing your files. When you have an account given to you on the VAX system, you are allotted a block directory for your personal use. A block equates to around one kilobyte of storage space. You in essence have about Kilobytes of storage on the VAX. That is less than what can be stored on a modern floppy disk. DIR will five you a basic listing of what files are in your directory.

Chapter 5 : A Gentle Introduction to GIS

Free Download Gentle Introduction To The Vax Book PDF Keywords Free DownloadGentle Introduction To The Vax Book PDF, read, reading book, free, download, book, ebook, books, ebooks, manual.

Chapter 6 : MicroVAX - Wikipedia

A gentle introduction to interbank payment systems for business people and students. One of the most popular articles on the Bits on Blocks blog. Skip to content.

Chapter 7 : OpenVMS - Wikipedia

FreeDOS is an old operating system, but it is new to many people. In , several developers and I came together to create FreeDOSâ€”a complete, free, DOS-compatible operating system you can use to play classic DOS games, run legacy business software, or develop embedded systems.

Chapter 8 : A Gentle Introduction to Interbank Payment Systems â€“ Bits on Blocks

VAX/VMS Online Tutorials-TEACH. The VAX system has an online tutorial system. To call it up at the command prompt type in "teach". After typing in the command, you will be prompted to either type in the name of the tutorial you wish to take or a question mark.

Chapter 9 : Gentle Introduction to Database Systems - NYU - calendrierdelascience.com

services you would expect from an operating system, including hardware ab- straction, low-level device control, implementation of commonly-used func- tionality, message-passing between processes, and package management.