CPT Internship Report

Siemens PLM Software

Aijia Liu
Computer Science

Pengcheng Liu
Advanced Data Scientist
1. Company Information

Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, system and services with over 15 million licensed seats and more than 140000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter [1].

This summer, I worked as a data scientist intern at Siemens PLM Software Charlotte office. My team is called Cloud Services. The product we are building called Omneo, which is a cloud based, software-as-a-service (SaaS) solution that delivers the product performance intelligence needed to make impactful decisions across a global value chain. By providing greater visibility into both products and supply chain performance, Omneo can easily identify and monitor supplier issues, prevent costly recalls, spot emerging trends, solve quality problems faster, and improve customer’s experience [2].

Our team focus on building next-generation platform integrated with current Omneo workspace and also trying to provide users with an easy-use and strong platform to do different data-related work. As part of the Siemens PLM Software portfolio, Omneo is backed by world class industry experience and trusted partners.

The educational objective of this internship is enhancing my industry experience in Big Data area and also improving my general programming skills. In this internship, my main job duty is evaluating the performance and accuracy of different data mining and machine learning algorithms for solving defined problems and also designing solutions to enable non-technical users to benefit from advanced analytics, and work with development team for implementation.

Among these three months, I work in a team of 6 in an agile environment. I have accomplished several tasks:

1. Sentiment analysis Spark implementation.
2. Workspace out-of-box function implementation.
4. Real-world use case

2. Background about the problem

For Sentiment Analysis, which is a typical use case of natural language processing, text analysis and machine learning. It has been widely applied to public reviews and social network for a variety of applications ranging from marketing to customer service. Through Sentiment Analysis, we may be able to judge the attitude of a speaker or a writer to some topics or the overall contextual polarity of a document [3].

The reason we want to build sentiment analysis is that many enterprises and manufacturers want to have the capability to get the feedbacks from their customers through social network or get good and bad reviews from some public websites. Then, if we can provide this capability, instead of asking data scientists to do the data analysis, they can use Omneo directly.

In this task, I have used Apache Spark framework to develop the application, where Apache Spark is an in-memory computing system to process large volume of data. Previously, there is one ready version which has been implemented in Java. However, it is hard for the program to handle large data set, where the program
would get stuck since the JVM memory cannot hold this large volume of data. Besides, it takes several minutes to complete the training process. Therefore, my job is to rewrite the program using purely Spark API, which enables the whole program can run in parallel and get the capability to handle large data set.

In machine learning area, especially supervised learning, where we need to prepare the training data set to feed in the model. Therefore, if the size of the training data set is too small or the data quality is not good, the model won’t be powerful and the results won’t be accurate. In order to let the model be more powerful, we have to use big data technology, such as Apache Spark, to scale the program and optimize the program for time cost. Then, it will be very convenient and fast for both us and our customers to conduct the experiments. So for now if customers have large data set for the training part, our product can also handle it properly.

After this, my work is to build out-of-box functions for sentiment analysis, which enables users can get an easy-use interface. At the same time, we will also provide data visualization to improve the user experience.

Then, I am also in charge of backend implementation of platform workspace, which is a place for customers to easily perform some query functions on their data. For the computation engine, we apply various techniques to implement all the advanced query functions where each user operation will be treated as a single job and this job will be taken care of a free cluster.

In order to support more advanced transformation, I have used newest Spark Dataframe API and also Spark SQL to process the data in parallel. With the power of different distributed computing techniques, our workspace is able to do big-table joining, filtering, and other complex transformation. In addition, most of the transformations are much faster than querying on traditional relational database like MySQL, Postgresql.

At last, our team have fixed a problem in the real use case, where we have applied different data exploration, data mining and machine learning techniques to look for interesting patterns in the data.

3. Focus of the effort

For the Sentiment Analysis task, since I don’t have much experience on machine learning and natural language processing before, first thing I do is to understand the algorithm and each function in the original code. Therefore, I have read several papers about the sentiment analysis, get the details about the implementation parts, such as data exploration, feature selection and SVM model, etc. Then, I pick up two paper which are most suitable to our case and I use the algorithm and SVM model in Spark MLib to conduct the implementation. Then, since I will rewrite the whole training part code using Spark Java API, the next step is to learn syntax and usage of Spark Java API.

I have gone through many Spark programming guides and also the technical blogs about how to better use Spark API. In order to get some real practice, I plan to rewrite just feature selection code first and then rewrite other parts. Since I have taken CSE 512 Distributed Database System course, which contains a group project using Spark to develop Geo-Spatial operation library, I was able to grab Spark programming skills faster. So I applied what I have learned before and successfully rewrite the feature selection part. However, when I want to test the code, I found the program cannot finish even if the input contains just 10k records. Later on, I found that the revised feature selection code is called in another for loop where this for loop has to be executed on the master node, instead of distributing the job to each work node.

So I got the idea that Spark is for batch processing, where you have one RDD as your input, and you have to apply several transformation and action on it until you get the final result. In addition, you have to replace all
the nested loop with RDD transformation or action, which means any global value has to be integrate with the RDD one by one. This strategy has totally changed the original code logic and I need to redesign a whole data pipeline to get the final results instead of rewriting the original code. Also, when I finished the new Spark program and I try to run it on large data set, which contains 400k records, but it got memory exception. After some research, I was able to resolve this issue by using map-side join where I collect one RDD into Map and broadcast the Map to each worker node, and then I can fetch the key-value pair in this Map efficiently in another RDD transformation.

For the backend part of predictive workbench, I again have to learn some new APIs, DataFrame and Spark SQL. For Spark SQL, it is simple to use since Spark can recognize SQL query directly. But DataFrame, which is equivalent to a table in a relational database or a data frame in R/Python, is harder to write the code and also you have to make sure that Dataframe interacts with Spark SQL properly. In addition, since I am using Java for this task, I cannot debug the code until I finish the coding and run the whole program. Later on, after learned many other use cases and API docs, I was able to finish filter, join, ranking functions, etc. Every function I was using pure DataFrame API, which is much faster than SQL query when you are querying big table. Also in DataFrame API, each row can have the context of several preceding rows and also following rows, so I was able to apply window function and ranking function easily on the data set.

4. Results

The results are very good. My mentor and manager both are very delighted to see me contribute things to the team and the company, and for myself, I enjoyed the teamwork and working environment.

For sentiment analysis part, I was able to rewrite the training model part using Spark API, which has optimized the performance up to several times faster and the program can handle more than one million records successfully and finished in a short time, which is very efficient. In addition, I also have referred some paper and done some research on both sentiment analysis and machine learning area. After my modification on feature selection algorithm and text mining part, the model accuracy has gotten some improvement. Then for out-of-box function, I have provided both training model and data analysis functions, which enable users to build their model based on the input data and also to conduct analytics on their data. This sentiment analysis application will let the users be able to collect their review data and conduct different kinds of analysis in our workspace where they can gain some important information about the feedbacks and also detect some potential problems in their products in advance.

In addition, I have also developed several out-of-box function for workspace. With the Spark DataFrame API, I was able to implement different kinds of advanced queries. For example, I can partition by one column, order by one column, and then ranking another column or I can also generate window ID for different partition and join the datasets which are fall into same window. Those functions are very useful in case someone needs to sanitize their data in various ways to get the final results. Then, considered that customers may have a long-lineage operation pipeline, I have also tried to come up with some design ideas which can make the most use of the memory since it will be much faster if we can keep intermediate data in memory for future use. As for many other transformation functions, many of them can be applied to real-world use cases.

At last, we have gotten a real-world project where we need to find patterns in some complex data. With the leadership and instruction from my mentor, I was able to understand most of the parameters in the data, and then I have communicated with team members to conduct the plan on data exploration. I have utilized DataFrame to do all the query, which is much faster and flexible than using SQL. I have joined different tables with important columns and then I have conducted various statistics analysis based on different keys and
5. Learning experience

This internship really has brought me much knowledge and insights, which is a very good learning experience. After I have taken CSE 512 Distributed Database System class, I began my journey on Big Data, where I found that many areas are facing big data issue and trying to find a solution. In that course, I have joined a group project about a popular big data framework, called Apache Spark. Then, during that class, I got some basic knowledge about the Spark framework and infrastructure, but I haven’t done enough Spark programming. However, in this internship, I got the chance to work on Spark again, where I need to not only use Spark to do some performance testing, but also a lot of programming about the backend part of the Spark applications. Those programming practice brought me many fresh ideas and design solutions about a complex data pipeline, where I also got a deeper understanding about the Spark and distributed computing.

Then I have built several out-of-box functions about machine learning and Spark DataFrame where I worked in backend part and other team members are in charge of UI and Web services. This teamwork has strongly made me feel excited and motivated. Everybody in the team is friendly and willing to help you fix any problem. During the development process, I learned a lot about software integration and also Web Services knowledge where we worked closely and updated our own progress to the team in an efficient way.

For my last task, I have gained a lot of insights from this real-world problem. Since previously most of my work are related to software development, but this task is a pure data science task where I have to utilize different statistics analysis, data visualization and machine learning to help get some potential intelligence behind the data. During this task, we had meeting every day to share our ideas and understanding of the data and discussed the next step, which is a very good learning experience from teamwork.

Overall, this internship really lets me learn a lot and gain much experience and knowledge about data science and big data area. It also has enhanced my communication skills and professional experience about how to play a good role in a team. I really enjoy this experience. My mentor, who is a good leader to our team, has given me a lot of instructions and support, which enables me to complete the tasks well. In the future, I will keep my passion on big data and learning more.

6. Reference