LEARNING TO RANK, STABILISATION PROJECT

March 27, 2018

CHAPTER

ONE

ABOUT ME

- **Name**: Aditya Kumar
- **E-mail address:** f20150175@hyderabad.bits-pilani.ac.in , kumaradityagr@gmail.com
- **IRC nickname(s):** addy ,addy_
- **Any personal websites, blogs, social media, etc:**

  Facebook: https://www.facebook.com/AddyK7
- **github URL:** https://github.com/addy007-icy
- **Biography:**

  I'm a junior undergraduate majoring in Computer Science at BITS Pilani, Hyderabad Campus in India. My nurtured interest in programming simple algorithmic problems made way to a pursual of library and application development to make lives easier in the last couple of years. I have had formal programming exposure for more than 6 years. Machine learning, Information retrieval and related topics intrigue me the most. I find the use of mathematical algorithms to find patterns in data to solve various problems is exhilarating.

  As a member of the Computer Science Association and Programming club of my University, I have contributed to numerous algorithmic contests, talks and awareness programs including Machine Learning, Open Source and Algorithmic Coding. I have represented my college in ACM-ICPC India regionals twice in a row and my team secured an under 20 rank in both Amritapuri and IIT-Kharagpur regionals this year. I enjoy playing games that require strategising and I enjoy Chess, CS:GO and DotA 2 a lot. Watching TV shows and Stand Up Comedies are my other interests.

Background Information

Have you taken part in GSoC and/or GCI (https://codein.withgoogle.com/) and/or similar programmes before? If so, tell us about how it went, and any areas you would have liked more help with.

No.

Please tell us about any previous experience you have with Xapian, or other systems for indexed text search.

None.

Do you have previous experience with Free Software and Open Source other than Xapian?

Yes. I have been a part of "Crux Winter of Code" , our programming club's initiative to improve open-source culture at our college, and as a part of the "OpenSource Awareness" program. I was a co-mentor in a fest-manager project that was offered in the same.

What other relevant prior experience do you have (courses taken at college, hobbies, holiday jobs, etc)?

University level Information retrieval and Machine learning courses are the most significant courses that I have completed with respect to my project. I am enrolled in the advanced Machine Learning

course on Neural Networks for the spring semester of 2018. I also have read the various research papers provided in the project page of xapian-letor and have an understanding of its working.

What development platforms, tools and methods do you prefer to use?

I use Ubuntu 16.04 LTS and git and Sublime Text 3.

Have you previously worked on a project of a similar scope? If so, tell us about it.

Yes, I have worked on a couple of projects as a part of my Information Retrieval course, one of them was making a search engine on documents based on tf-idf ranking scheme. In C++, I have coded a neural network for face detection, as well as a Decision tree algorithm with random forest and ADA boosting, to predict how much a person would be earning based on various factors like location, age etc. I also have implemented a geometry library, hence am familiar with scalable programming.

What timezone will you be in during the coding period?

My timezone would be IST (Indian standard time) i.e +5:30 UTC.

Will your Summer of Code project be the main focus of your time during the program?

Yes. With no other work scheduled this summer, Summer of Code would be my main focus.

Expected work hours (e.g. Monday–Friday 9am–5pm UTC)

Preferably from Monday - Friday 12pm IST to 9pm IST (6:30 AM UTC - 3:30 PM UTC).

Are you applying for other projects in GSoC 2018? If so, with which organisation(s)?

No.

CHAPTER

TWO

MY PROJECT

Motivations

Why have you chosen this particular project?

My interest in Information Retrieval and Machine Learning, this project caught my eye. This project would not only help me in my career, but also would help me delve more into my interests.

Who will benefit from your project and in what ways?

Xapian-letor needs to be tested and improved upon, as currently it's in its release phase. Once my project is releasable, it will not only be stable enough to be merged, it will also improve the correctness of search results that are retrieved, thus improving the overall precision of xapian-letor. Moreover, it would help me, as a programmer, to be able to write highly efficient scalable code and get valuable experience in how an open-source organisation works

Project Details

Describe any existing work and concepts on which your project is based.

The project is in two parts, first of which is to stress test and benchmark the xapian-letor, on various datasets, and to report and solve any bugs or edge cases that might be encountered. Moreover my goal will also be to benchmark Xapian-Letor on the INEX and FIRE datasets vs the BM25 and report its performance.

The first part of project is based on testing and stabilising the letor project. The current state of the xapian-letor is that it has not been benchmarked nor stress tested, on various data sets. Benchmarking is planned to be done against the INEX 2009 dataset, where we have a tool "2009 Assessments and 2009 inex_eval evaluation tool" to test our retrieval. Standard results of BM25 are also given under a segment. The other dataset that I plan to use is the "FIRE" dataset. It is accessible through one of my mentors, which will also be used to benchmark our data, as well as to test it.

Our model also has to be checked for stability against various datasets. Depending on the behaviour of various datasets, our model might misbehave. So stress testing along with handwriting specific edge - test cases will ensure that xapian-letor is stable and can be merged into the main directory.

The second part is working on multiple improvements,

- The first improvement to be made is to combine multiple rankers by linear regression using a feed forward, back propagation algorithm giving varying weights to each model, which would be balanced by the algorithm, to give a good possible outcome for the data. This ensemble model would be then benchmarked to show what difference it brings to xapian-letor. The basic idea is to let the backpropagation algorithm decide what weights we would like to give each of the model, so that the overall combination of multiple models, is better than the individual ones, as flaws and fits by each of the model are adjusted even further to provide a better ranked retrieval. The ensemble model is generally better than a single model as in the worst case it takes the best model, and makes its weight maximum possible, taking the other weights as zero Hence this model, is at least always better or equal to the best model that we currently have.

The feedforward algorithm is an algorithm which takes a decision for a node based on the input features along with the weights assigned to each of the connection between the two. The backpropagation algorithm is the magical concept by which this works which states the rules of the error being propagated back to the input layer from the outermost layer As this update happens, weights for each connection are updated by a certain value depending on it's previous weight and a learning rate hyperparameter.

I plan to implement this by making a new ranker file, which in turn will call rankers in the ranker file; as requested and will run the feed-forward backpropagation algorithm.

- The second improvement to be made is to implement Principal Component analysis which is used for dimensionality reduction. Currently we're taking 19 features for each document-query pair; our aim is to reduce the number of dimensions by one if it is possible to represent one feature as a linear combination of the others, making it a tradeoff against time complexity for a marginal difference in precision.

In PCA we represent our input data matrix after decomposing the data into a 'p' dimensional space, where we have 'p' orthogonal vectors, called eigenvectors, along with the weight of each vector denoting it's importance. (More formally, it denotes eigenvalues for each vector.) In a case where two features overlap, an eigenvector with a very low eigenvalue will be obtained denoting that this feature was already implemented as some other vector. So, that feature can be considered as not-very relevant for our dataset, and we can reduce our features to a p-1 dimensionality plane, where required running time is less than the previous higher dimensional plane.

I plan to implement this by making a new "Reduction" folder, where I will take the feature vector, and the data vector as inputs, and apply a reduction technique to return a "Xapian::FeatureVector" with a lower dimension, which can be processed further to give results.

I will look at the work done on ADArank to the xapian-letor project, chalk out any implementation details, and integrate it with xapian-letor.

Parallelizations based on OpenMP can be added as an stretch goal, if all of the above works out well before completition time, along with adding backend support to track the length of each field.

Do you have any preliminary findings or results which suggest that your approach is possible and likely to succeed?

By my experience in Machine learning, and Data mining, both the "Ensemble model by regression" and "principal component analysis" have generally provided me with better results for mostly all datasets.

What other approaches to have your considered, and why did you reject those in favour of your chosen approach?

Dimensionality reduction can be done mainly by two methods, one being the "Principal Component Analysis" and the second being "Singular Value Decomposition". The singular value decomposition is an $O(n^3)$ approach, whereas Principal Component Analysis is a min($O(p^3+n*p^2)$, $O(n^3)$) , where p is the number of features in the input vector, So, implementing PCA made the most sense to me. Other methods of dimensionality reduction are variants of PCA, where we take our input dataset to a higher dimension, where they are linearly separable, but without knowing what our input dataset is, implementing this feature would be really difficult. So providing a feature of PCA, which is valid for a lot of datasets would be my choice.

Please note any uncertainties or aspects which depend on further research or investigation.

The addition of backend support to track the length of fields would avoid having to handle this specially as a feature for Letor is something I'm currently thinking upon. I'm placing this subproblem as a part of a stretch goal so that if I am able to think of a plausible solution, it can be implemented in the remaining time left.

How useful will your results be when not everything works out exactly as planned?

The first part of my project is benchmarking, and testing xapian-letor, as well as adding a regression to combine rankings by different algorithms, so that xapian-letor will be in a releasable stable state with a combining ranking algorithm in place, with the PCA implemented.

Project Timeline

April 23 – May 14 2018: Community Bonding Period:

Expected Deliverables: All working data downloaded, and testing started, with preliminary test examples and reporting as well as solving possible bugs.

1. Writing tests.
2. Setting up tools required.

Also, I will not be able to contribute very actively in the last week due to my end semester exams from May 7th - May 14th.

Week 1: May 14 - May 21

Testing and benchmarking period.

Expected Deliverables: Benchmarked reports for each of the INEX2009 and FIRE dataset. And a report on the run time performance.

1. Run benchmarking tests on the data-sets (INEX and FIRE).
2. Report the evaluation to the mentors.
3. Make sure xapian-letor is stable by reporting and fixing bugs if any.

Week 2: May 14 - May 28

Testing and benchmarking period.

Expected Deliverables: Stability of xapian-letor stress tested with various example datasets.

1. Writing examples to test proper execution.
2. Make sure xapian-letor is stable by reporting and fixing bugs if any.
3. Cleaning up any previous work.

Week 3-4: May 28 - June 8:

The next goal would be to adding a regression to combine multiple tests.

Expected deliverables: Most of the regression implemented along with most working completed.

1. Implement the feedforward backpropagation algorithm to combine various rankers by assignming random weights and then let them adjust according to the algorithm and learning rate.
2. Add regression to xapian-letor, along with the tests.

Week 5: June 8 - June 15:

Expected deliverables: All of the regression code cleaned up, and ready to merge. This period will be kept as a buffer for any pending work.

1. Complete and clean out the code.
2. Will act as a buffer period for any unreported work.
3. Phase 1 evaluation reportable.

Week 6: June 15 - June 22:

Expected deliverables: principal component analysis implemented maintaining basic input vector dimensions and giving a FeatureVector space output.

1. Implementing independent principal component analysis.
2. Check it's functioning.

Week 7: June 22 - June 29:

Expected deliverables: Merging PCA implementation into Xapian-letor.

1. Implementing principal component analysis in the Xapian module.
2. Writing tests for the same.

Week 8: June 29 - July 13:

Expected deliverables: Any previous work not delivered.

1. Get evaluation for the PCA implementation and get it merged into the main module.
2. Clean code and get done with documentation.

Week 10-11: July 13 - July 27:

Expected deliverables: Adding ADARank to xapian-letor rankers.

1. Will chalk out implementation details done by Vhasu and integrate it into xapian.
2. Ensure working after cleaning up and documenting the code.

Week 11-13: July 27 -Aug 10:

Expected deliverables: Clean documented code completed so far, along with proper tests and to pursue one of the stretch goals, mergeable into the main project.

Working on stretch goals and cleaning up existing code and writing good tests to run for the code.

1. Adding a support for backend to track the length of the fields. To allow implementation of weighting schemes like BM25F
2. Where our stretch goal is to add OpenCL and OpenMP parallelization support to training models and improving overall performance.

Previous Discussion of your Project

I've discussed various sub-problems of my project in the IRC under the nick of "addy" with my mentors.

Licensing of my contributions to Xapian

Do you agree to dual-license all your contributions to Xapian under the GNU GPL version 2 and all later versions, and the MIT/X licence?

For the avoidance of doubt this includes all contributions to our wiki, mailing lists and documentation, including anything you write in your project's wiki pages.

Yes. I agree to all your conditions to Xapian under the GNU GPL version 2 and all later versions.

Use of Existing Code

If you already know about existing code you plan to incorporate or libraries you plan to use, please give details.

I plan on using a linear algebra library to implement PCA, to get the eigenvalues and eigenvectors, most probably using the "Eigen" library, if not instructed to use anything else by my mentors.