Implementing of RankNet

Ryan Greenup

February 16, 2021

Contents

#+TODO: TODO IN-PROGRESS WAITING DONE

1 Introduction

Ranknet is an approach to *Machine-Learned Ranking (often refered to as "//Learning to Rank*"/ [10])/ that began development at Microsoft from 2004 onwards [3], although previous work in this area had already been undertaken as early as the 90s (see generally [6, 5, 6, 7, 18]) these earlier models didn't perform well compared to more modern machine learning techniques [11, §15.5].

This paper hopes to serve as an introduction to the implementation of this technique.

The Ranknet approach is typically implemented using Neural Networks, an early goal of this research was to evaluate the performance of different machine learning algorithms to implement the Ranknet method, this research however is still ongoing.

Further Research to look at the implementation of Ranknet for documents and comparing different approaches to apply the method to on-demand queries is required, although this seems to have been implemented by open source Apache Solr project [12], which may provide guidance for further study. An open question is how the performance of Ranknet performs compared to alternative pre-existing methods like Recoll [9] and docfetcher [4].

Ranking/ is the process of applying machine learning algorithms to ranking problems, it .

2 Motivation

A lot of data cannot be clearly categorised or quantified even if there is a capacity to compare different samples, the motivating example is a collection of documents, it might be immediately clear to the reader which documents are more relevant than others, even if the reader would not be able to quantify a "relevance score" for each document.

By training a model to identify a more relevant document, a ranking can be applied to the data.

An example of this might be identifying documents in a companies interwiki that are relevant for new employees, by training the model to rank whether one document is more relevant than an other, ultimately an ordered list of documents most relevant for new employees could be created.

3 Implementation

This implementation will first apply the approach to a simple data set so as to clearly demonstrate that the approach works, following that the model will be extended to support wider and more complex data types before finally being implemented on a corpus of documents.

3.1 Neural Network

Neural Networks [15]

The Ranknet method is typically implemented using a Neural Network, although other machine learning techniques can also be used [3, p. 1], Neural Networks are essentially a collection of different regression models that are fed into one another to create a non-linear classifier, a loss function is used to measure the performance of the model with respect to the parameters (e.g. RMSE [1] or BCE [2]) and the parameters are adjusted so as to reduce this error by using the *Gradient Descent Technique* (although there are other optimisation algorithms such as RMSProp and AdaGrad [13] that can be shown to perform better see [1]). The specifics of Neural Networks are beyond the scope of this paper (see [8] or more generally [15]).

3.1.1 The Ranknet Method

The Ranknet method is concerned with a value p_{ij} that measures the probability that an observation i is ranked higher than an observation j.

A Neural Network (n) is trained to return a value s_k from a feature vector $\mathbf{X}_k$:

$$n(\mathbf{X}_i) = s_i \quad \exists k$$

So as to minimise the error of:

$$p_{ij} = \frac{1}{1 + e^{\sigma \cdot (s_i - s_j)}} \quad \exists \sigma \in \mathbb{R}$$

3.1.2 Implementation

The first step is to create a simple data set and design a neural network that can classify that data set, this can then be extended.

3.2 How to clone

How can the reader clone this onto there machine?

put on the summer repo then provide instructions to clone this working example onto there machine to try it out.

3.3 Blobs

3.4 Moons

3.5 Optimisers

3.6 Batches

3.7 Wine

3.8 Rank Wiki Articles

[1]**RMSE** *Root Mean Square Error*

[2]**BCE** *Binary Cross Entropy*

4 Conclusion

5 Further Research

- Apply this to documents to get a sorted list.
- The "Quicksort" algorithm likely needs a random pivot to be efficient [16]

6 Text and References

Fractals are complex shapes that often occur from natural processes, in this report we hope to investigate the emergence of patterns and complex structures from natural phenomena. We begin with an investigation into fractals and the concept of dimension and then discuss links between fractal patterns and natural processes.

This is a Reference [17] and another [14] and yet another [2].

7 Fractals

Images are shown in figure .

References

[1] Vitaly Bushaev. *Understanding RMSprop — Faster Neural Network Learning*. Medium. Sept. 2, 2018. URL: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a (visited on 02/16/2021) (cit. on p. 3).

[2] Christopher Burges. *From RankNet to LambdaRank to LambdaMART: An Overview (MSR-TR-2010-82*. Jan. 1, 2010. URL: https://www.microsoft.com/en-us/research/uploads/prod/2016/02/MSR-TR-2010-82.pdf (cit. on p. 4).

[3] Christopher Burges. *RankNet: A Ranking Retrospective*. Microsoft Research. July 7, 2015. URL: https://www.microsoft.com/en-us/research/blog/ranknet-a-ranking-retrospective/ (visited on 02/15/2021) (cit. on pp. 2, 3).

[4] Docfetcher Development Team. *DocFetcher - Fast Document Search*. URL: http://docfetcher.sourceforge.net/en/index.html (visited on 02/15/2021) (cit. on p. 2).

[5] Norbert Fuhr. "Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle". In: *ACM Transactions on Information Systems* 7.3 (July 1, 1989), pp. 183–204. ISSN: 1046-8188. DOI: 10.1145/65943.65944. URL: https://doi.org/10.1145/65943.65944 (visited on 02/15/2021) (cit. on p. 2).

[6] Norbert Fuhr. "Probabilistic Models in Information Retrieval". In: *The Computer Journal* 35.3 (June 1, 1992), pp. 243–255. ISSN: 0010-4620. DOI: 10.1093/comjnl/35.3.243. URL: https://doi.org/10.1093/comjnl/35.3.243 (visited on 02/15/2021) (cit. on p. 2).

[7] Fredric C. Gey. "Inferring Probability of Relevance Using the Method of Logistic Regression". In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '94. Berlin, Heidelberg: Springer-Verlag, Aug. 1, 1994, pp. 222–231. ISBN: 978-0-387-19889-7 (cit. on p. 2).

[8] HMKCode. *Backpropagation Step by Step*. URL: https://hmkcode.com/ai/backpropagation-step-by-step/ (visited on 02/16/2021) (cit. on p. 3).

[9] Jean-Francois Dockes. *Recoll User Manual*. URL: https://www.lesbonscomptes.com/recoll/usermanual/usermanual.html (visited on 02/15/2021) (cit. on p. 2).

[10] Tie-Yan Liu. "Learning to Rank for Information Retrieval". In: *Foundations and Trends® in Information Retrieval* 3.3 (June 26, 2009), pp. 225–331. ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000016. URL: https://www.nowpublishers.com/article/Details/INR-016 (visited on 02/15/2021) (cit. on p. 2).

[11] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. New York: Cambridge University Press, 2008. 482 pp. ISBN: 978-0-521-86571-5 (cit. on p. 2).

[12] 21 Nov 2017 Michael A. Alcorn Michael Alcorn. *An Introduction to Machine-Learned Ranking in Apache Solr*. Opensource.com. URL: https://opensource.com/article/17/11/learning-rank-apache-solr (visited on 02/15/2021) (cit. on p. 2).

[13] Mahesh Chandra Mukkamala and Matthias Hein. "Variants of RMSProp and Adagrad with Logarithmic Regret Bounds". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70.* ICML'17. Sydney, NSW, Australia: JMLR.org, Aug. 6, 2017, pp. 2545–2553 (cit. on p. 3).

[14] Olympia Nicodemi, Melissa A. Sutherland, and Gary W. Towsley. *An Introduction to Abstract Algebra with Notes to the Future Teacher.* Upper Saddle River, NJ: Pearson Prentice Hall, 2007. 436 pp. ISBN: 978-0-13-101963-8 (cit. on p. 4).

[15] Philip Picton. *Neural Networks.* Basingstoke, Hampshire ; New York: Palgrave, 1994. 195 pp. ISBN: 978-0-333-94899-6 (cit. on pp. 2, 3).

[16] Tim Roughgarden, director. *Quicksort Overview.* Jan. 28, 2017. URL: https://www.youtube.com/watch?v=ETo1cpLN7kk&list=PLEAYkSg4uSQ37A6_NrUnTHEKp6EkAxTMa&index=25 (visited on 02/15/2021) (cit. on p. 4).

[17] Enmei Tu et al. *A Graph-Based Semi-Supervised k Nearest-Neighbor Method for Nonlinear Manifold Distributed Data Classification.* June 3, 2016. arXiv: 1606.00985 [cs, stat]. URL: http://arxiv.org/abs/1606.00985 (visited on 12/02/2020) (cit. on p. 4).

[18] S. K.M. Wong and Y. Y. Yao. "Linear Structure in Information Retrieval". In: *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR '88. New York, NY, USA: Association for Computing Machinery, May 1, 1988, pp. 219–232. ISBN: 978-2-7061-0309-4. DOI: 10.1145/62437.62452. URL: https://doi.org/10.1145/62437.62452 (visited on 02/14/2021) (cit. on p. 2).