Rank Expert Classification System

Varun Agrawal

Sardar Vallabhbhai National Institute of Technology, Surat.
u09co280@coed.svnit.ac.in

***Abstract*—In this report, an expert system is discussed which helps analyze and predict the best options for students in terms of branch and institute based on their rank in a competitive exam. During the counseling stage for admission to institutes of higher learning, many students are unaware of their best options at their rank hence make erroneous selections during counselling rounds resulting in underestimated or overestimated choices. Hence, the Rank Expert Classification System (RECS) has been devised to help ease this burden from students and help them make better choices for their careers. This system uses machine learning to examine previous records of admissions and predicts the k most likely branches and colleges the student can get admitted in based on their rank, and optionally their preference for any stream or institute. This report starts with explaining the motivation, then moves on to examine approach behind RECS, the advantages of the system and finally the results of test conducted to validate the usefulness of this system**

I. Motivation

Every year, lakhs of students from all over the country appear in various competitive examinations for the chance to obtain a seat in some of the most coveted institutes in India. For example, 10 lakh students appeared for the AIEEE in 2009, 11 lakh in 2011 and 13 lakh students in 2013[1][2].

However, only the top 2% of students roughly who qualify the examinations every year can be assured of which branch and college they are highly likely to obtain. The remaining 98% of students either have to make expensive visits to colleges, or heavily weigh their options against poor choices in order to determine the best way to fill their counseling forms in order to be assured of admittance into a program. This is mostly done by trial and error in most cases to due a lack of clear information, and is thus a tedious process.

Also, this situation is not ameliorated by the Central Counseling Board publishing closing ranks of each department in each institute every year, as these closing ranks are very year specific and subject to change, do not capture changes in admission patterns and do not indicate admission probabilities for individual ranks. The biggest drawback is that these do not capture the preferences of the student being admitted. For example, a student in the top $30,000$ ranks may opt for Mechanical Engineering in an institute due to his preference while another student ranking in lakhs could opt for electronics, thus severely denting the inferential correctness of the closing ranks.

Thus, the Rank Expert Classification System (RECS) has been developed to help overcome these various problems faced by lakhs of students each year and help them make better choices for their careers.

II. Approach

The Matching problem for Bipartite graphs[3] is a well known problem where one tries to match vertices from one part of the graph to the vertices of the other part of the bipartite graph in such a way as to optimize the weight of the edges. This approach is best where the preferences for matching are well-known and unambiguous. This can be extended to our problem of matching students to various departments and colleges. The major issue, however, is that many students are either unaware or confused about which institute should be given greater preference. One good example is that a student might be confused with the choice of a better institute far from his home or an institute closer to his hometown but not as good as the former. Also, the subjective nature of judging an institute adds to the confusion. Hence, even though the Matching problem has a solution, it cannot be applied to this particular instance.

Each year, when students take admissions into colleges, on careful analysis of the admission patterns, it is easy to notice that ranks get clustered for each branch and college. That is, the rank distribution for each branch and institute follows a Gaussian distribution, with one rank having the mean and variance depending on the quality and popularity among students of that branch in the institute. This pattern motivates the decision to use a machine learning algorithm. Since we already have labelled data, and we plan to label new instances of features, this problem comes under the aegis of supervised learning. The above two observations motivate the use of the K-Nearest Neighbour Algorithm[4] for labelling the query point.

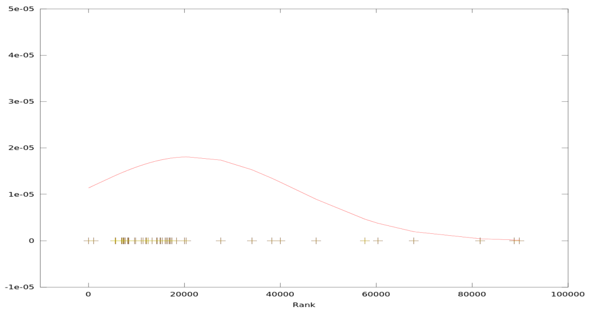

Fig. 1. Clustering of Ranks for admission into Electronics Engineering programme taken from the training data

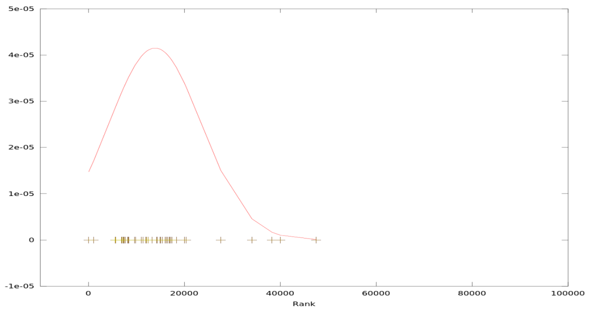

Fig. 2. Clustering of Ranks for admission into Electronics Engineering programme without the last 4 values. The Gaussian curve is much more pronounced.

III. Algorithm

The basic idea of the K-Nearest Neighbour (KNN) algorithm is, given a Hilbert space of classified points and a single unlabelled point, we find the k labelled points that have the minimum distance to the unclassified point, where minimum distance is defined as any function that maps the similarity between two points in the Hilbert space, with more similar points being given higher weightage.

The idea employed is that given the query point, we know it will be in one of the clusters formed as mentioned before, and thus the majority of the points around it will be in that particular cluster, thus giving us a high posterior probability that the query point will be classified correctly. Hence, the decision to use the KNN algorithm.

The main design issue of the algorithm for this application is structuring the data of the students so as to allow for efficient searching of the K nearest neighbours. We require the data to be stored in databases for easy data warehousing, but conventional databases and their management systems are not designed to accommodate changes in their structure to facilitate fast nearest neighbour searching, especially when the minimum distance metric is user defined and not a standard equation. Yao et. al.[5] have proposed a solution to the problem of efficient nearest neighbour search in databases when there is a user defined function over which the distance of the database points are to be mapped. Their solution involves a 3-step procedure:

1) Preprocess the data in an interleaved format at the binary data and use this metric as the index in the database.
2) Using the defined index of the database and a randomly shifted permutation of the data, we look for a c-approximate solution for c>1.
3) We use the c-approximate solution to obtain the K Nearest Neighbours of the given query point.

This approach has a runtime of O(lg(n)) in expectation which is highly efficient for a database containing 10 million or more data points. This efficiency is achieved using the indexing structure of the database which manages the query in O(lg(n)) as well as randomly shifting our data points. However, since the lemma states that the number of random shifts should be O(1), we can assume for our system that the random shift provides no change to the data, hence satisfying the conditions.

The user defined function used for mapping the minimum distance in this case is the Manhattan distance which has hidden benefits as explained later.

IV. Design

The use of the algorithm is made to operate on the backend, using only Structured Query Language (SQL). The algorithm is encoded in SQL and this SQL code can thus be called from any program that can establish a connection to the database. This allows for easy migration and portability of the application. Since the main algorithm is encoded in SQL and there is no need to change the inherent structure of the Database engine, the cost and time for implementing a working, production model is very small, thus giving very high returns and proving optimum for implementation by organizations such as governments and private universities.

Scalability also proves easy as the there is a well defined interface between the front-end and the back-end of the system.

Once the training data is obtained from previous years results, a simple script can be written to preprocess and insert the data into the database, using a programming language Python or Ruby. This a one time operation only for a year and thus the cost in terms of runtime is slight as it can be done immediately after the counseling is completed an automation script.

For the front-end system, all that is required from the user is his rank and optionally his preference for branch and institute which can be used to further refine the search. The branches and institutes are encoded into the database via a string-integer mapping where more popular branches and institutes are given smaller integer values. This gives the benefit that if a student does not provide any preferences, he will automatically be given the preference of the most popular branch and best institute by virtue of the Manhattan distance function as the default value for these fields in zero. Ranking of branches and institutes can be easily obtained by using surveys conducted by various magazines such as Outlook, thus giving students the best choices for their ranks.

All in all, the system is simple to implement and efficient to use. The simplicity allows for all organizations to quickly provide this service online through the internet, and can be thus easily accessed from exam centres or government offices, for free hopefully, if the student does not have internet access. Thus its benefits are very high for the majority of the student community, especially for those who come from poor or rural backgrounds or families where they are the first to qualify in that particular exam such as the AIEEE, paving the way for better education for the youth of our country and establishing our nation onto the road of success and development.

V. Benefits

The main advantages of this system are:

1) Easy to use. A user only has to provide his rank and optionally choose a branch or institute from a list.
2) Availability. Can be hosted on a website that can be accessed from anywhere with an internet connection.
3) Portability since the backend and frontend are separated using a well defined interface and there is no change to the inherent structure of the database and the database engine.
4) Simple to implement as it involves just a few lines of SQL and a few easy scripts.
5) Cost effective. Very less cost overhead as this system can be implemented using free and open source software.
6) Wide use as students all over the country can avail this service and this can be provided by all Institutes that have an entrance examination.

VI. Test Results

On a crowd-sourced dataset from students of Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, consisting of 277 data entries for admissions from the years 2009 and 2010, the model was trained and tested against a test set of 130 data entries for admission in the year 2011 so as to model a real world setting.

Three metrics were used for performance analysis:

1) Does the answer set have atleast one entry showing the correct label - Atleast One Test.
2) How many entries of the answer set have the correct label - Prediction Accuracy Test.
3) What is the minimum variation in ranks for the query point and the entries in the answer set having the correct label - Variation Test.

For the Atleast One Test, the test results provided a 100% prediction accuracy, i.e. the answer set always had one entry with the correct label.

For the Prediction Accuracy Test, the results came at 59.77% which is quite good considering the sparsity of the data.

The results of the Variation Test returned an average variation of 1733.5 ranks in the query point and the closest correct answer point. The variations of all the query points are shown in figure 3. Here, it is quite apparent that the variation in ranks is to an acceptable degree. The 2 outliers can be attributed to the fact that there are very few data points in our training set in the range of the outliers.

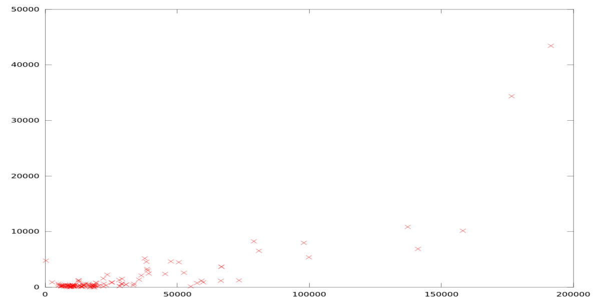

Fig. 3. Scatter-plot showing the minimum variation of ranks for the answer set of each test set query point.

The current results are highly affected by the lack of a dataset. The author has tried his best to obtain a comprehensive dataset but it is yet to be made available. This lack of a dataset means that the test dataset is highly sparse and hence gives poorer results. As found by Banko and Brill[6], the effectiveness of a machine learning system can be highly affected by providing a high density, comprehensive dataset. Thus, better test results may be obtained with the help of a more accurate and complete dataset.

VII. Future Work

The Expert System shown here uses a very basic model for predicting the branches and colleges for a student?s competitive exam rank. This is because of the ease of implementation and that it satisfies many of the required criteria of our problem. This system can be improved further by possibly changing the Database structure to use R-Trees[7] and other machine learning algorithms may be used, such as Neural Networks[8], to provide better performance. In all, it is very much possible to deploy this kind of system for use by the general population and any improvements to the system is at the discretion of the implementing organization.

VIII. Conclusion

The design and implementation of the RECS system has been documented and explained. The advantages of this system have also been shown and they prove this system is not only easy to implement, but is highly cost-effective and provides superior performance. Analysis of the test results shows promise as well, but since the dataset is not comprehensive and official, the test results may hide more than what they reveal. All in all, this system is one which has widespread social and educational use and can help hundreds of students make better choices in their education.

IX. Acknowledgements

The author would like to thank Mrs. Nirali Nanavati, faculty at SVNIT Surat for providing the guidance and evaluation for this project, Mrs. Rupa G. Mehta, Asst. Professor, SVNIT Surat, for her expert guidance and advice in the field of machine learning and Prof. Andrew Ng, Professor, Stanford University, for providing guidance in machine learning system evaluation.

References

[1] "Aieee speed news." http://timesofindia.indiatimes.com/topic/Aieee, 2012.

[2] Wikipedia, "All india engineering entrance examination." http://en.wikipedia.org/wiki/All_India_Engineering_Entrance_Examination, 2012.

[3] R. Sedgewick and K. Wayne, *Algorithms, 4th Edition*. Addison-Wesley, 2011.

[4] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, pp. 21 –27, january 1967.

[5] B. Yao, F. Li, and P. Kumar, "K nearest neighbor queries and knn-joins in large relational databases (almost) for free," in *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pp. 4 –15, march 2010.

[6] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, (Stroudsburg, PA, USA), pp. 26–33, Association for Computational Linguistics, 2001.

[7] A. Guttman, "R-trees: a dynamic index structure for spatial searching," *SIGMOD Rec.*, vol. 14, pp. 47–57, June 1984.

[8] G. Zhang, "Neural networks for classification: a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 30, pp. 451 –462, nov 2000.