BOND LIQUIDITY PROBLEM

team_lcm1

02nd October 2016

1 Libraries Used

- Sklearn - Provides Machine Learning Algorithms
- Pandas - Provides high-performance, easy-to-use data structures for handling large dataset.
- Numpy - Provides N-dimensional array objects on which machine learning algorithms are easily operable.

2 Pre-Processing

- The data is read by using panda's internal read_csv function.
- The fields are divided into various types :
  a. Numerical
  b. Categorical
  c. Date
  d. Boolean field
- Proper cleaning is performed depending on the type of the field.
- Normalization is then performed in order to transform the integer fields within [-1,1].

3 Assumptions

- Approximated missing values by the mean average
- Didn't consider the impact of session of trade (morning/afternoon) as we need to calculate the volumes of trade for complete day and not a particular section
- The Trade volume of bond is majorly affected by its nearest neighbours in terms of distance

4 Algorithms

- **Partial K-Means Clustering**
  We identify the bonds which are close to each other with respect to bond characteristics. This is done to reduce the computation required while determining the volume for future bonds. The volume of any bond then affects the **expected** volumes of related bonds.

- **Main Idea**
  Initially we have the total buy and sell volumes of each bond in the three months. Our idea is based on the observation that the effect on bond's volumes of trade is dominantly affected by similar bonds and depends insignificantly on less similar bonds.
  Thus, formally, the Expected Volume for any bond 'i' is equal to

  $$EV(i) = \sum z(i,j) * v(j) \tag{1}$$

  where 'j' corresponds to all the bonds in the same cluster

  $$z(i,j) = \frac{1}{1+r^2} \tag{2}$$

  where r is the distance between bond i and j.
  Now, to normalize this expected volume, we multiply it by (sum_initial/sum_final) where sum_initial is sum of total volumes and sum_final is sum of total expected volumes.

  This ensures that our processing just changes the distribution of the volumes and not the total value i.e.
  Sum of Total Expected Volume of Trade = Sum of Total Volumes.

  Now, to get the value for the next three days, we assumed it to be independent of the weekday and thus multiply it by (3/90) i.e. (1/30).

5 Executing The Code

- Create a local folder /data with the following files :
  a. Bond_Metadata.csv
  b. dataset.csv

- Create a local folder /output. The final output file would be saved at this location

- Run using the command "python predict.py"

6 Further Ideas

- The trade volume of a given bond is affected dominantly by trades in the recent days than those in the earlier time. Thus, an exponential factor need to be added as per the date of the trade while considering the total volume traded.

- We wanted to analyze the impact of different week-days on trade performance. So, we can analyze the volumes according to week days and include that factor as well.