Null and Outlier Detection

Zhe Huang, Daoyang Shan, Yueqiu Sun

Team Singular Point

Introduction

In our project, we proposed to implement **a generalized automated procedure that** cleans and selects the appropriate outlier detection strategy for any given data set (if possible) **without exploring its specific domain knowledge.**

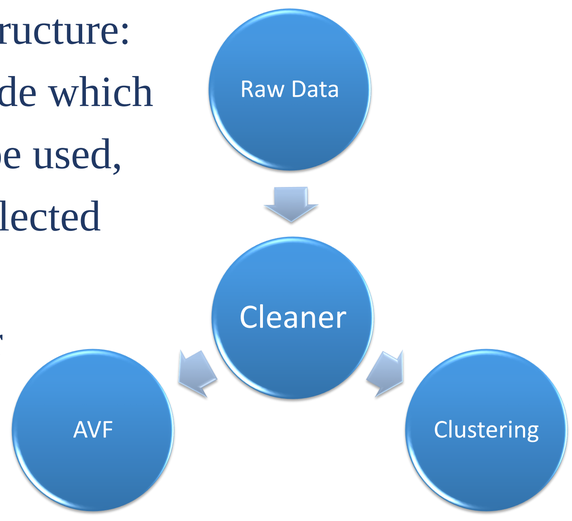
Methodology

Our design follows the following structure:

1: Cleaner cleans the raw data, decide which outlier detection algorithm should be used, and preprocess the data based on selected algorithm.

2: Processed data is passed to either k-means clustering or AVF, where nulls and outliers are detected.

3: Use shell script to automate the whole process instead of manual work on individual file.



Cleaner:

- Remove those columns with too many nulls or blanks.
- Classify all columns into numerical, categorical or other.
- Drop ‘others’ since we assume no domain knowledge.
- If the rest are mainly numerical, select k-means, otherwise apply binning on numerical columns and select AVF.

K-Means (distance based algorithm):

- Identify rows with blank or null as null points.
- Specify a range of possible k’s that we explore. For each k, apply clustering and get the corresponding WSSSE score.
- Find the optimal k based on the largest second order derivative of the approximated WSSSE curve.
- Assign each point to a cluster, find those points that far away from their centroids as outliers.

AVF (frequency based algorithm):

- Calculate the AVF score for each point (row).
- Points with the least AVF score are chosen as outliers.
- Our first implementation is finished with Spark.

Animal	Tag	AVF Score
Cat	A	2+2=4
Cat	B	2+1=3
Dog	A	1+2=3

foreach *point* $\mathbf{x}_i$, $i = 1 \ldots n$ **do**
  foreach *attribute* l, $l = 1 \ldots m$ **do**
    Count frequency $f(x_{il})$ of attribute value x_{il};
  end
end
foreach *point* $\mathbf{x}_i$, $i = 1 \ldots n$ **do**
  foreach *attribute* l, $l = 1 \ldots m$ **do**
    $AVFScore(\mathbf{x}_i)$ += $f(x_{il})$;
  end
  $AVFScore(\mathbf{x}_i)$ /= m;
end
Return k outliers with $min_i(AVFScore)$;

MR-AVF (Map-Reduce version of AVF):

- Finished with 2 Map-Reduce phases.
- O(1) memory, O(n) mapper/reducer time complexity.
- Provides speedup with respect to number of processing nodes.
- Algorithm detail available in our final report.

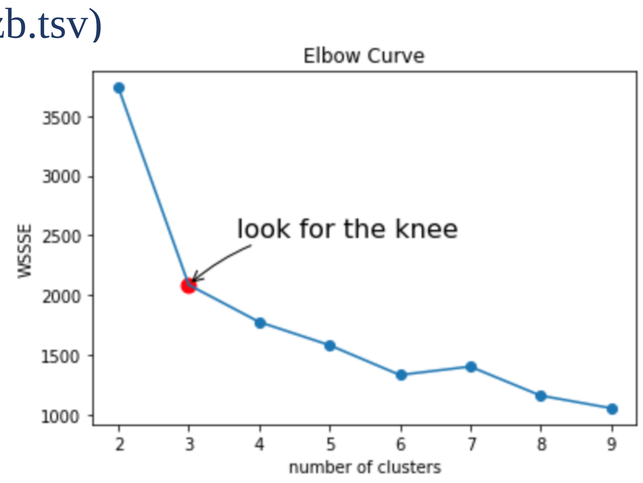
Result and Discussion

Cleaner:

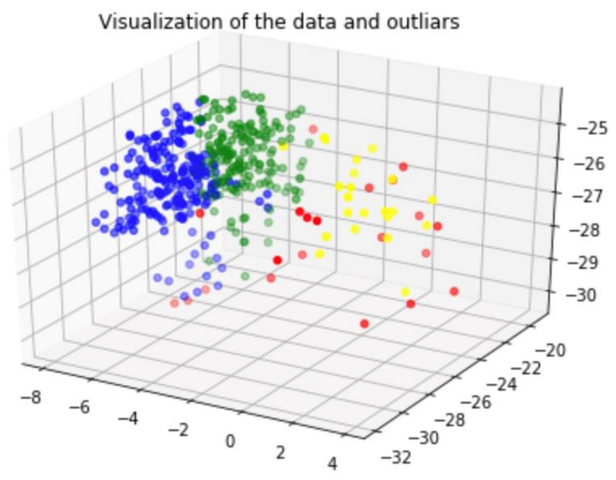
- Our cleaner determines that 37 out of 50 data sets are suitable for automatic outlier detection.
- Dropped data sets include: time series data, messy data, too many blank/null for every column……
- 8 suitable for clustering, rest for AVF.

Sample Clustering (ny8v-zzzb.tsv)

J	K	L
mean_scale_score	prediction	distance
671	1	6.874173641
652	1	6.305948257
651	1	6.281490326
665	1	5.65323782
664	1	4.99644804
669	1	4.964539528
658	1	4.883923054
635	1	4.731093407
667	2	4.613997459
636	1	4.500597954



- Points with max ‘distance’ are classified as outliers.
- k=3, visualization.



AVF vs. MR-AVF

- Sample output

index	avf
186822	6317842
342131	6758317
514121	6858309
562523	6859581
631511	6861442

Points with least AVF Score are classified as outliers

- We record the **wall time** of the whole process.
- Time may be influenced by I/O, num of rows, difficulty of processing, etc.
- It seems that MR-AVF is better for large data sets.

Name	Size	MR-AVF(time)	AVF(time)
pvqr-7yc4	6.4*1e8	356	1400
tm6d-hbzd	2.6*1e8	123	200
r4s5-tb2g	360725	55	20

More Discussion

- More complicated cleaning mechanism?
- Better outlier detection algorithm for hybrid data set (to our surprise, almost no research is focused on hybrid data)