Beyond outliers and on to micro-clusters: Vision-guided Anomaly Detection
Supplementary Document

Wenjie Feng[1,2], Shenghua Liu[1,2], Christos Faloutsos[3], Bryan Hooi[3], Huawei Shen[1,2], Xueqi Cheng[1,2]

[1] CAS Key Laboratory of Network Data Science & Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] School of Computer Science, Carnegie Mellon University, PA, USA
fengwenjie@ict.ac.cn, liu.shengh@gmail.com, christos@cs.cmu.edu, bhooi@andrew.cmu.edu, {shenhuawei,cxq}@ict.ac.cn

Abstract. This document supplements by giving detail information of EagleMine algorithm, the proof of theoretical analysis for the time complexity, and details on additional experiments.

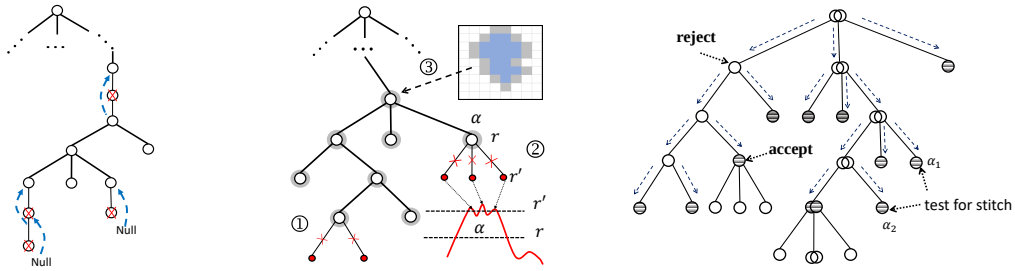
1 EagleMine Detail Information

For the WaterLevelTree algorithm, we utilize following steps to refine the raw tree $\mathcal{T}$: Contract, Prune, and Expand, and the Fig. 1 illustrates intuitive pictorial explanation.

- **Contract** aims to remove the single-child nodes in $\mathcal{T}$. This process is shown in Fig. 1a, where the dashed lines with arrow depict that the single-child's children are linked to its parent, and the gray links are removed.
- **Prune** is to alleviate the noise peaks on some island. An example is shown at the bottom right of Fig. 1b. The island α at water-level r contains fluctuation noises on its top. When the water level raises to r', these noises become three separated 'tiny' islands linking to their parent α. So we will remove these 'noisy' for smoothing.
- **Expand** includes some surrounding bins of an island to avoid over-fitting learning. The expansion of node is illustrated with shadowed rings in Fig.1b. The node ③ (light-blue part) is expand with 1-st step outer-peripheral gray bins. For the node ③, the light-blue irregular part represents the original island area and the outer-peripheral gray bins are expanded part in 1-st step as the pictorial depiction shows, and the further expansion follows a similar process until it gets to above constraint.

In the TreeExplore algorithm for determining the optimal islands and their description, we search the tree $\mathcal{T}$ and get the summarizations by using statistical hypothesis test as the selection criteria. The dashed lines with arrow in Fig. 1c depict the search trace. Moreover, the islands α_1 and α_2 in Fig. 1c, which are physically close to each other, depicts our motivation for the **stitch** process, that means α_1 and α_2 can be described with the same distribution rather than separately.

In the sequel, the above careful designed refinement empower EagleMine to better recognize and summarize the node group distribution in the histogram.



(a) Tree contract (b) Tree prune and island expand (c) Optimal islands search

Fig. 1: Key steps in proposed EagleMine algorithms.

2 PROOF of Time Complexity In SEC. 3.4

For the histogram $\mathcal{H}$, as we described before, $h_{max} = \max \mathcal{H}$. Let $nnz(\mathcal{H})$ denote the number of non-empty bins in $\mathcal{H}$ and $\mathcal{C}$ be the number of clusters. We use gradient-descent to learn the parameters in $DistributionFit(\cdot)$ of EagleMine algorithm, where we assume that the number of iterations is T. So T is related to the differences between initial and optimal objective values. Then the time complexity of EagleMine is: $O(\frac{\log h_{max}}{\rho} \cdot nnz(\mathcal{H}) + \mathcal{C} \cdot T \cdot nnz(\mathcal{H}))$.

Proof. First, WaterLevelTree is invoked by EagleMine as the first subprocedure, in which we compare all $nnz(\mathcal{H})$ non-empty bins with water level r in step 3, and then do binary opening [5] to remove small blobs (noise) by checking non-empty ones in step 4, both of which cost $O(nnz(\mathcal{H}))$. From steps 5 to 6, for each island, we connect its children (# of islands $< nnz(\mathcal{H})$) to it, so the time cost equals to the number of links, i.e. $O(nnz(\mathcal{H}))$. Hence the whole iteration from step 2 to 7 takes $O(\tau \cdot nnz(\mathcal{H}))$, where $\tau = {}^{\log h_{\max}}/_{\rho}$. As a result, we get a tree $\mathcal{T}$, whose height is τ and width is at most $nnz(\mathcal{H})$. The total number of links in that tree are less than $\tau \cdot nnz(\mathcal{H}))$. Afterwards, the operation of contracting takes $O(\tau \cdot nnz(\mathcal{H}))$. In each tree level, the summation of bins in islands is less than $nnz(\mathcal{H})$, so the complexity of both pruning and expanding process is also $O(\tau \cdot nnz(\mathcal{H}))$. Consequently, the costs of constructing the water-level tree is $O(\tau \cdot nnz(\mathcal{H}))$.

In the TreeExplore algorithm, function $DistributionFit(\cdot)$ costs $O(T \cdot nnz(\mathcal{H}))$, where each gradient-descent cost $O(nnz(\mathcal{H}))$, the number of training data. Since our algorithm finds C micro-clusters when stops, the subtree with visited nodes by BFS search on $\mathcal{T}$ has C leaves. Due to the contraction of WaterLevelTree algorithm at step 8, each non-leaf node in the subtree has at least two children, hence the subtree has at most $2 \cdot C$ nodes, which means the step 3 in TreeExplore have at most $2 \cdot C$ times of choosing the largest island, conducting $DistributionFit(\cdot)$, and applying hypothesis tests. The cost of statistical hypothesis test on each node (island) is linear of the number of bins in the island, which is less than $nnz(\mathcal{H})$. During stitching, we only test those islands close to each other in a plane, which costs less than the above process on tree $\mathcal{T}$.

Therefore, the time complexity of EagleMine is

$$O(\tau \cdot nnz(\mathcal{H}) + 2C \cdot (T \cdot nnz(\mathcal{H}) + nnz(\mathcal{H}))) = O(\frac{\log h_{max}}{\rho} \cdot nnz(\mathcal{H}) + C \cdot T \cdot nnz(\mathcal{H}))$$

where $C \ll nnz(\mathcal{H})$.

3 Additional Experiments

3.1 Experiments setting detail

To quantitative evaluate the summarization performance, we selected X-means [7], G-means[6], DBSCANc̃iteester1996dbscan, STING [8] and EagleMine (DM) as the comparisons, their settings are listed as follows.

- X-means: initialize with k-means and 5 clusters.
- G-means: set $max\_depth = 5$, limiting no more than 16 clusters to avoid too many clusters; set p-value= 0.01 which is insensitive.
- DBSCAN: set Eps=1, and use $\|h_i - h_j\|_\infty$ as distance function; we searched MinPts from the average number of nodes in a histogram bin until the max number by step 50, and manually select the one consistent well with human vision[4] judgment.
- STING: $c \approx \frac{\text{Minpts}+1}{\pi \text{Eps}^2}$ with DBSCAN's tuned optimal MinPts and Eps for clusters as initial, and refine the visual result by fine-tuning.
- EagleMine and EagleMine (DM): are our proposed EagleMine with DTM Gaussian and whole multivariate Gaussian respectively.

In addition, the Minimum Description Length (MDL) is used as the measure for summarization. The MDL lengths for the baselines are calculated as [2,3,4], while the MDL of EagleMine is:

$$L = \log^*(C) + L_{\mathcal{S}} + L_{\Theta} + L_{\mathcal{O}} + L_{\epsilon}$$

This description of model consists of following terms:

- The number of clusters requires $\log^*(C)$ bits. [5].
- The assignment $\mathcal{S}$ of distribution vocabulary to C groups requires $L_{\mathcal{S}} = C \cdot \log(\mathcal{Y})$ bits.
- Each DTM Gaussian need $|\theta| = F + \frac{(1+F)F}{2} + 1$ free parameters. If we have two features, i.e., $F = 2$, then $|\theta| = 6$ for 2D distribution. So its encoding requires $|\theta| \cdot l_0$ bits, where l_0 is the floating point cost. We used 4×8 bits in our setting. The total parameter code-length is $L_{\Theta} = C|\theta| \cdot l_0$.
- The outliers $\mathcal{O}$ require $L_{\boldsymbol{O}}$ bits to encode bin indices.
- The model error requires L_ϵ bits. For a bin $\boldsymbol{b}$ in group (island) α_i, the expected number of nodes is $\tilde{h}=\left\lfloor 2^{\tilde{N}_i \cdot P(\boldsymbol{b}|\theta_i)} \right\rfloor$. Then the original count can be accurately recovered as $h=\tilde{h}+\epsilon$, Thus we encode the total description error as $L_\epsilon=\sum_{\boldsymbol{b}}(\log^*(h-\tilde{h})+1)$, where 1 is the code length of the sign.

[4] Since DBSCAN is manually tuned, we do not use OPTICS [1] to search parameters for DBSCAN.

[5] Here, $\log^*$ is the universal code length for integers, defined as $\log^*(x) \approx \log_2(x) + \log_2\log_2(x) + \cdots$. where only the positive terms are included in the sum. [4]

3.2 Q2. Summarization Evaluation of EagleMine

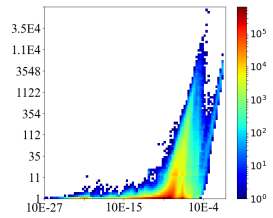
We conduct extensive experiments on real dataset to verify the summarization performance of EagleMine algorithm, the Fig. 2 provides more examples for qualitative evaluation on different datasets and vary feature spaces including out-degree vs hubness, in-degree vs authority, and #triangle vs degree, etc.

As the Fig. 2 shows, the original histogram plots are given at the beginning of each group following with label figures (only reflect the micro-cluster areas) of different methods. Outliers (bins) are labeled with the *blue* color and 'x' marker. Different colors represent different groups by corresponding methods. From original plots, people will naturally expect that bins with the similar color (density) and jointed locations should be in one group. Hence we can see that G-means and X-means produce a number of groups, over-separating the groups recognized by human vision. Although manually tuned DBSCAN and STING can capture the majority dense region in each plot, while overlooking some suspicious micro-clusters, e.g., micro-clusters Ⓐ and Ⓒ in Fig. 2c.

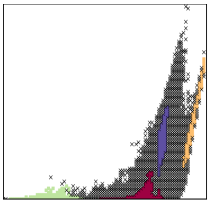
Thus, EagleMine illustrates its advantages of recognizing groups, especially identifying micro-clusters, which is more consistent with human vision.

References

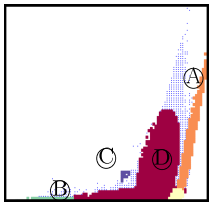
1. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. In: SIGMOD. pp. 49–60 (1999)
2. Böhm, C., Faloutsos, C., Pan, J.Y., Plant, C.: Robust information-theoretic clustering. In: KDD
3. Chakrabarti, D., Papadimitriou, S., Modha, D.S., Faloutsos, C.: Fully automatic cross-associations. In: SIGKDD. pp. 79–88 (2004)
4. Elias, P.: Universal codeword sets and representations of the integers. IEEE trans. on information theory pp. 194–203 (1975)
5. Gonzalez, R.C., Woods, R.E.: Image processing. Digital image processing (2007)
6. Hamerly, G., Elkan, C.: Learning the k in k-means. NIPS (2004)
7. Pelleg, D., Moore, A.W., et al.: X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML. pp. 727–734 (2000)
8. Wang, W., Yang, J., Muntz, R., et al.: Sting: A statistical information grid approach to spatial data mining. In: VLDB. pp. 186–195 (1997)



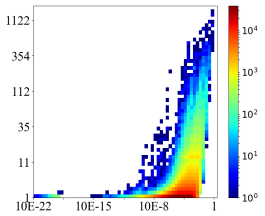
(a) In-degree vs Authority



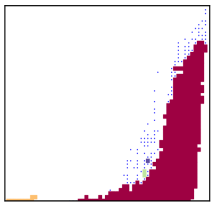
(b) Watershed label {4}



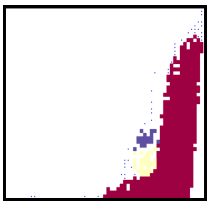
(c) EagleMine label {5}



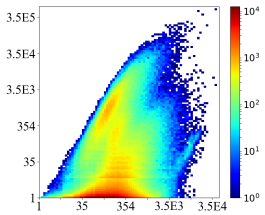
(d) Out-degree vs Hubness



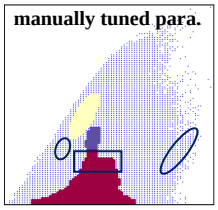
(e) STING label {4}



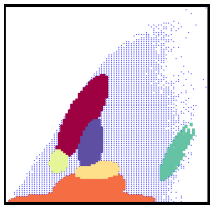
(f) EagleMine label {3}



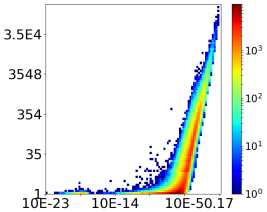
(g) #Triangle vs Degree



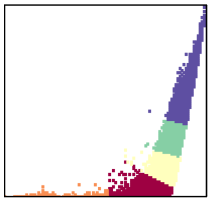
(h) DBSCAN label {3}



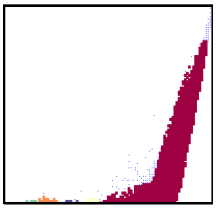
(i) EagleMine label {6}



(j) In-degree vs Authority



(k) X-means label {5}



(l) EagleMine label {4}

Fig. 2: EagleMine visually recognizes better node groups than baselines in qualitative comparison. '{·}' gives the number of node groups recongnized by each algorithm. **(a)-(c):** is in-degree vs. authority plot of Sina weibo data, which are message nodes corresponding to user nodes in out-degree vs.hubness feature space. **(d)-(f):** is out-degree vs. hubess plot for user-products online review in Yelp. **(g)-(i):** is #triangle vs. degree plot of homogeneous graph for users from Tagged website. **(j)-(l):** is in-degree vs. authority plot for the users to associated groups in Flickr.