

Improving iForest with Relative Mass

Sunil Aryal¹, Kai Ming Ting², Jonathan R. Wells¹, and Takashi Washio³

¹ Monash University, Victoria, Australia
{sunil.aryal, jonathan.wells}@monash.edu
² Federation University, Victoria, Australia
kaiming.ting@federation.edu.au
³ Osaka University, Osaka, Japan
washio@ar.sanken.osaka-u.ac.jp

Abstract. iForest uses a collection of isolation trees to detect anomalies. While it is effective in detecting global anomalies, it fails to detect local anomalies in data sets having multiple clusters of normal instances because the local anomalies are masked by normal clusters of similar density and they become less susceptible to isolation. In this paper, we propose a very simple but effective solution to overcome this limitation by replacing the global ranking measure based on path length with a local ranking measure based on relative mass that takes local data distribution into consideration. We demonstrate the utility of relative mass by improving the task specific performance of iForest in anomaly detection and information retrieval tasks.

Keywords: Relative mass, iForest, ReFeat, anomaly detection.

1 Introduction

Data mining tasks such as Anomaly Detection (AD) and Information Retrieval (IR) require a ranking measure in order to rank data instances. Distance or density based methods are widely used to rank instances in these tasks. The main problem of these methods is that they are computationally expensive in large data sets because of their high time complexities.

Isolation Forest (iForest) [1] is an anomaly detector that does not use distance or density measure. It performs an operation to isolate each instance from the rest of instances in a given data set. Because anomalies have characteristics of being ‘few and different’, they are more susceptible to isolation in a tree structure than normal instances. Therefore, anomalies have shorter average path lengths than those of normal instances over a collection of isolation trees (iTrees).

Though iForest has been shown to perform well [1], we have identified its weakness in detecting local anomalies in data sets having multiple clusters of normal instances because the local anomalies are masked by normal clusters of similar density; thus they become less susceptible to isolation using iTrees. In other words, iForest can not detect local anomalies because the path length measures the degree of anomaly globally. It does not consider how isolated an instance is from its local neighbourhood.

iForest has its foundation in mass estimation [2]. Ting et al [2] have shown that the path length is a proxy to mass in a tree-based implementation. From this basis, we analyse that iForest’s inability to detect local anomalies can be overcome by replacing the global ranking measure based on path length with a local ranking measure based on relative mass using the same iTrees. In general, relative mass of an instance is a ratio of data mass in two regions covering the instance, where one region is a subset of the other. The relative mass measures the degree of anomaly locally by considering the data distribution in the local regions (superset and subset) covering an instance.

In addition to AD, we show the generality of relative mass in IR that overcomes the limitation of a recent IR system called ReFeat [3] that uses iForest as a core ranking model. Even though ReFeat performs well in content-based multimedia information retrieval (CBMIR) [3], the ranking scheme based on path length does not guarantee that two instances having a similar ranking score are in the same local neighbourhood. The new ranking scheme based on relative mass provides such a guarantee.

The contributions of this paper are as follows:

1. Introduce relative mass as a ranking measure.
2. Propose ways to apply relative mass, instead of path length (which is a proxy to mass) to overcome the weaknesses of iForest in AD and IR.
3. Demonstrate the utility of relative mass in AD and IR by improving the task specific performance of iForest and ReFeat using exactly the same implementation of iTrees as employed in iForest.

The rest of the paper is organised as follows. Section 2 introduces the notion of relative mass and proposes ways to apply to AD and IR. Section 3 provides the empirical evaluation followed by conclusions in the last section.

2 Relative Mass: A Mass-Based Local Ranking Measure

Rather than using the global ranking measure based on path length in iForest, an instance can be ranked using a local ranking measure based on relative mass w.r.t its local neighbourhood. In a tree structure, the relative mass of an instance is computed as a ratio of mass in two nodes along the path the instance traverses from the root to a leaf node. The two nodes used in the calculation of relative mass depend on the task specific requirement.

- In AD, we are interested in the relative mass of \mathbf{x} w.r.t its local neighbourhood. Hence, the relative mass is computed as the ratio of mass in the immediate parent node and the leaf node where \mathbf{x} falls.
- In IR, we are interested in the relative mass of \mathbf{x} w.r.t to a query \mathbf{q} . Hence, the relative mass is computed as the ratio of mass of the leaf node where \mathbf{q} falls and the lowest node where \mathbf{x} and \mathbf{q} shared along the path \mathbf{q} traverses.

We convert iForest [1] and ReFeat [3] using the relative mass, and named the resultant relative mass versions, ReMass-iForest and ReMass-ReFeat, respectively. We describe iForest and ReMass-iForest in AD in Section 2.1; and ReFeat and ReMass-ReFeat in IR in Section 2.2.

2.1 Anomaly Detection: iForest and ReMass-iForest

In this subsection, we first discuss iForest and its weakness in detecting local anomalies and introduce the new anomaly detector, ReMass-iForest, based on the relative mass to overcome the weakness.

iForest

Given a d -variate database of n instances ($D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$), iForest [1] constructs t iTrees (T_1, T_2, \dots, T_t). Each T_i is constructed from a small random sub-sample ($\mathcal{D}_i \subset D, |\mathcal{D}_i| = \psi < n$) by recursively dividing it into two non-empty nodes through a randomly selected attribute and split point. A branch stops splitting when the height reaches the maximum (H_{max}) or the number of instances in the node is less than $MinPts$. The default values used in iForest are $H_{max} = \log_2(\psi)$ and $MinPts = 1$. The anomaly score is estimated as the average path length over t iTrees as follows:

$$L(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^t \ell_i(\mathbf{x}) \quad (1)$$

where $\ell_i(\mathbf{x})$ is the path length of \mathbf{x} in T_i

As anomalies are likely to be isolated early, they have shorter average path lengths. Once all instances in the given data set have been scored, the instances are sorted in ascending order of their scores. The instances at the top of the list are reported as anomalies.

iForest runs very fast because it does not require distance calculation and each iTree is constructed from a small random sub-sample of data.

iForest is effective in detecting global anomalies (e.g., a_1 and a_2 in Figures 1a and 1b) because they are more susceptible to isolation in iTrees. But it fails to detect local anomalies (e.g., a_1 and a_2 in Figure 1c) as they are less susceptible to isolation in iTrees. This is because the local anomalies and the normal cluster C_3 have about the same density. Some fringe instances in the normal cluster C_3 will have shorter average path lengths than those for a_1 and a_2 .

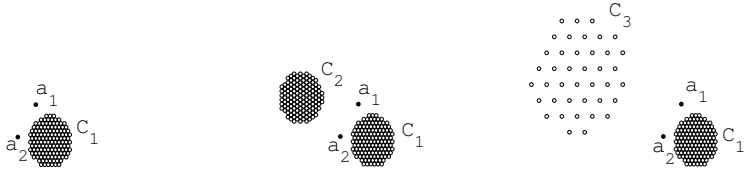
ReMass-iForest

In each iTree T_i , the anomaly score of an instance \mathbf{x} w.r.t its local neighbourhood, $s_i(\mathbf{x})$, can be estimated as the ratio of data mass as follows:

$$s_i(\mathbf{x}) = \frac{m(\check{T}_i(\mathbf{x}))}{m(T_i(\mathbf{x})) \times \psi} \quad (2)$$

where $T_i(\mathbf{x})$ is the leaf node in T_i in which \mathbf{x} falls, $\check{T}_i(\mathbf{x})$ is the immediate parent of $T_i(\mathbf{x})$, and $m(\cdot)$ is the data mass of a tree node. ψ is a normalisation term which is the training data size used to generate T_i .

$s_i(\cdot)$ is in $(0, 1]$. The higher the score the higher the likelihood of \mathbf{x} being an anomaly. Unlike $\ell_i(\mathbf{x})$ in iForest, $s_i(\mathbf{x})$ measures the degree of anomaly locally.



(a) Global anomalies (b) Global anomalies (c) Local anomalies

Fig. 1. Global and Local anomalies. Note that both anomalies a_1 and a_2 are exactly the same instances in Figures (a), (b) and (c). In Fig.(a) and Fig.(b), a_1 and a_2 have low density than that in the normal clusters C_1 and C_2 . In Fig.(c), a_1 , a_2 and the normal cluster C_3 have the same density but a_1 and a_2 are anomalies relative to the normal cluster C_1 with a higher density.

The final anomaly score can be estimated as the average of local anomaly scores over t iTrees as follows:

$$S(\mathbf{x}) = \frac{1}{t} \sum_{i=1}^t s_i(\mathbf{x}) \tag{3}$$

Once every instance in the given data set has been scored, instances can be ranked in descending order of their anomaly scores. The instances at the top of the list are reported as anomalies.

Relation to LOF and DEMass-LOF

The idea of relative mass in ReMass-iForest has some relation to the idea of relative density in Local Outlier Factor (LOF) [4]. LOF uses k nearest neighbours to estimate density $\bar{f}_k(\mathbf{x}) = \frac{|N(\mathbf{x}, k)|}{n \sum_{\mathbf{x}' \in N(\mathbf{x}, k)} \text{distance}(\mathbf{x}, \mathbf{x}')}$ where $N(\mathbf{x}, k)$ is the set of k nearest neighbours of \mathbf{x} . It estimates its anomaly score as the ratio of the average density of \mathbf{x} 's k nearest neighbours to $\bar{f}_k(\mathbf{x})$. In LOF, the local neighbourhood is defined by k nearest neighbours which requires distance calculation. In contrast, in ReMass-iForest, the local neighbourhood is the immediate parent in iTrees. It does not require distance calculation.

DEMass-LOF [5] computes the same anomaly score as LOF from trees, without distance calculation. The idea of relative density of parent and leaf nodes was used in DEMass-LOF. It constructs a forest of t balanced binary trees where the height of each tree is $b \times d$ (b is a parameter that determines the level of division on each attribute and d is the number of attributes). It estimates its anomaly score as the ratio of average density of the parent node to the average density of the leaf node where \mathbf{x} falls. The density of a node is estimated as the ratio of mass to volume. It uses mass to estimate density and ranks instances based on the density ratio. Like iForest, it is fast because no distance calculation is involved. But, it has limitation in dealing problems with even a moderate number of dimensions because each tree has $2^{(b \times d)}$ leaf nodes.

Table 1. Ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF

	ReMass-iForest	iForest	DEMMass-LOF	LOF
Ranking Measure	$\frac{1}{t\psi} \sum_{i=1}^t \frac{m(\tilde{T}_i(\mathbf{x}))}{m(T_i(\mathbf{x}))}$	$\frac{1}{t} \sum_{i=1}^t \ell_i(\mathbf{x})$	$\frac{\sum_{i=1}^t \frac{m(\tilde{T}_i(\mathbf{x}))}{\tilde{v}_i}}{\sum_{i=1}^t \frac{m(T_i(\mathbf{x}))}{v_i}}$	$\frac{\sum_{\mathbf{x}' \in N(\mathbf{x}, k)} \frac{\tilde{f}_k(\mathbf{x}')}{ N(\mathbf{x}, k) }}{\tilde{f}_k(\mathbf{x})}$
Time Complexity	$O(t(n + \psi) \log \psi)$	$O(t(n + \psi) \log \psi)$	$O(t(n + \psi)bd)$	$O(dn^2)$
Space Complexity	$O(t\psi)$	$O(t\psi)$	$O(td\psi)$	$O(dn)$

\tilde{v}_i and v_i are the volumes of nodes $\tilde{T}_i(\mathbf{x})$ and $T_i(\mathbf{x})$, respectively.

In contrast to LOF and DEMass-LOF, ReMass-iForest does not require density estimation, it uses relative mass directly in order to estimate the local anomaly score from each iTree.

The ranking measure and complexities (time and space) of ReMass-iForest, iForest, DEMass-LOF and LOF are provided in Table 1.

2.2 Information Retrieval: ReFeat and ReMass-ReFeat

In this subsection, we first describe how ReFeat uses iForest in IR and its weakness. Then, we introduce a new IR system, ReMass-ReFeat, based on the relative mass to overcome the weakness.

ReFeat

Given a query instance \mathbf{q} , ReFeat [3] assigns a weight $w_i(\mathbf{q}) = \frac{\ell_i(\mathbf{q})}{c} - 1$ (where c is a normalisation constant) to each T_i . The relevance feedback process [6] allows user to refine the retrieved result by providing some ‘relevant’ and ‘irrelevant’ examples for the query. Let $\mathcal{Q} = \mathcal{P} \cup \mathcal{N}$ is a set of feedback instances to the query \mathbf{q} where \mathcal{P} and \mathcal{N} are the sets of positive and negative feedbacks, respectively. Note that \mathcal{P} includes \mathbf{q} . In a relevance feedback round, ReFeat assigns a weight to T_i using positive and negative feedback instances as: $w_i(\mathcal{Q}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{y}^+ \in \mathcal{P}} w_i(\mathbf{y}^+) -$

$\gamma \frac{1}{|\mathcal{N}|} \sum_{\mathbf{y}^- \in \mathcal{N}} w_i(\mathbf{y}^-)$, where $0 \leq \gamma \leq 1$ is a trade-off parameter for the relative contribution of positive and negative feedbacks. The relevance of \mathbf{x} w.r.t \mathcal{Q} is estimated as the weighted average of its path lengths over t iTrees as follows:

$$R_{ReFeat}(\mathbf{x}|\mathcal{Q}) = \frac{1}{t} \sum_{i=1}^t (w_i(\mathcal{Q}) \times \ell_i(\mathbf{x})) \quad (4)$$

Even though ReFeat has been shown to have superior retrieval performance over other existing methods in CBMIR, the ranking scheme does not guarantee

that two instances having similar ranking scores are in the same local neighbourhood. Two instances can have similar score if they have equal path lengths in an iTree even though they lie in two different branches which shares few common nodes. This effect will degrade the performance of ReFeat especially when the tree height (h) is increased. Hence, ReFeat must use a low h (2 or 3) in order to reduce this weakness. The superior performance of ReFeat is mainly due to its large ensemble size ($t = 1000$). We will discuss the effect of h and t in ReFeat in Section 3.2. In a nutshell, ReFeat does not consider the positions of instances in the feature space as it computes the path length in iTrees.

ReMass-ReFeat

In each iTree T_i , the relevance of \mathbf{x} w.r.t. \mathbf{q} , $r_i(\mathbf{x}|\mathbf{q})$, is estimated using relative mass as follows:

$$r_i(\mathbf{x}|\mathbf{q}) = \frac{m(T_i(\mathbf{q}))}{m(T_i(\mathbf{x}, \mathbf{q}))} \quad (5)$$

where $T_i(\mathbf{x}, \mathbf{q})$ is the smallest region in T_i where \mathbf{x} and \mathbf{q} appear together.

In equation 5, the numerator corresponds with $w_i(\mathbf{q})$ in ReFeat. The denominator term measures how relevant \mathbf{x} is to \mathbf{q} . In contrast, ReFeat's $\ell_i(\mathbf{x})$ is independent of \mathbf{q} (it does not examine whether \mathbf{x} and \mathbf{q} are in the same locality [3]); whereas $m(T_i(\mathbf{x}, \mathbf{q}))$ measures how close \mathbf{x} and \mathbf{q} are in the feature space. In each T_i , $r_i(\mathbf{x}|\mathbf{q})$ is in the range of $(0, 1]$. The higher the score the more relevance of \mathbf{x} w.r.t \mathbf{q} . If \mathbf{x} and \mathbf{q} lie in the same leaf node in T_i , $r_i(\mathbf{x}|\mathbf{q})$ is 1. This relevance measure gives a high score to an instance which lies deeper in the branch where \mathbf{q} lies.

The final relevance score of \mathbf{x} w.r.t \mathbf{q} , $R(\mathbf{x}|\mathbf{q})$, is the average over t iTrees:

$$R(\mathbf{x}|\mathbf{q}) = \frac{1}{t} \sum_{i=1}^t r_i(\mathbf{x}|\mathbf{q}) \quad (6)$$

Once the relevance score of each instance is estimated, the scores can be sorted in descending order. The instances at the top of the list are regarded as the most relevant instances to \mathbf{q} .

ReMass-ReFeat estimates the relevance score with relevance feedback as follows:

$$R(\mathbf{x}|\mathcal{Q}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{y}^+ \in \mathcal{P}} R(\mathbf{x}|\mathbf{y}^+) - \gamma \frac{1}{|\mathcal{N}|} \sum_{\mathbf{y}^- \in \mathcal{N}} R(\mathbf{x}|\mathbf{y}^-) \quad (7)$$

Note that equations 5 and 6 do not make use of any distance or similarity measure, and $R(\mathbf{x}|\mathbf{q})$ is not a metric as it does not satisfy all metric axioms. It has the following characteristics. For $\mathbf{x}, \mathbf{y} \in D$,

- i. $0 < R(\mathbf{x}|\mathbf{y}) \leq 1$ (Non-negativity)
- ii. $R(\mathbf{x}|\mathbf{x}) = R(\mathbf{y}|\mathbf{y}) = 1$ (Equal self-similarity; maximal similarity)
- iii. $R(\mathbf{x}|\mathbf{y}) \neq R(\mathbf{y}|\mathbf{x})$ (Asymmetric)

Note that ReMass-ReFeat and ReFeat have the same time complexities. If indices of data instances falling in each node are recorded in the modelling stage, the joint mass of \mathbf{q} and every $\mathbf{x} \in D$ can be estimated in one search from

Table 2. Time and space complexities of ReMass-ReFeat and ReFeat

	ReMass-ReFeat	ReFeat
Time Complexity	$O(t(n + \psi) \log \psi)$ (Model building) $O(t(n + \log \psi))$ (On-line query)	$O(t(n + \psi) \log \psi)$ (Model building) $O(t(n + \log \psi))$ (On-line query)
Space Complexity	$O(t(n + \psi))$	$O(n + t\psi)$

the root to $T_i(\mathbf{q})$ in each tree. But, it will increase the space complexity as it requires to store n indices in each iTree. The time and space complexities of ReMass-ReFeat and ReFeat are provided in Table 2.

3 Empirical Evaluation

In this section, we evaluate the utility of relative mass in AD and CBMIR tasks. In AD, we compared ReMass-iForest with iForest [1], DEMass-LOF [5] and LOF [4]. In CBMIR, we compared ReMass-ReFeat with ReFeat [3] and the other existing CBMIR systems: MRBIR [7], InstRank [8] and Qsim [9]. Both the AD and CBMIR experiments were conducted in unsupervised learning settings. The labels of instances were not used in the model building process. They were used as the ground truth in the evaluation stage. The AD results were measured in terms of the area under ROC curve (AUC). In CBMIR, the precision at the top 50 retrieved results (P@50) [3] was used as the performance measure. The presented result was the average over 20 runs for all randomised algorithms. A two-standard-error significance test was conducted to check whether the difference in performance of two methods was significant.

We used the same MATLAB implementation of iForest provided by the authors of ReFeat [3], the JAVA implementation of DEMass-LOF in the WEKA [10] platform, and the JAVA implementation of LOF in the ELKI [11] platform.

We present the empirical evaluation results in the following two subsections.

3.1 Anomaly Detection: ReMass-iForest versus iForest

In the first experiment, we used a synthetic data set to demonstrate the strength of ReMass-iForest over iForest to detect local anomalies. The data set has 263 normal instances in three clusters and 12 anomalies representing global, local and clustered anomalies. The data distribution is shown in Figure 2a. Instances a_1, a_2 and a_3 are global anomalies; four instances in A_4 and two instances in A_5 are clustered anomalies; and a_6, a_7 and a_8 are local anomalies; C_1, C_2 and C_3 are normal instances in three clusters of varying densities.

Figures 2b-2d show the anomaly scores of all data instances obtained from iForest and ReMass-iForest. With iForest, local anomalies a_6, a_7 and a_8 had lower anomaly scores than some normal instances in C_3 ; and it produced AUC of 0.98. In contrast, ReMass-iForest had ranked local anomalies a_6, a_7, a_8 higher than any instances in normal clusters C_1, C_2 and C_3 along with global anomalies a_1, a_2 and a_3 . But, ReMass-iForest with $MinPts = 1$ had some problem in ranking

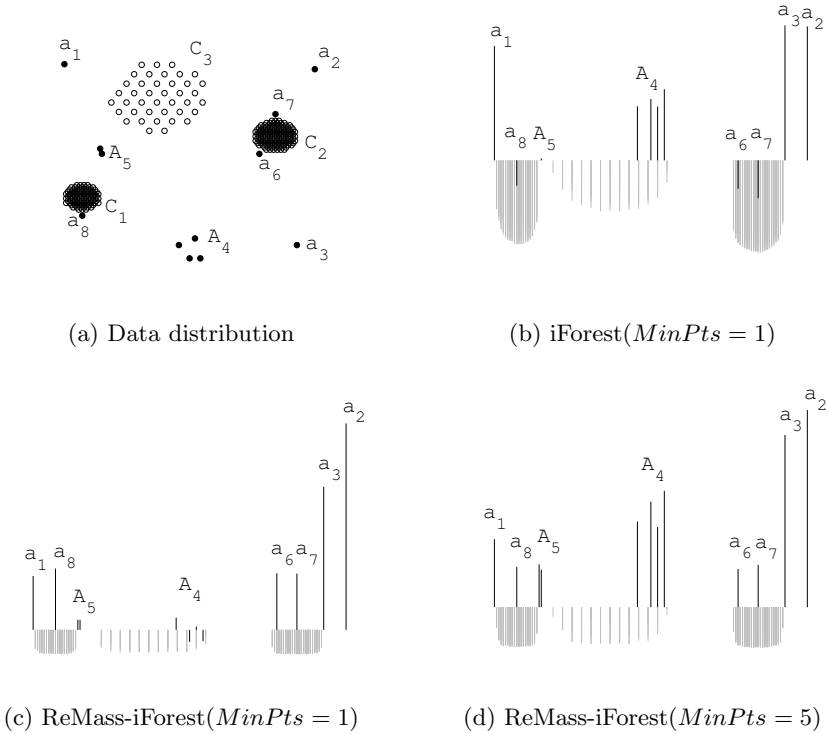


Fig. 2. Anomaly scores by iForest and ReMass-iForest using $t = 100, \psi = 256$. Note that in anomaly score plots, instances are represented by their values on x_1 dimension. Anomalies are represented by black lines and normal instances are represented by gray lines. The height of lines represents the anomaly scores. In order to differentiate the scores of normal and anomaly instances, the maximum score for normal instances is subtracted from the anomaly scores so that all normal instances have score of zero or less.

clustered anomalies in A_4 and produced AUC of 0.99. One fringe instance in the cluster C_3 was ranked higher than two clustered anomalies in A_4 . This is because cluster anomalies have similar mass ratio w.r.t their parents as that for the instances in sparse normal cluster C_3 . Clustered anomalies were correctly ranked and AUC of 1.0 was achieved when $MinPts$ was increased to 5. The performance of iForest did not improve when $MinPts$ was increased to any values in the range (2, 3, 4, 5 and 10).

In the second experiment, we used the ten benchmark data sets previously employed by Liu et al (2008) [1]. In ReMass-iForest, iForest and DEMass-LOF, the parameter t was set to 100 as default and the best value for the sub-sample size ψ was searched from 8, 16, 32, 64, 128 to 256. In ReMass-iForest, $MinPts$ was set to 5 as default. iForest uses the default settings as specified in [1], i.e, $MinPts = 1$. The level of subdivision (b) for each attribute in DEMass-LOF was searched from 1, 2, 3, 4, 5, and 6. In LOF, the best k was searched between 5 and 4000 (or to $\frac{n}{4}$ for small data sets), with steps from 5, 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000 to 4000. The best results were reported.

Table 3. AUC and runtime (seconds) of ReMass-iForest (RM), iForest (IF), DEMass-LOF (DM), and LOF in benchmark datasets

Data set	n	d	AUC				Runtime			
			RM	IF	DM	LOF	RM	IF	DM	LOF
Http	567K	3	1.00	1.00	0.99	1.00	71	99	19	19965
ForestCover	286K	10	0.96	0.88	0.87	0.94	42	56	4	2918
Mulcross	262K	4	1.00	1.00	0.99	1.00	20	23	16	2169
Smtpt	95K	3	0.88	0.88	0.78	0.95	10	12	16	373
Shuttle	49K	9	1.00	1.00	0.95	0.98	4	9	7	656
Mammography	11K	6	0.86	0.86	0.86	0.68	1	1	5	127
Satellite	6K	36	0.71	0.70	0.55	0.79	1	4	0.6	24
Breastw	683	9	0.99	0.99	0.98	0.96	0.1	0.4	0.3	0.4
Arrhythmia	452	274	0.80	0.81	0.52	0.80	0.3	0.5	5	1
Ionosphere	351	32	0.89	0.85	0.85	0.90	2	3	0.5	0.3

The characteristics of the data sets, AUC and runtime (seconds) of ReMass-iForest, iForest, DEMass-LOF and LOF are presented in Table 3.

In terms of AUC, ReMass-iForest had better or at least similar results to iForest. Based on the two-standard-error significance test, it produced better results than iForest in the ForestCover and Ionosphere data sets. Most of these datasets do not have local anomalies. So, both methods had similar AUC in eight data sets. Note that iForest did not improve AUC when $MinPts$ was set to 5. ReMass-iForest had produced significantly better AUC than DEMass-LOF in relatively high dimensional data sets (Arrhythmia - 274, Satellite - 36, Ionosphere - 32, ForestCover - 10, Shuttle - 9). These results show that DEMass-LOF has problem in handling data sets with a moderate number of dimensions (9 or 10). ReMass-iForest was competitive to LOF. It was better than LOF in the Mammography data set, worse in the Smtpt and Satellite data sets, and equal performance in the other seven data sets.

As shown in Table 3, the runtime of ReMass-iForest, iForest and DEMass-LOF were of the same order of magnitude whereas LOF was upto three order of magnitude slower in large data sets. Note that we can not conduct a head-to-head comparison of runtime of ReMass-iForest and iForest with DEMass-LOF and LOF because they were implemented in different platforms (MATLAB versus JAVA). The results are included here just to provide an idea about the order of magnitude of runtime. The difference in runtime of ReMass-iForest and iForest was due to the difference in ψ and $MinPts$. $MinPts = 5$ results in smaller size iTrees in ReMass-iForest than those in iForest ($MinPts = 1$). Hence, ReMass-iForest runs faster than iForest even though the same ψ is used.

3.2 CBMIR: ReMass-ReFeat versus ReFeat

The performance of ReMass-ReFeat was evaluated against that of ReFeat in music and image retrieval tasks with GTZAN music data set [12] and COREL image data set [13], respectively. GTZAN is a data set of 1000 songs uniformly distributed in 10 genres. Each song is represented by 230 features. COREL is a data set of 10,000 images uniformly distributed over 100 categories. Each image

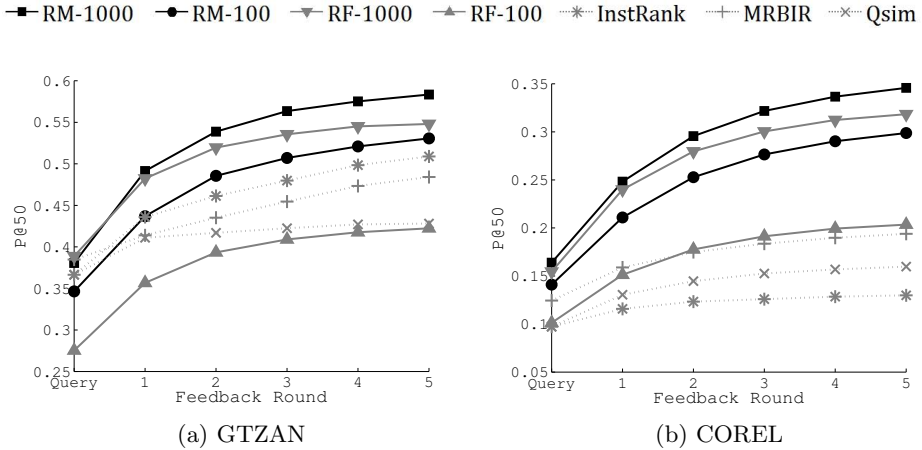


Fig. 3. Precision at top 50 returned results (P@50)

is represented by 67 features. These are the same data sets used in [3] to evaluate the performance of ReFeat. The results of the existing CBMIR systems InstRank, Qsim and MRBIR were taken from [3].

We conducted our experiments using the same experimental design as in [3]. Initially five queries were chosen randomly from each class. For each query, instances from the same class were regarded as relevant and the other classes were irrelevant. At each round of feedback, two relevant (instances from the same class) and two irrelevant (instances from the other classes) instances were provided. Upto five rounds of feedback were conducted for each query. The instance was not used in ranking if it was used as a feedback instance. The feedback process was repeated five times with different relevant and irrelevant feedbacks. The above process was repeated 20 times and average P@50 was reported.

In ReMass-ReFeat, the parameters ψ and $MinPts$ were set as default to 256 and 1, respectively. In ReFeat, ψ was set to 4 for GTZAN and 8 for COREL as reported in [3]. Other settings of ψ in ReFeat were found to perform worse than these settings. In order to show how their retrieval performance varies when ensemble size was increased, we used two settings for t : ReMass-ReFeat and ReFeat with (i) $t = 100$ (RM-100 and RF-100) and (ii) $t = 1000$ (RM-1000 and RF-1000). The feedback parameter γ was set as default to 0.5 in ReMass-ReFeat and 0.25 in ReFeat (as used in [3]).

P@50 of ReMass-ReFeat (RM-100 and RM-1000), ReFeat (RF-100 and RF-1000), InstRank, MRBIR and Qsim in the GTZAN and COREL data sets are shown in Figure 3. P@50 curves in both the data sets show that ReMass-ReFeat (RM-1000) has better retrieval performance than all contenders, especially in feedback rounds. In round 1 or no feedback (query only), ReMass-ReFeat (RM-1000) and ReFeat (RF-1000) produced similar retrieval performance but in latter feedback rounds, RM-1000 produced better results than RF-1000.

It is interesting to note that the performance of RF-100 was worse than that of RM-100 in all feedback rounds including query only (no feedback). In GTZAN,

RF-100 had worst performance than all contenders. The increase in P@50 from RF-100 to RF-1000 was a lot larger than that of RM-100 to RM-1000. This result shows that the retrieval performance of ReFeat is mainly due to the large ensemble size of 1000. The difference in P@50 of RM-100 and RF-1000 was decreasing in subsequent feedback rounds. This indicates that ReMass-ReFeat produces better result than ReFeat even with a smaller ensemble size if more feedback instances are available.

In terms of runtime, ReMass-ReFeat had slightly higher runtime than ReFeat because of the higher ψ that allows trees to grow deeper (256 vs. 4 in GTZAN and 8 in COREL). The model building time of RM-1000 was 21 seconds (vs. 4 seconds of RF-1000) in COREL and 20 seconds (vs. 2 seconds of RF-1000) in GTZAN. The on-line retrieval time for one query of RM-1000 was 0.9 seconds (vs. 0.3 seconds of RF-1000) in COREL and 0.2 seconds (vs. 0.2 seconds of RF-1000) in GTZAN.

Figure 4 shows the effect of ψ on the P@50 of ReMass-ReFeat and ReFeat at feedback round 5 (one run) in the GTZAN data set. In ReFeat, when ψ was increased above 4, the retrieval performance degraded. This is due to the increase in the height of iTrees ($h = \log_2(\psi)$) and instances falling in two distinct branches having similar relevance score based on the same path lengths. In contrast, ReMass-ReFeat improved its retrieval performance up to 64 and then remained almost flat beyond that. Similar effect was observed in the COREL data set where the performance of ReFeat degraded when ψ was set above 8.

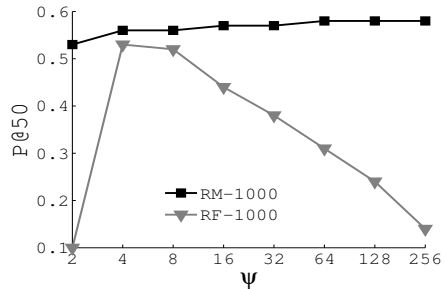


Fig. 4. P@50 at feedback round 5 with varying sample size (ψ) in the GTZAN data set

4 Conclusions

While the relative mass was motivated to overcome the weakness of iForest in detecting local anomalies, we have shown that the idea has a wider application. In information retrieval, we apply it to overcome the weakness of a state-of-the-art system called ReFeat. Our empirical evaluations show that ReMass-iForest and ReMass-ReFeat perform better than iForest and ReFeat, respectively, in terms of task-specific performance. In comparison with other state-of-the-art systems in both tasks, ReMass-iForest and ReMass-ReFeat are found to be either competitive or better.

The idea of relative mass in ReMass-iForest is similar to that of relative density in LOF and our empirical results show that ReMass-iForest and LOF have similar anomaly detection performance. However, ReMass-iForest runs significantly faster than LOF in large data sets because it does not require distance or density calculations.

Acknowledgement. This work is partially supported by the U.S. Air Force Research Laboratory, under agreement#FA2386-13-1-4043. Sunil Aryal is supported by Australian Postgraduate Award (APA), Monash University. The paper on mass-based similarity measure [14] has inspired us in creating the relevance score based on relative mass used in ReMass-ReFeat; though the motivations of the two papers differ.

References

1. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: Proceedings of the Eighth IEEE International Conference on Data Mining, pp. 413–422 (2008)
2. Ting, K.M., Zhou, G.T., Liu, F.T., Tan, S.C.: Mass estimation. *Machine Learning* 90(1), 127–160 (2013)
3. Zhou, G.T., Ting, K.M., Liu, F.T., Yin, Y.: Relevance feature mapping for content-based multimedia information retrieval. *Pattern Recognition* 45(4), 1707–1720 (2012)
4. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying Density-Based Local Outliers. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
5. Ting, K., Washio, T., Wells, J., Liu, F., Aryal, S.: DEMass: a new density estimator for big data. *Knowledge and Information Systems* 35(3), 493–524 (2013)
6. Rui, Y., Huang, T., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* 8(5), 644–655 (1998)
7. He, J., Li, M., Zhang, H.J., Tong, H., Zhang, C.: Manifold-ranking based image retrieval. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 9–16. ACM, New York (2004)
8. Giacinto, G., Roli, F.: Instance-based relevance feedback for image retrieval. In: *Advances in Neural Information Processing Systems*, vol. 17, pp. 489–496 (2005)
9. Zhou, Z.H., Dai, H.B.: Query-sensitive similarity measure for content-based image retrieval. In: Proceedings of the Sixth International Conference on Data Mining, pp. 1211–1215 (2006)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009)
11. Achtert, E., Hettab, A., Kriegel, H.-P., Schubert, E., Zimek, A.: Spatial outlier detection: Data, algorithms, visualizations. In: Pfoser, D., Tao, Y., Mouratidis, K., Nascimento, M.A., Mokbel, M., Shekhar, S., Huang, Y. (eds.) *SSTD 2011*. LNCS, vol. 6849, pp. 512–516. Springer, Heidelberg (2011)
12. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10(5), 293–302 (2002)
13. Zhou, Z.H., Chen, K.J., Dai, H.B.: Enhancing relevance feedback in image retrieval using unlabeled data. *ACM Transactions on Information Systems* 24(2), 219–244 (2006)
14. Ting, K.M., Fernando, T.L., Webb, G.I.: Mass-based Similarity Measure: An Effective Alternative to Distance-based Similarity Measures. Technical Report 2013/276, Calyton School of IT, Monash University, Australia (2013)