Identifying Recurrent and Unknown Performance Issues based on Hidden Markov Random Field Model

- The Companion Page

(September 2013)

This is the companion page for our paper entitled "Identifying Recurrent and Unknown Performance Issues based on Hidden Markov Random Field Model". This page gives more details about the proposed HMRF-based approach. If you have any comments or questions regarding this research work, please feel free to contact Dr Jian-Guang Lou at: jlou@microsoft.com.

1. More Details about the HMRF-based Diagnostic Approach

We focus on deriving a clustering objective function in terms of discretization thresholds and cluster representatives from the posterior probability of a HMRF model. Suppose that there exist v instances of performance issue where each of which belongs to one of the k issue types for $k \leq v$. The v issue instances and the non-issue instances (e.g. compliance instances) together form $v+1$ sets of *temporal-neighboring* constraints. Being bound by such $v+1$ sets of constraints, the T records will eventually be grouped into $k+1$ clusters to indicate which of the performance issues are similar and which are not. Note that one of the $k+1$ clusters is reserved for the non-issue instances.

The issue types can be modeled as a *hidden Markov random field* $L = \{L_t, t \in \daleth\}$, where $\daleth = \{1,2,\ldots,T\}$ denotes a set of time-epoch indices. The hidden variable L_t represents the issue type (0 for non-issue and 1~k for k issues) corresponding to each record at the t-th epoch. The hidden variables are mutually related via a neighborhood system $\mathcal{N} = \{\bigcup_{p\in\{0,1,\ldots,v\}} \mathcal{N}_p\}$ where $\mathcal{N}_p = \left\{t_{p_1},\ldots,t_{p_y},\ldots\right\} \subset \daleth$ denotes the p-th non-overlapping neighborhood such that $t_{p_y} \notin \mathcal{N}_{\{0,1,\ldots,v\}\backslash\{p\}}$.

On the other hand, each metric component $x_{t,i}$ is discretized into a binary value $\bar{x}_{t,i}$ according to a metric threshold τ_i (an unknown parameter to be identified through optimization in Section IV). If $x_{t,i}$ has a value larger (resp. not larger) than τ_i, then $\bar{x}_{t,i} = 1$ (resp. 0), signifying that $x_{t,i}$ is more likely to come from the distribution of values associated with SLO violation (resp. SLO compliance). The discretization can be written as

$$\bar{\boldsymbol{x}}_{t,i} = \boldsymbol{U}(\boldsymbol{x}_{t,i} - \boldsymbol{\tau}_i) = \begin{cases} \mathbf{0} & \mathbf{if}(\boldsymbol{x}_{t,i} - \boldsymbol{\tau}_i) \leq \mathbf{0} \\ \mathbf{1} & \mathbf{if}(\boldsymbol{x}_{t,i} - \boldsymbol{\tau}_i) > \mathbf{0} \end{cases} \quad (1)$$

The discretized metric vectors can then be modeled as an *observable random field* $\bar{X} = \{\bar{X}_t, t \in \daleth\}$, in which each metric vector X_t follows a conditional probability distribution $p(\bar{\boldsymbol{x}}_t|l_t)$ determined by its underlying violation state l_t.

Let $\boldsymbol{\ell}$ be a configuration of L and $\boldsymbol{\chi}$ be a metric configuration of $\bar{X}$, the posterior probability of configuration $\boldsymbol{\ell}$ can be obtained as

$$\boldsymbol{P}(\ell|\chi) = \frac{\boldsymbol{P}(\chi|\ell)\, \boldsymbol{P}(\ell)}{\boldsymbol{P}(\chi)} \quad \mathbf{(2)}$$

1) Likelihood $P(\boldsymbol{\chi}|\boldsymbol{\ell})$

By assuming that the probability density takes an exponential form, the conditional probability of $\boldsymbol{\chi}$ given $\boldsymbol{\ell}$ can hence be expressed as

$$\boldsymbol{P}(\chi|\ell) = \boldsymbol{p}\left(\chi, \{\bar{\mu}_l\}_{l=0}^{k}\right) = \frac{\mathbf{1}}{\boldsymbol{Z_2}} \prod_{t\in\mathbb{T}} \boldsymbol{e}^{-\boldsymbol{D}(\bar{x}_t, \bar{\mu}_{l_t})} \quad \textbf{(3)}$$

Note that the conditional probability in (**3**) can be described in terms of a distortion measure D and a normalization constant Z_2. More specifically, the distortion measure quantifies the intra-cluster distance between $\bar{\boldsymbol{x}}_t$ and $\bar{\boldsymbol{\mu}}_{l_t}$ in a M-dimensional Hamming space such that

$$\boldsymbol{D}\left(\bar{x}_t, \bar{\mu}_{l_t}\right) = \bar{x}_t \oplus \bar{\mu}_{l_t} \quad \textbf{(4)}$$

where $\oplus$ denotes the XOR operator. Note that since the records are represented in binary, the distortion is measured in terms of Hamming distance. In our scenario, the way to minimize the distortion is to optimize the underlying discretization thresholds that have a direct influence on the binary records $\bar{\boldsymbol{x}}_t$.

2) Prior probability $P(\boldsymbol{\ell})$

According to Hammersley-Clifford theorem, the prior probability of $\boldsymbol{\ell}$ in a HMRF can be written as

$$\boldsymbol{P}(\ell) = \frac{\mathbf{1}}{\boldsymbol{Z_1}} \boldsymbol{e}^{-\boldsymbol{V}(\ell)} = \frac{\mathbf{1}}{\boldsymbol{Z_1}} \boldsymbol{e}^{-\sum_{\mathcal{N}_p \in \mathcal{N}} \boldsymbol{V}_{\mathcal{N}_p}(\ell)} \quad (5)$$

where Z_1 is a normalizing constant, $V(\boldsymbol{\ell})$ denotes the configuration potential function that can be expressed as a sum of potentials $V_{\mathcal{N}_p}(\boldsymbol{\ell})$ over $v+1$ neighborhoods corresponding to v performance issue instances and a non-issue instance in the violation-state configuration $\boldsymbol{\ell}$. We restrict the MRF over the hidden variable to have pairwise potentials by levying a *temporal-neighboring* constraint on every possible pair of discretized records within each neighborhood $\mathcal{N}_p$, leading to $\binom{n_p}{2}$ constraints with n_p denoting the number of elements in $\mathcal{N}_p$. With this, $V_{\mathcal{N}_p}(\boldsymbol{\ell})$ can be defined as

$$\boldsymbol{V}_{\mathcal{N}_p}(\ell) = \boldsymbol{w_p} \sum\nolimits_{\substack{\forall t,t' \in \mathcal{N}_p, \\ t \neq t'}} \mathbb{I}[\boldsymbol{l_t} \neq \boldsymbol{l_{t'}}] \quad (6)$$

where $\mathbb{I}$ denotes an indicator function ($\mathbb{I}[\text{true}] = 1$, $\mathbb{I}[\text{false}] = 0$) and w_p denotes the normalizing weight for the total violations of *temporal-neighboring* constraint in the p-th neighborhood. The potential $V_{\mathcal{N}_p}$ in (6) acts as a penalty function that punishes poor clustering configuration and shapes towards an ideal cluster formation where instances from a neighborhood are perfectly enclosed within one of the $k+1$ clusters.

Note that the total *temporal-neighboring* constraints within every neighborhood are likely to be different, since the quantity of records in each neighborhood is not equal. For instance, as violation instances are relatively uncommon, each compliance record in $\mathcal{N}_0$ is bound with a higher number of constraints than the violation records. Consider that a cut has to be applied to separate a minority of records from a neighborhood (i.e., into two clusters) during the clustering optimization process. In such a case, small violation neighborhoods (short performance issues) are always preferred as it incurs a lower count of constraint violations. To avoid such an unfair selection, a normalization factor w_p is added for normalizing $V_{\mathcal{N}_p}(\boldsymbol{\ell})$ to the range of $[0,1]$. Let $n_{p,l}$ be the number of elements of the p-th neighborhood in the l-th cluster, $|\mathcal{N}_p| = \sum_{l=0}^{k} n_{p,l}$ be the cardinality of the p-th neighborhood, $\lambda_p = \sum_{l=0}^{k} \mathbb{I}\left[n_{p,l} \neq 0\right]$ be the number of clusters that contain at least an element of p-th neighborhood, and $r_p = \frac{|\mathcal{N}_p|}{\lambda_p}$ be the number of the p-th neighborhood elements being divided equally to each of the λ_p clusters. Finally, w_p can be described as

$$\boldsymbol{w_p} = \frac{\boldsymbol{\lambda_p}}{\binom{\lambda_p}{2}\boldsymbol{r_p^2}}. \qquad (7)$$

Although two or more non-adjacent performance issue instances (usually separated by long compliance phases) might belong to the same type of performance issue (e.g., caused by the same type of reason), we do not bind the records with *temporal-neighboring* constraints across these issues. This is because we do not have sufficient information to differentiate similar performance issues from different issues under such an unsupervised scenario. On the other hand, compliance instances in the training set, regardless of whether they are connected in time, are annotated with a single value indicating that the system is healthy.

3) The objective function

To obtain the maximum-a-posteriori configuration of HMRF, we minimize the posterior energy of HMRF (negative logarithm of $P(\boldsymbol{\ell}|\boldsymbol{\chi})$) through optimizing the M discretization thresholds $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^{M}$ and the $k+1$ cluster representatives $\{\overline{\boldsymbol{\mu}}_{\boldsymbol{l}}\}_{l=0}^{k}$.In accordance with (3) and (5), the objective function can be expressed as

$$\boldsymbol{J_{obj}} = -\log\left(\frac{\mathbf{1}}{\boldsymbol{Z}}\right) + \sum\nolimits_{t\in\daleth} \boldsymbol{D}(\bar{x}_t, \bar{\mu}_{l_t}) + \sum_{\boldsymbol{\mathcal{N}_p}\in\boldsymbol{\mathcal{N}}} \boldsymbol{V}_{\boldsymbol{\mathcal{N}_p}}(\ell). \qquad \textbf{(8)}$$

Note that $\frac{1}{Z} = \frac{1}{Z_1 Z_2 Z_3}$ is a constant. The second factor (D component) is resulted from the probability of generating the discretized records based on the corresponding conditional probability distribution, which are parameterized by the cluster representatives and the discretization thresholds. The minimization of D component encourages binary representation of the discretized metric instances $\overline{\boldsymbol{x}}_{\boldsymbol{t}}$ to be driven towards the corresponding binary cluster representative $\overline{\boldsymbol{\mu}}_{\boldsymbol{l_t}}$ via optimizing the discretization thresholds. The minimization of the third factor V component optimizes the cluster formation characterized by the representatives $\overline{\boldsymbol{\mu}}_{\boldsymbol{l_t}}$and the cluster assignment of the records to satisfy most of the *temporal-neighboring* constraints.

To prevent J_{obj} being dominated by either D or V component, two normalizing constants are introduced: $w_D = \frac{1}{\sum_{t\in\daleth} \overline{\boldsymbol{x}}_{\boldsymbol{t}} \oplus \overline{\boldsymbol{\mu}}_{\mathbf{0}}}$ and $\mathrm{w}_V = \frac{1}{v}$ for$D$ and V components respectively to normalize them to a similar range: $[0, 1]$. Substituting (1), (4), (6) and (7) into (8), the minimization of J_{obj} eventually becomes

$$\underset{\tau=\{\tau_i\}_{i=1}^{M}, \{\bar{\mu}_l\}_{l=0}^{k}}{\mathbf{arg\,min}} \left(\frac{\mathbf{1}}{\sum_{t\in\daleth} \bar{x}_t \oplus \bar{\mu}_0} \sum\nolimits_{t\in\daleth} \boldsymbol{U}(x_t - \tau) \oplus \boldsymbol{U}(\mu_{l_t} - \tau) \right.$$

$$\left. + \frac{\mathbf{1}}{\boldsymbol{v}} \sum_{\boldsymbol{\mathcal{N}_p}\in\boldsymbol{\mathcal{N}}} \frac{\boldsymbol{\lambda_p}}{\binom{\lambda_p}{2}\boldsymbol{r_p^2}} \sum\nolimits_{\substack{\forall t,t'\in\mathcal{N}_p,\\ t\neq t'}} \mathbb{I}[\boldsymbol{l_t} \neq \boldsymbol{l_{t'}}] \right). \qquad \textbf{(9)}$$

2. More Details about the Threshold and Clustering Optimization

The minimization of J_{obj} can be solved with an iterative EM-like approach. As binary data is in concern, it does not make much sense to compute the mean of the binary data within each cluster to obtain the corresponding cluster representative. Therefore, our algorithm adopts a variant of mean, called *medoid* **Error! Reference source not found.**that is one of the data points belonging to the dataset, as a representative of a cluster. Our algorithm shown in Figure 1 begins with an initialization of metric thresholds and medoids. Given the records being discretized in accordance with the initial threshold configuration, the first stage (**Step_A)** searches for a decent clustering configuration that reduces the objective value with a k-*medoid***Error! Reference source not found.** like process. In the second stage, **Step_B** then seeks for a good threshold for a particular metric based on the resultant clustering configuration. These two stages of the algorithm re-iterate one after another until a pre-specified maximum number of iterations is reached. Ultimately, the records will be categorized into $k+1$ clusters corresponding to a compliance and k issue types. At the same time, the optimal threshold for each metric can also be obtained during the iterative optimization. Based on this, our approach extends the original HMRF model to find metric discretization thresholds along with cluster parameters.

Initialization. Good initializations of both discretization thresholds and medoids are important in producing satisfactory outcome. The value of initial metric thresholds should not be fixed too low or too high, as this would lead to too few data points being considered for selection of initial medoids. That is, the extreme case where almost all violation records being represented by a common binary representation should be avoided. The detailed procedures are given in Figure 1.

As for the medoids, the selection of initial values is repeated N_{init} times and the cluster configuration associated with the lowest D component of $J_{obj}$is chosen. Our intent is to seek for a configuration that achieves minimum intra-cluster distance to begin with, although such a configuration might violate most of the *temporal-neighboring* constraints. In fact, this initialization leads to a cluster configuration that is inclined to capture data points uniformly, while getting the medoids well separated at the same time. Note that among the $k+1$ medoids, the binary medoid of the compliance cluster is set as an all-zero binary string, while the k remaining medoids are chosen from the v performance issues instances that are associated with SLO violation for the evaluation of D component.

Optimization. Based on the data $\left\{\bar{\boldsymbol{x}}_{\boldsymbol{t}}^{(q-1)}\right\}_{t=1}^{T}$ discretized using metric thresholds from the previous iteration, **Step_A** evaluates every clustering configuration $\{\bar{\boldsymbol{\mu}}_{\boldsymbol{l}}\}_{l=1}^{k}$ by switching a medoid with every non-medoid and repeatedly reassigning the labels of the non-medoids affected by such a switch. This process is repeated for all clusters until J_{obj} converges to a local minimum. On the other hand, **Step_B** focuses on selecting a good threshold from a set of threshold candidates for a given metric (say the i-th metric) by taking distortion with respect to the single-dimensional medoids $\left\{\bar{\mu}_{l,i}^{(q)}\right\}_{l=0}^{k}$ (single-dimensional D component of J_{obj}) as a goodness measure. These threshold candidates are determined by sorting the observed metric values and computing the average value of every adjacent pair of sorted values. Every update of a single metric discretization threshold in **Step_B** is followed by a search of better medoids in **Step_A**. Both steps are re-iterated until all metrics are optimized $N_{EM(B)}$ times.

Input: A set of contiguous records $\{x_t\}_{t=1}^{T}$, number of estimated clusters k, neighborhood annotation of the records $\{a_t\}_{t=1}^{T} \in \{0,1,\dots,v\}$, medoid of the compliance cluster $\bar{\mu}_0^{(q=0)} \longleftarrow \{0 \dots 00\}$, iteration variables N_{init} and $N_{EM(B)}$.

Output: k non-overlapping partitions and M metric discretization thresholds such that the objective function J_{obj} in (9) is minimal.

1. Initialization: $i = q = 0$. Iterate steps (b) and (c) for N_{init} times and select threshold $\tau^{\star}$ and clustering configurations $\{\bar{\mu}_l^{\star}\}_{l=1}^{k}$ associated with $D^{\star}$ (the lowest D component of J_{obj} in (9)) and set $\tau^{(q=0)} = \tau^{\star}$, $\left\{\bar{\mu}_l^{(q=0)}\right\}_{l=1}^{k} = \{\bar{\mu}_l^{\star}\}_{l=1}^{k}$ and $J_{obj}^{(q=0)} = D^{\star} + V$.
2. select M initial metric thresholds $\tau = \left\{\tau_j\right\}_{j=1}^{M}$ and discretize the records $\{x_t\}_{t=1}^{T} \Longrightarrow \{\bar{x}_t\}_{t=1}^{T}$.
   a) initialize k unique medoids $\{\bar{\mu}_l\}_{l=1}^{k} \in \{x_t\}_{t=1}^{T}$(each with different annotation a) and assign cluster labels to the non-medoids.
   b) compute D component of J_{obj}.
3. Step_A: $q \leftarrow q + 1$. Given $\left\{\bar{x}_t^{(q-1)}\right\}_{t=1}^{T}$, repeat steps (d) to (f) for every non-medoid and every cluster l (except $l = 0$) until convergence. Find the best clustering configuration with $\{\bar{\mu}_l^{\star}\}_{l=1}^{k}$ that gives the lowest $J_{obj}^{\star}$ and let $\left\{\bar{\mu}_l^{(q)}\right\}_{l=1}^{k} \leftarrow \{\bar{\mu}_l^{\star}\}_{l=1}^{k}andJ_{obj}^{(q)} \leftarrow J_{obj}^{\star}$.
   c) swap $\bar{\mu}_l$ with a non-medoid in a cluster l.
   d) assign cluster labels to the non-medoids.
   e) compute J_{obj}.
4. Step_B: $i \leftarrow i + 1 \pmod{M}$. Given $\left\{\bar{\mu}_l^{(q)}\right\}_{l=0}^{k}$, repeat steps (g) and (h) over all the available threshold candidates for the i-th metric. Find the best threshold configuration $\tau_i^{\star}$ for the i-th metric that gives the lowest i-th single dimensional D component of J_{obj} and let $\tau_i^{(q)} \leftarrow \tau_i^{\star}$ and $\left\{\bar{x}_{t,i}^{(q-1)}\right\}_{t=1}^{T} = \left\{\bar{x}_{t,i}^{\star} \middle| \tau_i^{\star}\right\}_{t=1}^{T}$.
   f) swap $\tau_i^{(q)}$ with a threshold candidate and re-discretize $\{x_{t,i}\}_{t=1}^{T}$.
   g) compute the i-th single dimensional D component of J_{obj}.
5. Reiterate Step_A and Step_B for $N_{EM(B)} \times M$ times and output the best clustering and threshold configuration that achieves the lowest J_{obj}.

Figure 1. HMRF-kMedoid-EM Algorithm

It is worth to note that threshold optimization in **Step_B** is carried out a metric at a time instead of considering all the metrics at once. This is because the latter may incur massive combinations of candidate for consideration that would eventually render the search of the optimum threshold configuration exhaustively infeasible. Note also that since V component of J_{obj} is not considered in **Step_B**, the selected threshold for a given metric might violate additional constraints and cause J_{obj} to increase in many iterations. However, these interim increases of J_{obj} are essential to avoid the solution being stuck in a poor local minimum. We found that it is a necessary step to obtain a better configuration that achieves a much lower value of J_{obj} at a later time.

Exploitation of Supervised Data. Recall that each record is associated with a known SLO state (i.e. violation or compliance). Different from the existing approaches, we wisely exploit this knowledge to make our optimization robust and efficient. First, in the computation of D component of J_{obj}, all the compliance data points should be compared against the medoid of the compliance cluster $\bar{\boldsymbol{\mu}}_{\mathbf{0}}$, even if they fall within a violation cluster in that specific iteration. Similarly, the intra-cluster distance of the violation data points should be computed based on the nearest medoid of violation clusters and should not be computed based on the compliance cluster medoid within which they might fall during the optimization process. Second, in the context of system diagnosis, the value of a metric measurement within the compliance epochs is usually treated as a normal value (See (1)). Therefore, the binary medoid of the compliance cluster is fixed (neither initialized, nor shifted) to an all-zero binary string throughout the optimization process. This helps the values of thresholds to not be trapped into local minimums (e.g., the extreme low threshold values).

Convergence Rate. Figure 2 exemplifies the convergence rate of HMRF-kMedoid-EM algorithm in one of the 4-fold cross-validated experiment conducted on synthetic dataset where the details of such a dataset can be found in Section 5.1. In this fold of the experiment, there are $M = 62$ metrics whose measurement values are not constant and therefore the measurements of such 62 metrics were considered for the optimization of discretization thresholds.

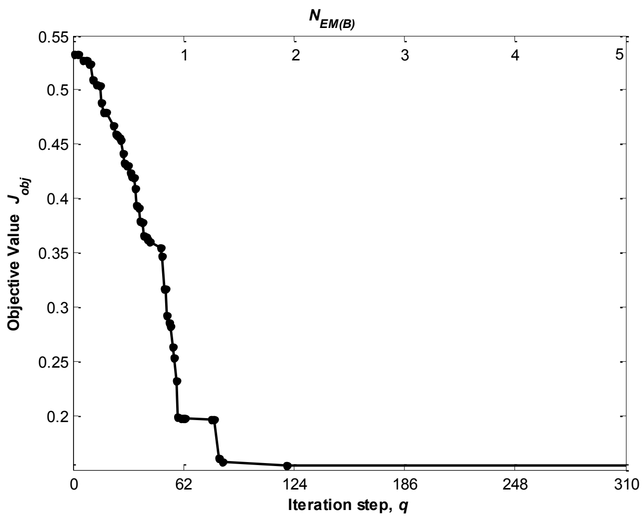

Figure 2. Convergence rate of HMRF-kMediod-EM

In Figure 2, each objective value corresponds to the computed value of J_{obj} upon **Step_B**. Each value of $N_{EM(B)}$ indicates the number of *rounds* where the threshold optimization of all 62 metrics has been completed. It is observed that our algorithm takes 120 iteration steps (or approximately two rounds) to converge. The objective value decreases drastically in the first round of threshold optimization and the decrement occurs rather progressively via gradual refinement of metric thresholds during the second round of optimization. As a result, the convergence happens slightly above 0.15 during the second round of optimization. The threshold and cluster configurations associated with the lowest objective value at $q = 120$ are selected as the optimum configurations.

3. Diagnosis with Clustering Results

The medoids are derived from the objective function and may not be appropriate for realizing all our diagnosis objectives (metric attribution, recurrent-issue association, and unknown-issue detection). This is because the objective function of our algorithm does not depend merely on the intra-cluster distance.In fact, the medoids are used to define the best partitions that satisfy most of the *temporal-neighboring*constraints and minimize the data distortion concurrently. In addition, the position of a medoid could simply be adjusted with respect to the adjacent medoids to induce the desired boundary and thus may not best characterize the data distribution within such a cluster. To realize the diagnostic tasks more effectively, we need to rely on another pure distance-based cluster representative known as *distrep*. Similar to medoid, distrep is one of the data points that has a minimum sum of distance from all other points in a cluster. Considering Q points being captured by a cluster, the index of a data point for distrep assignment can be identified as follows:

$$\boldsymbol{c^{\star}} = \underset{c=\{1,\ldots,Q\}}{\mathbf{arg\,min}} \sum_{\boldsymbol{q=1}}^{\boldsymbol{Q}} \left(\bar{x}_q \oplus \bar{x}_c\right) \qquad \mathbf{(10)}$$

With this, $\bar{\boldsymbol{x}}_{c^{\star}}$is assigned as the distrep of the cluster. This is applied to all the clusters for distrep assignment. Distrep appears to be a better cluster representative than medoid due to: (1) it is more robust to outliers in the cluster; (2) it is selected solely based on the distribution of the data points within the cluster and is independent of data points in the adjacent clusters.

Metric Attribution. Since distrep is a better cluster representative than medoid, we identify the attributed metrics from the encoded bits of each cluster distrep for representing the symptom of the corresponding issue. Intuitively, the binary '1's in each distrep reflect the attributed metrics relevant to the corresponding issue, since the discretization of distrep is optimized with respect to the intra-cluster distance and the *temporal-neighboring* constraints. The root cause of each type of issue can often be identified through analyzing these attributed metrics.

Recurrent-Issue Association and Unknown-Issue Detection. Given a query record (unknown testing record), it would be desirable if past similar records can be associated accurately so that past diagnoses can be leveraged and appropriate repairs can be identified. On top of this, a good diagnostic approach should be capable of detecting unknown violation issues when the query records and all previous records are sufficiently dissimilar. Once the optimized thresholds and medoids are obtained from the HMRF-kMedoid-EM algorithm, the distrep assignment is performed according to (10). A Hamming similarity threshold $\tau_{sim}$is then defined with reference to each distrep as a similarity measure. With the discretization thresholds, medoids, distreps, and similarity thresholds, the recurrent-issue association and unknown-issue detection can be performed as follows: First, the query records of a performance issue are discretized using the optimized threshold configuration. The records are then associated with the nearest medoid and annotated with the corresponding cluster index. Based on the distrep of the same cluster, a query issue is deemed belonging to the issue type that the distrep represents if majority of the discretized records within the issue have a distance not larger than τ_{sim}as compared to the distrep. Otherwise, if most records give larger distance than τ_{sim}, the issue is tagged as unknown.

--- End ---