Time Series Discretization via MDL-based Histogram Density Estimation

Yoshitaka Kameya
Graduate School for Information Science and Engineering, Tokyo Institute of Technology
Ookayama 2-12-1, Meguro-ku, Tokyo 152-8552, Japan
Email: kameya@mi.cs.titech.ac.jp

***Abstract*—In knowledge discovery from real-valued time series, discretization is often a key preprocessing that extends the applicability of sophisticated tools for symbolic data mining or logic-based machine learning. For finding meaningful discrete values that can be directly translated into some intuitive symbols, this paper proposes a novel discretization method based on density estimation using a two-dimensional (measurement vs. time) histogram of variable-width bins. We extend Kontkanen and Myllymäki's histogram construction method into our two-dimensional case, keeping the efficiency brought by dynamic programming. Experimental results with artificial and real datasets show the robustness and the usefulness of the proposed method.**

***Keywords*-discretization; histogram density estimation; model selection; minimum description length; dynamic programming;**

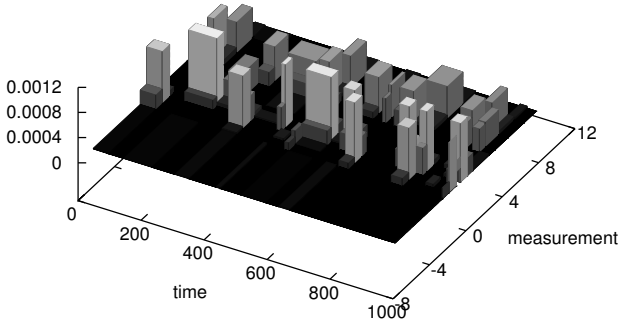

Fig. 1. A two-dimensional (time vs. measurement) histogram. The bins with higher density are highlighted by brighter colors.

I. Introduction

Oftentimes we face with real-valued data which are obtained by experiments or from sensors. For knowledge discovery from such raw data, discretization [1], [2], [3] is occasionally performed as preprocessing to make applicable a variety of sophisticated tools for symbolic data mining (such as frequent pattern mining) or logic-based machine learning (such as inductive logic programming). There have also been applications where discretized data are used for qualitative reasoning [4], [5], and now discretization can be viewed as one of the basic components in symbolic AI tools that are required to handle real-valued data. Real-valued time series are typically given as time series with no class annotation, and hence in this paper, we focus on *unsupervised* discretization of time series data.

Although discretization has not been paid much attention compared to the mining step for patterns or clusters, several authors proposed unsupervised discretization methods for time series, further to traditional equal-width/frequency binning. For example, Geurts uses a regression tree that segments the time axis to find the flat portions in the input time series [6]. SAX [7] combines a smoothing method called piecewise aggregate approximation (PAA) and equal-frequency binning under a Gaussian assumption on the distribution of the measurements. In the Persist algorithm [8], the measurement axis is segmented based on an information-theoretic score, where the segmented sections form a state space of a Markov process. Continuous hidden Markov models (HMMs) [8], [9] can also be used by regarding the most probable state sequence for the input time series as a sequence of discretized values.

In this paper, we aim to find meaningful discrete values that can be directly translated into some intuitive symbols such as "low," "medium" and "high." For this purpose, the boundaries among discrete values need to be determined adaptively from the input time series. Besides, in the present paper, we will take a non-parametric approach, since it seems more suitable for discretization methods to be neutral. In other words, we would like to avoid the preprocessing step from having extra assumptions, like the Gaussian assumption in SAX and continuous HMMs, and the Markov assumption in the Persist algorithm and continuous HMMs. In particular, we focus on histogram density estimation, a popular non-parametric modeling method, to capture the axis-parallel subregions in the measurement-time space where the data points are densely distributed. Also, for having more meaningful discrete values, we also prefer the histogram to be constructed to have variable-width bins, while many previous methods concentrate on histograms with equal-width bins (e.g. [10]).

Kontkanen and Myllymäki [11] proposed a method for construction of a single-dimensional histogram of variable-width bins that optimize a minimum description length (MDL) score using the *normalized maximum likelihood* (NML). Their histogram construction method, hereafter called the K&M method, efficiently finds the optimal set of bin boundaries, with respect to NML, in a dynamic programming manner. Based on the K&M method, this paper proposes an unsupervised discretization method for time series data, which adopts a two-dimensional (time vs. measurement) histogram illustrated in Fig. 1. More specifically, a two-dimensional

histogram is first constructed from an input time series by an extended K&M method, and the time series is then discretized using the density information from the constructed histogram. The extended K&M method we develop is also designed in a dynamic programming manner, and runs in polynomial time.

In addition to neutrality described above, our proposed method has a couple of advantages. First, using axis-parallel bins of variable-width, we can introduce the notion of continuous time and perform stronger smoothing (or noise reduction) along with the time axis, considering a global nature of the input time series. By this feature, the proposed method would work for noisy time series that continuous HMMs under a discrete-time Markov process cannot precisely deal with. Secondly, there is a representational merit in axis-parallel bins that they give us readable propositions of the form "$\alpha < X \leq \beta$" where α and β are bin boundaries, and X is a random variable for measurement. The last advantage of the proposed method is that the extended K&M method only requires easily-specifiable control parameters, compared to Bayesian discretizers [9]. Indeed, under the MDL framework using NML, we have a promising built-in penalty factor that works as well as a 'tuned' prior distribution or hyperparameters in Bayesian learning methods.

The rest of this paper is structured as follows. Section II presents an overview and details of the proposed method. In Section III, we show some experimental results that exhibit the robustness and the usefulness of the proposed method. Section IV concludes the paper.

II. Proposed method

First of all, in the proposed method, the input time series $\boldsymbol{s}$ of length n is simply seen as a set of data points $\{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$, where x_i and t_i are the measurement and the time at the i-th observation, respectively, and for simplicity, we assume that $t_i < t_{i'}$ holds when $i < i'$. As mentioned above, we are considering continuous time, so both x_i and t_i are real numbers. Then, we perform a discretization of the time series $\boldsymbol{s}$, where each raw measurement x_i is replaced by a plausible discrete value k_i. In the paper, these discrete values are numbered as integers from 1 to K, and called *discrete levels*. We map higher raw measurements (except noises) into higher discrete levels. The number K of discrete levels is also determined from the input $\boldsymbol{s}$ in our method.

In outline, the proposed method runs in two steps:

1) Conduct a density estimation of the time series $\boldsymbol{s}$ in the measurement-time space.
2) Discretize $\boldsymbol{s}$ using the estimated density.

In the first step, a two-dimensional histogram like Fig. 1 is constructed using the extended K&M method. On the other hand, the second step is illustrated in Fig. 2. We start from the input time series in Fig. 2 (a), and consider a grid comprised of bin boundaries in the constructed histogram, as shown in Fig. 2 (b). The measurement axis is segmented into discrete levels by the bins, and there are 11 discrete levels in Fig. 2 (b). Then, for each $1 \leq i \leq n$, the bin with the largest probability mass covering t_i indicates the most probable discrete level k_i at the time point. Such bins are shaded in Fig. 2 (c). Finally, we suppress the discrete levels that have not been chosen at any time point, and return the most probable discrete levels (with new indices) as the output. For example, as shown in Fig. 2 (d), we finally have five discrete levels, and return a sequence $\langle 1, 4, 5, 2, 3, \ldots \rangle$ of the most probable discrete levels. It is remarkable in Fig. 2 (d) that, despite the many outliers in the input, discretization is done quite robustly thanks to strong smoothing along the time axis.

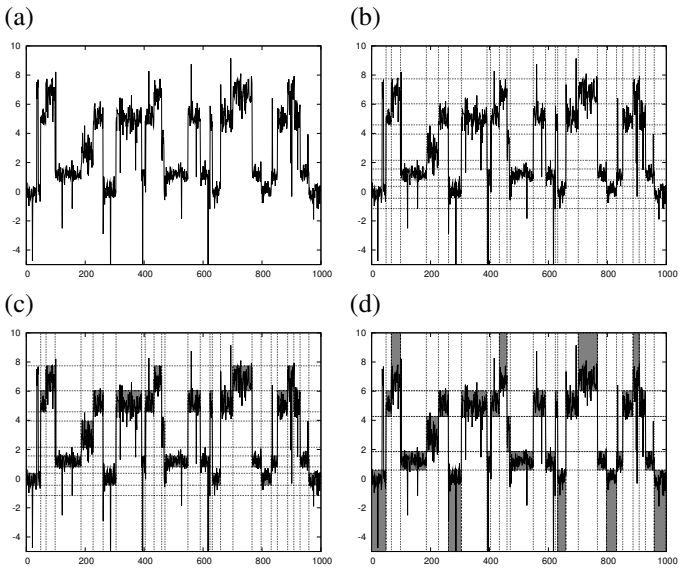

Fig. 2. Discretization using the estimated density.

In the remainder of this section, we describe the details of the first step. Specifically, our histogram model and a MDL-based setting for histogram construction in Section II-A and Section II-B, respectively. Then, the detailed procedure of the extended K&M method will be presented in Section II-C. Lastly in Section II-D, we discuss the computational complexity of the extended K&M method.

A. Histogram model

Now we formulate our histogram model. The formulation here is actually an extension of the one given in [11] into our two-dimensional case. First, as written before, the input time series $\boldsymbol{s}$ of length n is given by $\boldsymbol{s} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i$ is a pair of the time t_i and the measurement x_i at the i-th observation ($1 \leq i \leq n$). We assume that data points are recorded at a finite accuracy ϵ (so t_i's and x_i's are all divisible by ϵ), and a discrete measurement-time space $\mathcal{X} \times \mathcal{T}$ is introduced by a fine-grained 'pre-discretization' where:

$$\begin{aligned} \mathcal{X} &= \{x_{\min} + j\epsilon \mid j = 0, \ldots, (x_{\max} - x_{\min})/\epsilon\}, \\ \mathcal{T} &= \{t_{\min} + j'\epsilon \mid j' = 0, \ldots, (t_{\max} - t_{\min})/\epsilon\}, \end{aligned}$$

$x_{\min} = \min_i x_i$, $x_{\max} = \max_i x_i$, $t_{\min} = \min_i t_i$ and $t_{\max} = \max_i t_i$. We also consider that the candidates of bin boundaries, or *cut points*, are given as the neighbors of quantile

values excluding the minimum and the maximum:

$$\mathcal{C} = \left\{ Q\left(\frac{m}{M}, \{x_1, \ldots, x_n\}\right) + \frac{\epsilon}{2} \;\middle|\; m = 1, \ldots, M-1 \right\},$$
$$\mathcal{D} = \left\{ Q\left(\frac{m'}{M'}, \{t_1, \ldots, t_n\}\right) + \frac{\epsilon}{2} \;\middle|\; m' = 1, \ldots, M'-1 \right\},$$

where $Q(q, Z)$ is the q-quantile value of the values in Z,[1] and both M and M' are the control parameters that determine the granularity of the cut points. In our histogram model, the measurement axis and the time axis are segmented into K intervals and K' intervals, respectively. So the cut points $C = \{c_1, c_2, \ldots, c_{K-1}\}$ and $D = \{d_1, d_2, \ldots, d_{K'-1}\}$ are chosen from $\mathcal{C}$ and $\mathcal{D}$, respectively ($C \subseteq \mathcal{C}$ and $D \subseteq \mathcal{D}$). We also define $c_0 = x_{\min} - \epsilon/2$, $c_K = x_{\max} + \epsilon/2$, $d_0 = t_{\min} - \epsilon/2$ and $d_{K'} = t_{\max} + \epsilon/2$. For each $1 \le k \le K$ and $1 \le k' \le K'$, the (k, k')-th bin covers the region $[c_{k-1}, c_k] \times [d_{k'-1}, d_{k'}]$,[2] and the widths along the measurement axis and the time axis of this region are respectively denoted by $L_k = c_k - c_{k-1}$ and $L'_{k'} = d_{k'} - d_{k'-1}$. Furthermore, the (k, k')-bin is assigned a probability mass $\theta_{kk'}$ such that $0 \le \theta_{kk'} \le 1$ and $\sum_{k,k'} \theta_{kk'} = 1$. Here $\theta_{kk'}$'s are also considered as the parameters of the histogram model. For a data point $\boldsymbol{x} = (x, t) \in [c_{k-1}, c_k] \times [d_{k'-1}, d_{k'}]$, the density[3] is:

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}, C, D) = \frac{\epsilon^2 \theta_{kk'}}{L_k L'_{k'}} \tag{1}$$

and the likelihood of $\boldsymbol{s}$ is computed as:

$$f(\boldsymbol{s} \mid \boldsymbol{\theta}, \mathcal{M}) = \prod_{k=1}^{K} \prod_{k'=1}^{K'} \left(\frac{\epsilon^2 \theta_{kk'}}{L_k L'_{k'}} \right)^{h_{kk'}} \tag{2}$$

where $h_{kk'}$ is the number of data points falling into the (k, k')-th bin, and $\mathcal{M}$ indicates the model, i.e. $\mathcal{M} = (C, D)$ in our case. After estimating the maximum likelihood parameters $\hat{\theta}_{kk'} = h_{kk'}/n$, we have the maximized likelihood:

$$f(\boldsymbol{s} \mid \hat{\boldsymbol{\theta}}(\boldsymbol{s}), \mathcal{M}) = \prod_{k=1}^{K} \prod_{k'=1}^{K'} \left(\frac{\epsilon^2 h_{kk'}}{L_k L'_{k'} n} \right)^{h_{kk'}}. \tag{3}$$

B. MDL-based histogram construction

Finding the optimal cut points C and D of a histogram can be translated into a model selection problem where we choose the optimal histogram model $\mathcal{M} = (C, D)$, which is formulated above. MDL is a general framework for model selection, and following [11], now we will describe the settings for our histogram construction problem under the MDL framework. First, we introduce the normalized maximum likelihood (NML):

$$f_{\mathrm{NML}}(\boldsymbol{s} \mid \mathcal{M}) = \frac{f(\boldsymbol{s} \mid \hat{\boldsymbol{\theta}}(\boldsymbol{s}), \mathcal{M})}{\mathcal{R}_{\mathcal{M}}} \tag{4}$$

[1]For example, the median is 0.5-quantile value, and the first quartile value is 0.25-quantile value.

[2]The overlaps on the bin boundaries can be ignored since it is guaranteed that there are no data points on the bin boundaries.

[3]More precisely, $f(\boldsymbol{x} \mid \boldsymbol{\theta}, C, D)$ is the probability of $\boldsymbol{x}$ falling into the region $[x - \epsilon/2, x + \epsilon/2] \times [t - \epsilon/2, t + \epsilon/2]$, which has an area of ϵ^2. Besides, the (k, k')-th bin has an area of $L_k L'_{k'}$ and thus we have Eq. 1.

where $\mathcal{R}_{\mathcal{M}}$ is the normalizing constant called the *parametric complexity* or *minimax regret*, which is defined as:

$$\mathcal{R}_{\mathcal{M}} = \textstyle\sum_{\boldsymbol{s} \in \mathcal{S}} f(\boldsymbol{s} \mid \hat{\boldsymbol{\theta}}(\boldsymbol{s}), \mathcal{M}). \tag{5}$$

Here, $\mathcal{S}$ is a set of possible time series of length n, i.e. $\mathcal{S} = (\mathcal{X} \times \mathcal{T})^n$. The stochastic complexity $SC(\boldsymbol{s} \mid \mathcal{M})$ is then defined as the negative of the logarithm of the NML. Furthermore, following [11], we add the code length for the model index, and finally obtain the following MDL score:

$$\begin{aligned} B(\boldsymbol{s} \mid \mathcal{M}) &= SC(\boldsymbol{s} \mid \mathcal{M}) + \log \tbinom{E}{K-1} \tbinom{E'}{K'-1} \\ &= -\log f_{\mathrm{NML}}(\boldsymbol{s} \mid \mathcal{M}) + \log \tbinom{E}{K-1} \tbinom{E'}{K'-1} \\ &= -\log f(\boldsymbol{s} \mid \hat{\boldsymbol{\theta}}(\boldsymbol{s}), \mathcal{M}) \\ &\qquad + \log \mathcal{R}_{\mathcal{M}} + \log \tbinom{E}{K-1} \tbinom{E'}{K'-1} \\ &= -\sum_{k,k'} h_{kk'} \log \frac{\epsilon^2 h_{kk'}}{L_k L'_{k'} n} \\ &\qquad + \log \mathcal{R}_{\mathcal{M}} + \log \tbinom{E}{K-1} \tbinom{E'}{K'-1} \end{aligned} \tag{6}$$

where $E = |\mathcal{C}|$ and $E' = |\mathcal{D}|$. Note here that, in Eq. 6, the term depending on ϵ is constant to the model $\mathcal{M} = (C, D)$, and so is ignorable in choosing the model.

Computing the normalizing constant $\mathcal{R}_{\mathcal{M}}$ is not feasible in general, so it has been one of the main concerns in MDL researches how to compute $\mathcal{R}_{\mathcal{M}}$ [12], [13], [14]. As a special case of [12], we can derive an efficient way for computing $\mathcal{R}_{\mathcal{M}}$ for our purpose. The derivation is given in the appendix of this paper. Eventually $\mathcal{R}_{\mathcal{M}}$ only depends on K, K' and n, so we write $\mathcal{R}_{\mathcal{M}} = \mathcal{R}(K, K', n)$ and obtain its recursive form as follows:

$$\begin{aligned} \mathcal{R}(K, K', n) = \sum_{r_1 + r_2 = n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2} \cdot \\ \mathcal{R}(K^*, K', r_1) \mathcal{R}(K - K^*, K', r_2). \end{aligned} \tag{7}$$

We choose K^* in Eq. 7 from $1 \le K^* < K$ so that the depth of recursion is small, and the following properties also hold:

$$\begin{aligned} \mathcal{R}(1, K', n) &= \mathcal{R}_0(K', n), \\ \mathcal{R}(K, K', 0) &= 1, \\ \mathcal{R}(K, K', n) &= \mathcal{R}(K', K, n). \end{aligned} \tag{8}$$

Here $\mathcal{R}_0(K, n)$ is the parametric complexity used for single-dimensional histogram models [11], and can be efficiently computed in a recursive form [13]:

$$\begin{aligned} \mathcal{R}_0(K+2, n) &= \mathcal{R}_0(K+1, n) + \frac{n}{K} \mathcal{R}_0(K, n) \\ \mathcal{R}_0(1, n) &= 1 \\ \mathcal{R}_0(2, n) &= \sum_{r1 + r2 = n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n}\right)^{r_1} \left(\frac{r_2}{n}\right)^{r_2}. \end{aligned}$$

Also it is easy to see from above that $\mathcal{R}_0(K, 0) = 1$ holds.

To summarize, based on $B(\boldsymbol{s} \mid \mathcal{M})$ (computed from Eqs. 6 and 7), we choose plausible cut points C and D from the candidates $\mathcal{C}$ and $\mathcal{D}$, respectively. This can be seen as a model selection problem, and we will solve it efficiently in a dynamic programming manner as described in the next section.

C. Iterative histogram construction

For a single-dimensional case, the K&M method is designed to find the *optimal* histogram in a dynamic programming fashion. However, in our two-dimensional case, it does not seem easy to simultaneously optimize all cut points at the measurement axis and the time axis, since the configuration of cut points at one axis globally influences the configuration of those at the other axis. Instead, we propose an extended version of the K&M method that takes an approximate and iterative approach. That is, in this extended K&M method, we alternately optimize the cut points at one axis, fixing the configuration of cut points at the other axis, until reaching some locally optimal MDL score. The optimization at each axis is efficiently done in a dynamic programming fashion, as in the single-dimensional case. In what follows, we describe the details of the extended K&M method.

Before starting, let us introduce some notations. we consider to have at most $(K_{\max}-1)$ cut points at the measurement axis, and at most $(K'_{\max}-1)$ cut points at the time axis. Then, there can be $K_{\max} \times K'_{\max}$ bins in total at maximum. The elements in $\mathcal{C}$ are enumerated as $\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_E$ so that $\tilde{c}_{e_1} < \tilde{c}_{e_2}$ when $e_1 < e_2$, and we let $\tilde{c}_{E+1} = x_{\max} + \epsilon/2$. We also enumerate $\mathcal{D}$ as $\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_{E'}$ where $\tilde{d}_{e'_1} < \tilde{d}_{e'_2}$ when $e'_1 < e'_2$, and let $\tilde{d}_{E'+1} = t_{\max} + \epsilon/2$. The prefixes of $\mathcal{C}$ and $\mathcal{D}$ are then introduced as $\mathcal{C}_e = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_{e-1}\}$ and $\mathcal{D}_{e'} = \{\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_{e'-1}\}$. Besides, for $1 \le e \le E+1$ and $1 \le e' \le E'+1$, we define partial datasets $\boldsymbol{s}_e = \{(x,t) \in \boldsymbol{s} \mid x \le \tilde{c}_e\}$ and $\boldsymbol{s}'_{e'} = \{(x,t) \in \boldsymbol{s} \mid t \le \tilde{d}_{e'}\}$, and their sizes $n_e = |\boldsymbol{s}_e|$ and $n'_{e'} = |\boldsymbol{s}'_{e'}|$.

Now consider to find the optimal cut points C at the measurement axis, with the cut points $D = \{d_1, d_2, \ldots, d_{K'-1}\}$ at the time axis being fixed. For a partial (k, K')-bin histogram H that covers $\boldsymbol{s}_e$ ($1 \le k \le K_{\max}$, $1 \le e \le E+1$), we introduce its MDL score $B(\boldsymbol{s}_e \mid \hat{C}, D)$ like Eq. 6, focusing on the region $[x_{\min} - \epsilon/2, \tilde{c}_e] \times [t_{\min} - \epsilon/2, t_{\max} + \epsilon/2]$, where the cut points $\hat{C}$ are chosen from $\mathcal{C}_e$ and $|\hat{C}| = k-1$. Furthermore, for $k > 1$ and along the measurement axis, we consider an H's sub-histogram of $(k-1, K')$ bins where its cut points $\hat{C}^-$ is the immediate prefix of $\hat{C}$, i.e. $\hat{C}^- = \hat{C} \setminus \{\tilde{c}_{e^*}\}$ and $\tilde{c}_{e^*} = \max \hat{C}$. Then, the MDL score of H can be computed using the MDL score of this sub-histogram:

$$\begin{aligned} & B(\boldsymbol{s}_e \mid \hat{C}, D) \\ &= B(\boldsymbol{s}_{e^*} \mid \hat{C}^-, D) - \sum_{k'=1}^{K'} \hat{h}_{kk'} \log \frac{\epsilon^2 \hat{h}_{kk'}}{(\tilde{c}_e - \tilde{c}_{e^*}) L'_{k'} n} \\ &\quad + \log \frac{\mathcal{R}(k, K', n_e)}{\mathcal{R}(k-1, K', n_{e^*})} + \log \frac{E-k+2}{k-1} \end{aligned} \tag{9}$$

where the last term in the right hand side of Eq. 9 is obtained from $\log\left(\binom{E}{k-1} / \binom{E}{k-2}\right)$, and $\hat{h}_{kk'}$ is the number of data points falling into the (tentative) (k,k')-th bin which covers the region $[\tilde{c}_{e^*}, \tilde{c}_e] \times [d_{k'-1}, d_{k'}]$.

To realize dynamic programming in finding the optimal C, we introduce the following intermediate optimal MDL score for each $1 \le k \le K_{\max}$ and $1 \le e \le E+1$:

$$\hat{B}(k,e) = \min_{\hat{C}: \hat{C} \subseteq \mathcal{C}_e, |\hat{C}| = k-1} B(\boldsymbol{s}_e \mid \hat{C}, D). \tag{10}$$

Note here that D is implicitly given behind $\hat{B}(k,e)$ for notational brevity. Since $\mathcal{C}_{E+1} = \mathcal{C}$ and $\boldsymbol{s}_{E+1} = \boldsymbol{s}$ by definition, it is easy to see from Eq. 10 that $\min_k \hat{B}(k, E+1)$ is the optimal MDL score, and the number K of bins on the measurement side is determined as its minimizer, i.e. $K = \operatorname{argmin}_k \hat{B}(k, E+1)$. A similar discussion is also possible in finding the optimal cut points D at the time axis with the cut points C at the measurement axis being fixed, and using the property in Eq. 9, we finally obtain the extended K&M method as shown in Fig. 3. In Fig. 3, $\psi(k,e)$ is used for keeping track the minimizer e^* of $\hat{B}(k,e,e^*)$ in the induction phase. In the backtrack phase, on the other hand, we build the optimal cut points by tracing back ψ.

It is important to note that each optimization at one axis basically decreases, and does never increase the MDL score $B(\boldsymbol{s} \mid C, D)$. Hence, the extended K&M method is guaranteed to find some locally optimal cut points with respect to the MDL score, after several alternate updates of C and D.

D. Computational complexity

By introducing an iterative strategy, the extended K&M method achieves polynomial computation time. The main computational burden often lies in computing the parametric complexity $\mathcal{R}(K, K', n)$, and its computation time is evaluated as $O(n^2 \log \min\{K_{\max}, K'_{\max}\})$ from Eq. 7 and Eq. 8 (symmetry).[4] Here n is the number of data points in the input time series, and $K_{\max}$ and $K'_{\max}$ are the control parameters of the extended K&M method (i.e. we potentially have $(K_{\max}, K'_{\max})$-bin histogram at maximum). The term $\log K_{\max}$ or $\log K'_{\max}$ corresponds to the depth of recursion in Eq. 7.

In addition to computing the parametric complexity, as can be seen from Fig. 3, the computation of $\hat{B}(k,e,e^*)$ and $\hat{B}'(k',e',e^*)$ respectively take $O(E^2 K_{\max} K'_{\max})$ and $O(E'^2 K_{\max} K'_{\max})$ time, and are computationally dominant in the extended K&M method. Here E and E' are also the control parameters of the extended K&M method, which indicate the number of candidate cut points at the measurement axis and the time axis, respectively.

Finally, letting U be the number of iterations until convergence, the total computation time of the extended K&M method is $O(n^2 \log \min\{K_{\max}, K'_{\max}\} + U(E^2 + E'^2) K_{\max} K'_{\max})$. Furthermore, we specify $K_{\max} \ll K'_{\max}$ and $E \ll E'$ in many cases, and then the computation time will be $O(n^2 \log K_{\max} + U E'^2 K_{\max} K'_{\max})$. In our experiments described later, U is not so large (typically $U \le 10$). Obviously, larger $K_{\max}$, $K'_{\max}$, E and E' produce a more precise histogram, so we can specify these control parameters, *just* balancing the time and the quality of the histogram.

[4] The computation of $\mathcal{R}_0(K, n)$ takes only $O(n+K)$ time [11], [13], and so is negligible.

1) Choose initial cut points $D \subseteq \mathcal{D}$ which have as equal intervals as possible at time axis.
2) Alternate the following two steps until $B(\boldsymbol{s} \mid C, D)$ converges:
   a) Choose the optimal cut points $C \subseteq \mathcal{C}$ at the measurement axis, with D (accordingly K' and E') being fixed: (Induction)

$$\begin{aligned}
\hat{B}(k, e, e^*) &:= \hat{B}(k-1, e^*) - \sum_{k'=1}^{K'} \hat{h}_{kk'} \log \frac{\epsilon^2 \hat{h}_{kk'}}{(\tilde{c}_e - \tilde{c}_{e^*}) L'_{k'} n} + \log \frac{\mathcal{R}(k, K', n_e)}{\mathcal{R}(k-1, K', n_{e^*})} + \log \frac{E-k+2}{k-1} \\
&\qquad (\text{for } 1 < k \le K_{\max},\ 1 \le e \le E+1 \text{ and } k-1 \le e^* < e), \\
\hat{B}(k, e) &:= \min_{k-1 \le e^* < e} \hat{B}(k, e, e^*) \qquad (\text{for } 1 < k \le K_{\max} \text{ and } 1 \le e \le E+1), \\
\psi(k, e) &:= \operatorname{argmin}_{k-1 \le e^* < e} \hat{B}(k, e, e^*) \qquad (\text{for } 1 < k \le K_{\max} \text{ and } 1 \le e \le E+1), \\
\hat{B}(1, e) &:= -\sum_{k'=1}^{K'} \hat{h}_{1k'} \log \frac{\epsilon^2 \hat{h}_{1k'}}{(\tilde{c}_e - (x_{\min} - \frac{\epsilon}{2})) L'_{k'} n} + \log \mathcal{R}_0(K', n_e) + \log \binom{E'}{K'-1} \quad (\text{for } 1 \le e \le E+1).
\end{aligned}$$

   (Backtrack) Let $C := \{\tilde{c}_{e_1}, \tilde{c}_{e_2}, \ldots, \tilde{c}_{e_{K-1}}\}$ after computing:

$$\begin{aligned}
K &:= \operatorname{argmin}_{1 \le k \le K_{\max}} \hat{B}(k, E+1), \\
e_{K-1} &:= \psi(K, E+1), \\
e_k &:= \psi(k, e_{k+1}) \qquad (\text{for } 1 \le k \le K-2).
\end{aligned}$$

   b) Choose the optimal cut points $D \subseteq \mathcal{D}$ at the time axis, with C (accordingly K and E) being fixed: (Induction)

$$\begin{aligned}
\hat{B}'(k', e', e^*) &:= \hat{B}'(k'-1, e^*) - \sum_{k=1}^{K} \hat{h}_{kk'} \log \frac{\epsilon^2 \hat{h}_{kk'}}{L_k(\tilde{d}_{e'} - \tilde{d}_{e^*}) n} + \log \frac{\mathcal{R}(K, k', n'_{e'})}{\mathcal{R}(K, k'-1, n'_{e^*})} + \log \frac{E'-k'+2}{k'-1} \\
&\qquad (\text{for } 1 < k' \le K'_{\max},\ 1 \le e' \le E'+1 \text{ and } k'-1 \le e^* < e'), \\
\hat{B}'(k', e') &:= \min_{k'-1 \le e^* < e'} \hat{B}'(k', e', e^*) \qquad (\text{for } 1 < k' \le K'_{\max},\ 1 \le e' \le E'+1), \\
\psi'(k', e') &:= \operatorname{argmin}_{k'-1 \le e^* < e'} \hat{B}'(k', e', e^*) \qquad (\text{for } 1 < k' \le K'_{\max},\ 1 \le e' \le E'+1), \\
\hat{B}'(1, e') &:= -\sum_{k=1}^{K} \hat{h}_{k1} \log \frac{\epsilon^2 \hat{h}_{k1}}{L_k(\tilde{d}_{e'} - (t_{\min} - \frac{\epsilon}{2})) n} + \log \mathcal{R}_0(K, n'_{e'}) + \log \binom{E}{K-1} \quad (\text{for } 1 \le e' \le E'+1).
\end{aligned}$$

   (Backtrack) Let $D := \{\tilde{d}_{e'_1}, \tilde{d}_{e'_2}, \ldots, \tilde{d}_{e'_{K'-1}}\}$ after computing:

$$\begin{aligned}
K' &:= \operatorname{argmin}_{1 \le k' \le K'_{\max}} \hat{B}'(k', E'+1), \\
e'_{K'-1} &:= \psi(K', E'+1), \\
e'_{k'} &:= \psi(k', e_{k'+1}) \qquad (\text{for } 1 \le k' \le K'-2).
\end{aligned}$$

Fig. 3. The extended K&M method.

III. Experiments

In this section, we present two experimental results. The dataset in the first experiment is an artificial dataset, which is originally used in a comparative study by Mörchen et al. [8]. In this paper, we call this dataset the *enduring-state dataset*. The second experiment uses a real dataset on muscle activation of a professional inline speed skater [15],[5] which is also originally provided by Mörchen et al. Throughout these experiments, we will show the robustness and the usefulness of the proposed method.

[5]The dataset is included in the package of the Persist algorithm's MATLAB implementation (http://www.mybytes.de/persist.php).

A. Enduring-state dataset

In the experiment with the enduring-state dataset, we compare the proposed method with the previous discretization methods including SAX [7], the Persist algorithm [8], continuous HMMs, a Bayesian hybrid method of continuous HMMs and the Persist algorithm [9], according to the predictive performance. The hybrid method is trained under a recently developed Bayesian learning framework called variational Bayes [16]. On the other hand, non-hybrid continuous HMM is trained under maximum likelihood estimation. The detailed behaviors of the previous methods are reported in [9].

In the enduring-state dataset, raw time series of length 1,000

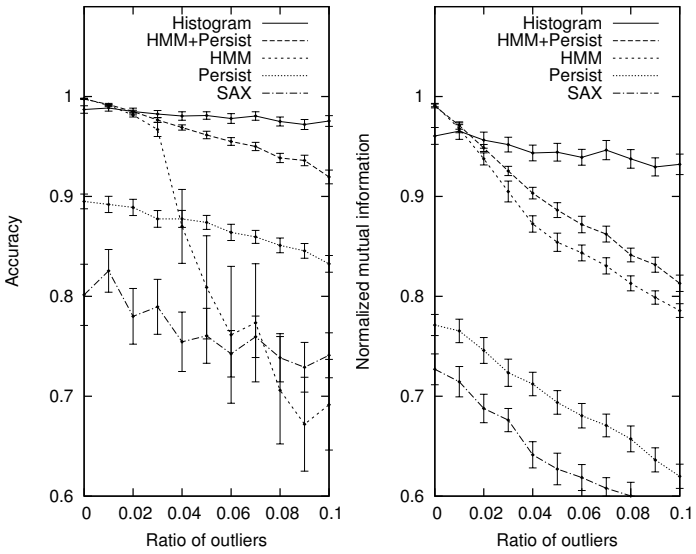

Fig. 4. Predictive performance in the enduring-state datasets with five discrete levels. "Persist," "HMM," "HMM+Persist," and "Histogram" respectively indicate the Persist algorithm, continuous HMM, the hybrid of continuous HMMs and the Persist algorithm, and the proposed method.

are generated by a state machine which randomly changes its state after a random duration. At each state, the data points (the measurements) are generated with Gaussian noises around the mean proper to the state. The generation process is thus close to a hidden Markov process, but additionally, some of the data points are replaced with outliers. The ratio of these outliers varies from 0% to 10%. Actually, Fig. 2 (a) shows an instance of enduring-state time series with five states and 5% outliers. Each state during the generation process corresponds to a discrete level, and hereafter the state sequence obtained in the generation process of a raw time series is called the *answer* sequence, and the output of a discretizer is referred to as the *predicted* sequence. [8] gives more details on the generation process of the enduring-state dataset. Furthermore, following [9], we use accuracy and normalized mutual information (NMI) [17] as evaluation criteria on predictive performance. NMI is frequently used in evaluation of a clustering result.

The goal here is to see how well the discretizers recover the answer sequence from (very) noisy time series. We tested the discretization methods above on 100 time series for each number $K = 2, 3, \ldots, 7$ of discrete levels and ratio $R = 0\%, 1\%, \ldots, 10\%$ of outliers. We follow [9] as to the detailed experimental procedure. In particular, in SAX, we picked up the frame width from $\{1, 2, 3, 5, 10, 20, 50\}$ that works best for each pair of K and R. Also for the hybrid method, we choose the best hyperparameter, which works as weights for prior knowledge brought by the Persist algorithm, from $\{0.5, 1, 5, 10, 20, 50, 70, 100\}$. The proposed method is tested under the control parameters $K_{\max} = 15$, $K'_{\max} = 50$, $E = 100$ and $E' = 1000$.

Fig. 4 shows the median accuracy (left) and the median NMI (right) for the time series with five discrete levels and various ratios of outliers. The error bars indicate the 95% median absolute deviation (MAD) t confidence interval [18]. From these graphs, we can see that non-hybrid continuous HMMs work nearly perfectly for time series with no outliers (this is not surprising since the generation process is an hidden Markov process), but their performance quickly degrades as the ratio of outliers increases. On the other hand, the Persist algorithm robustly works, but constantly makes errors, since the Persist algorithm only finds the cut points at the measurement axis and many small Gaussian noises (which are not outliers) easily go across these cut points [9]. The Bayesian hybrid of continuous HMMs and the Persist algorithm surely combines the strong points of the base discretizers, but its performance still degrades when we have many outliers. Compared to these previous methods, the proposed discretization method is quite robust against small noises and outliers. As visually illustrated in Fig. 2, this robustness seems to come from strong smoothing along the time axis, and from precise identification of the cut points at the measurement axis, which takes into account a global nature of the input time series.

TABLE I
WILCOXON'S RANK SUM TEST ON ACCURACY BETWEEN THE HYBRID METHOD AND THE PROPOSED METHOD.

# of discrete levels	Ratio R of outliers										
K	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
2	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	-	-	-	-	-	-
4	x	x	x	-	o	-	o	o	o	o	o
5	x	x	-	-	o	o	o	o	o	o	o
6	x	-	-	o	o	o	o	o	o	o	o
7	x	-	o	o	o	o	o	o	o	o	o

Furthermore, we conducted Wilcoxon's rank sum test with the significance level 0.01 on predictive performance. Then, the proposed method is shown to be better than SAX and the Persist algorithm for all cases with $K = 2, 3, \ldots, 7$ and $R = 0\%, 1\%, \ldots, 10\%$ under accuracy and NMI. Table I shows the result of comparison under accuracy between the hybrid method and the proposed method. In this table, "x" indicates that the hybrid method works significantly better than the proposed method, "o" indicates that the proposed method works significantly better than the hybrid method, and "-" indicates that the difference between these two methods is not significant. We obtained a similar result under NMI, and Table I clearly exhibits the robustness of the proposed method.

B. Muscle dataset

In the second experiment, we use a real dataset to show the usefulness of the proposed method in a knowledge discovery task. In this dataset, the measurement is the muscle activation calculated from the original EMG (Electromyography) as the logarithm of the energy, and nearly 30,000 measurements are recorded. Since the time series contains numerous measurements, we first compressed the time series by the smoothing method called piecewise aggregate approximation (PAA) used in SAX [7].

PAA takes as input a discrete-time raw time series $\boldsymbol{s} = (x_1, x_2, \ldots, x_T)$ of length T, and compresses $\boldsymbol{s}$ into a new

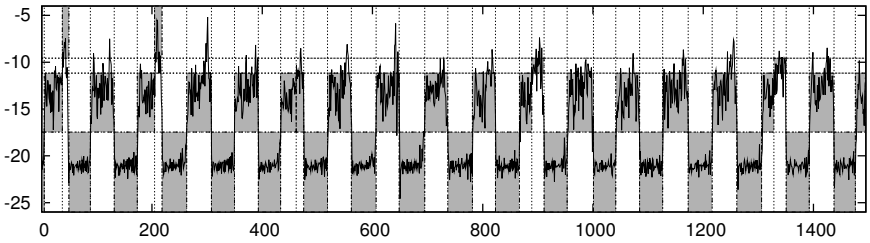

Fig. 5. Discretized muscle data. The x-axis indicates the time, and the y-axis indicates the measurement.

time series $\bar{\boldsymbol{s}} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{T'})$ where $T' < T$. In a simple case that $T = wT'$ holds where w is some positive integer which indicates the frame width (the other case is described in [7]), each $\bar{x}_{t'}$ is computed as $\bar{x}_{t'} = \frac{1}{w}\sum_{t=w(t'-1)+1}^{wt'} x_t$, the average of the measurements in the t'-th frame ($t' \in [1, T']$).

Fig. 5 shows the time series compressed by PAA with $w = 20$, and bin boundaries found by the proposed method. In the proposed method, we used the control parameters $K_{\max} = 50$, $K'_{\max} = 500$, $E = 1000$ and $E' = 1000$. The shaded regions in Fig. 5 indicate the most probable bins at each time point. From this result, we can observe a cyclic pattern of muscle activation in Fig. 5, and in some cycles, there are high activities at the end. This result is similar to the one reported in [8] and [9], where the second level (from -17.46 to -11.16) corresponds to the gliding phase for stabilizing the body's center of gravity, and the third and the fourth levels correspond to the last kick to the ground for moving forward. Fig. 5 shows that the proposed method can reveal the characteristics of the target time series as a symbolic sequence, in a well-founded manner under the MDL principle.

IV. Conclusion

This paper proposed a novel unsupervised discretization method for time series data, based on density estimation using a two-dimensional histogram under the MDL model selection framework. To realize this, we extend Kontkanen and Myllymäki's histogram construction method into the two-dimensional case. Histogram models, the base model class of the proposed method, are non-parametric, and hence require less assumptions on the input time series. This neutrality would enable the proposed method to be a ubiquitous preprocessing tool for AI tasks including knowledge discovery. Also, our extended histogram construction method achieves a polynomial computation time, and guarantees local optimality of the obtained histogram. Furthermore, as shown in the experimental results, the proposed method is more robust than the previous methods for noisy time series data, and would be useful for knowledge discovery tasks.

Acknowledgment

The author would like to thank the anonymous reviewers for valuable comments. This work is supported in part by Grant-in-Aid for Scientific Research (No. 20240016) from Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] C. Daw, C. Finney, and E. Tracy, "A review of symbolic analysis of experimental data," *Review of Scientific Instruments*, vol. 74, no. 2, pp. 915–930, 2003.

[2] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. of the 12th Int'l Conf. on Machine Learning (ICML-95)*, 1995, pp. 194–202.

[3] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: an enabling technique," *Data Mining and Knowledge Discovery*, vol. 6, no. 4, pp. 393–423, 2002.

[4] R. King, S. Garrett, and G. Coghill, "On the use of qualitative reasoning to simulate and identify metabolic pathways," *Bioinformatics*, vol. 21, no. 9, pp. 2017–2026, 2005.

[5] G. Synnaeve, K. Inoue, A. Doncescu, H. Nabeshima, Y. Kameya, M. Ishihata, and T. Sato, "Kinetic models and qualitative abstraction for relational learning in systems biology," in *Proc. of the 2011 Int'l Conf. on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS-2011)*, 2011.

[6] P. Geurts, "Pattern extraction for time series classification," in *Proc. of the 5th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD-01)*, 2001, pp. 115–127.

[7] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[8] F. Mörchen and A. Ultsch, "Optimizing time series discretization for knowledge discovery," in *Proc. of the 11th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-05)*, 2005, pp. 660–665.

[9] Y. Kameya, G. Synnaeve, A. Doncescu, K. Inoue, and T. Sato, "A Bayesian hybrid approach to unsupervised time series discretization," in *Proc. of the 2010 Conf. on Technologies and Applications of Artificial Intelligence (TAAI-2010)*, 2010, pp. 342–349.

[10] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.

[11] P. Kontkanen and P. Myllymäki, "MDL histogram density estimation," in *Proc. of the 11th Int'l Conf. on Artificial Intelligence and Statistics (AISTATS-07)*, 2007, pp. 219–226.

[12] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri, "An MDL framework for data clustering," in *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2005, pp. 323–354.

[13] P. Kontkanen and P. Myllymäki, "A linear-time algorithm for computing the multinomial stochastic complexity," *Information Processing Letters*, vol. 103, no. 6, pp. 227–233, 2007.

[14] T. Mononen and P. Myllymäki, "Fast NML computation for naive Bayes models," in *Proc. of the 10th Int'l Conf. on Discovery Science (DS-07)*, 2007, pp. 151–160.

[15] F. Mörchen, A. Ultsch, and O. Hoos, "Extracting interpretable muscle activation patterns with time series knowledge mining," *Int'l J. of Knowledge-based and Intelligence Engineering Systems*, vol. 9, no. 3, pp. 197–208, 2005.

[16] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University College London, 2003.

[17] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[18] M. O. Abu-Shawiesh, F. M. Al-Athari, and H. F. Kittani, "Confidence interval for the mean of a contaminated normal distribution," *J. of Applied Science*, vol. 9, no. 15, pp. 2835–2840, 2009.

APPENDIX

For self-containedness, following the detailed descriptions in [11], [12], we derive a recursive form (Eq. 7) of the parametric complexity $\mathcal{R}(K, K', n)$ for our case. First, from the definition of $\mathcal{R}(K, K', n)$ (Eq. 5) and the maximized likelihood of a histogram model (Eq. 3), we have:

$$\begin{aligned}
&\mathcal{R}(K, K', n) \\
&= \sum_{\boldsymbol{s} \in \mathcal{S}} \prod_{k,k'} \left(\frac{\epsilon^2 h_{kk'}}{L_k L'_{k'} n} \right)^{h_{kk'}} \\
&= \sum_{h_{kk'} : \sum_{kk'} h_{kk'} = n} \frac{n!}{\prod_{k,k'} h_{kk'}!} \prod_{k,k'} \left(\frac{L_k L'_{k'}}{\epsilon^2} \right)^{h_{kk'}} \cdot \\
&\qquad \prod_{k,k'} \left(\frac{\epsilon^2 h_{kk'}}{L_k L'_{k'} n} \right)^{h_{kk'}} \\
&= \sum_{h_{kk'}} \frac{n!}{\prod_{k,k'} h_{kk'}!} \prod_{k,k'} \left(\frac{h_{kk'}}{n} \right)^{h_{kk'}} .
\end{aligned}$$

Here, the term $(L_k L'_{k'}/\epsilon^2)^{h_{kk'}}$ considers all the possibilities where each of $h_{kk'}$ data points appears in one of $(L_k L'_{k'}/\epsilon^2)$ pre-discretized regions in the (k, k')-th bin. We further simplify the right hand side by introducing new intermediate variables $h_1, h_2, \ldots, h_K$:

$$\begin{aligned}
&\mathcal{R}(K, K', n) \\
&= \sum_{\substack{h_1 + \cdots + h_K = n \\ h_{11} + \cdots + h_{1K'} = h_1 \\ \cdots \\ h_{K1} + \cdots + h_{KK'} = h_K}} \frac{n!}{\prod_{k,k'} h_{kk'}!} \prod_{k,k'} \left(\frac{h_k}{n} \cdot \frac{h_{kk'}}{h_k} \right)^{h_{kk'}} \\
&= \sum_{h_k, h_{kk'}} \frac{n!}{\prod_{k,k'} h_{kk'}!} \prod_k \left(\frac{h_k}{n} \right)^{h_k} \prod_{k'} \left(\frac{h_{kk'}}{h_k} \right)^{h_{kk'}} \\
&= \sum_{h_k, h_{kk'}} \frac{n!}{\prod_k h_k!} \prod_k \left(\frac{h_k}{n} \right)^{h_k} \frac{h_k!}{\prod_{k'} h_{kk'}!} \prod_{k'} \left(\frac{h_{kk'}}{h_k} \right)^{h_{kk'}} \\
&= \sum_{h_1 + \ldots + h_K = n} \left(\frac{n!}{\prod_k h_k!} \prod_k \left(\frac{h_k}{n} \right)^{h_k} \right) \cdot \\
&\qquad \prod_k \sum_{h_{k1} + \cdots + h_{kK'} = h_k} \frac{h_k!}{\prod_{k'} h_{kk'}!} \prod_{k'} \left(\frac{h_{kk'}}{h_k} \right)^{h_{kk'}} \\
&= \sum_{h_1 + \ldots + h_K = n} \left(\frac{n!}{\prod_k h_k!} \prod_k \left(\frac{h_k}{n} \right)^{h_k} \right) \prod_k \mathcal{R}_0(K', h_k).
\end{aligned}$$

At this point, the sum in the last equation still requires exponential time. To avoid the problem, as described in [12], we consider to obtain a double recursive formula by introducing two new intermediate variable r_1 and r_2:

$$\begin{aligned}
&\mathcal{R}(K, K', n) \\
&= \sum_{h_1 + \ldots + h_K = n} \left(\frac{n!}{n^n} \prod_k \frac{h_k^{h_k}}{h_k!} \right) \prod_k \mathcal{R}_0(K', h_k) \\
&= \sum_{\substack{r_1 + r_2 = n \\ h_1 + \cdots + h_{K^*} = r1 \\ h_{K^*+1} + \cdots + h_K = r_2}} \Bigg[\frac{n!}{n^{r_1 + r_2}} \cdot \frac{r_1^{r_1}}{r_1!} \cdot \frac{r_2^{r_2}}{r_2!} \cdot \\
&\qquad \left(\frac{r_1!}{r_1^{r_1}} \prod_{k=1}^{K^*} \frac{h_k^{h_k}}{h_k!} \right) \left(\frac{r_2!}{r_2^{r_2}} \prod_{k=K^*+1}^{K} \frac{h_k^{h_k}}{h_k!} \right) \cdot \\
&\qquad \left(\prod_{k=1}^{K^*} \mathcal{R}_0(K', h_k) \right) \left(\prod_{k=K^*+1}^{K} \mathcal{R}_0(K', h_k) \right) \Bigg] \\
&= \sum_{r_1 + r_2 = n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n} \right)^{r_1} \left(\frac{r_2}{n} \right)^{r_2} \cdot \\
&\left[\sum_{\substack{h_1 + \cdots + h_{K^*} \\ = r_1}} \left(\frac{r_1!}{\prod_{k=1}^{K^*} h_k!} \prod_{k=1}^{K^*} \left(\frac{h_k}{r_1} \right)^{h_k} \right) \prod_{k=1}^{K^*} \mathcal{R}_0(K', h_k) \right] \cdot \\
&\left[\sum_{\substack{h_{K^*+1} + \cdots + h_K \\ = r_2}} \left(\frac{r_2!}{\prod_{k=K^*+1}^{K} h_k!} \prod_{k=K^*+1}^{K} \left(\frac{h_k}{r_1} \right)^{h_k} \right) \prod_{k=K^*+1}^{K} \mathcal{R}_0(K', h_k) \right] \\
&= \sum_{r_1 + r_2 = n} \frac{n!}{r_1! r_2!} \left(\frac{r_1}{n} \right)^{r_1} \left(\frac{r_2}{n} \right)^{r_2} \mathcal{R}(K^*, K', r_1) \mathcal{R}(K - K^*, K', r_2).
\end{aligned}$$