Generalized Local Aggregation for Large Scale Gaussian Process Regression

Yinghua Gao[†‡], Naiqi Li[†], Ning Ding[†], Yiming Li[†], Tao Dai[†‡] and Shu-Tao Xia[†‡]
[†]Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[‡]PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China
Email: {yh-gao18, lnq18, dingn18, li-ym18}@mails.tsinghua.edu.cn daitao.edu@gmail.com xiast@sz.tsinghua.edu.cn

***Abstract*—Despite being one of the most popular non-parametric approaches, Gaussian process regression (GPR) suffers from $\mathcal{O}(n^3)$ computational burden and the computation is infeasible for large-scale scenarios. To reduce the computational complexity, many Shannon-mutual-information-based aggregation methods were proposed, whereas these methods can not effectively identify the importance of experts in some cases. To address this problem, we generalize the traditional mutual-information-based methods (GPoE, RBCM, GRBCM) based on Tsallis mutual information. Accordingly, the generated weight distribution is more sparse tending to focus on those experts with good performance. To obtain adaptive and data-dependent entropic-index in Tsallis entropy, we propose three heuristic algorithms to solve our model. Extensive experiments show that, the proposed method can improve the prediction of both the mean and variance, and the improvement of variance prediction is significant in many cases.**

***Index Terms*—Gaussian process, Tsallis entropy, local aggregation**

I. Introduction

Gaussian Process Regression (GPR) is one of the most popular Bayesian statistical approaches [1]. Due to its powerful expression ability and elegant statistical property, GPR has been widely explored in various scenarios, such as Bayesian optimization [2], multi-task learning [3], computer vision [4] and reinforcement learning [5].

Although GPR possesses convenient and elegant properties in regression tasks, standard GPR suffers from $\mathcal{O}(n^3)$ computational burden and $\mathcal{O}(n^2)$ storage complexity with respect to the size of training set. Many previous works have been devoted to addressing this issue over past twenty years. The simplest strategy is to use an active set selected from the training set to train a GPR model. There are many rules to decide whether to choose a data point or not into the active set, such as informative vector machine [6] and matching pursuit [7]. Another stream of strategies is to employ m $(m \ll n)$ inducing points to summarize the whole training data. Some representative works belonging to this include Sparse Pseudo-inputs Gaussian Process (SPGP) [8] and Sparse Variational Gaussian Process (SVGP) [9]–[11]. Although the inference for inducing points method can be performed in $\mathcal{O}(nm^2)$ complexity, studies indicate that it is difficult to obtain a faithful representation for a quick-varying function with significant local structure [12]. Although some recent efforts of sparse approximately approaches have been made to better capture the local structure [13]–[15], they introduce extra hyperparameters that are difficult to tune and make the training more challenging. Therefore, current improvements for handling quick-varying functions by sparse approximately methods still have somewhat limitations.

In this paper, we concentrate on local aggregation models which are different from both strategies mentioned above. This stream of works give the predictive distributions of test outputs according to the feedback from a series of sub-models named GPR experts. The main advantage of local aggregation is that it directly operates on training data rather than inducing variables and the inference process can be achieved with high parallelization efficiency [16]. A recent review on scalable GPRs [17] also reveals that local aggregations are often superior to aforementioned methods in applications. Traditional aggregation algorithms usually assume the equality of each expert's contribution to the predictive distribution [18], [19]. To improve the predictive accuracy, current state-of-the-art aggregation algorithms employ a parameter to distinguish the importance of each expert [20]–[22], whose computation is based on Shannon mutual information between the prior distribution and posterior ones given the dependent subsets. However, can Shannon entropy based strategy explicitly reflect the importance of each expert?

This paper aims to answer the question above. we use a simple yet representative example to show that current methods can not efficiently distinguish the "good" experts (close to the test inputs) and "terrible" experts (far from the test inputs), as shown in Figure 1. Specifically, experts with terrible performance may also have relatively big weights. Based on this observation, we introduce Tsallis entropy to obtain a sparse weight distribution, which tends to focus on those experts with good performance.

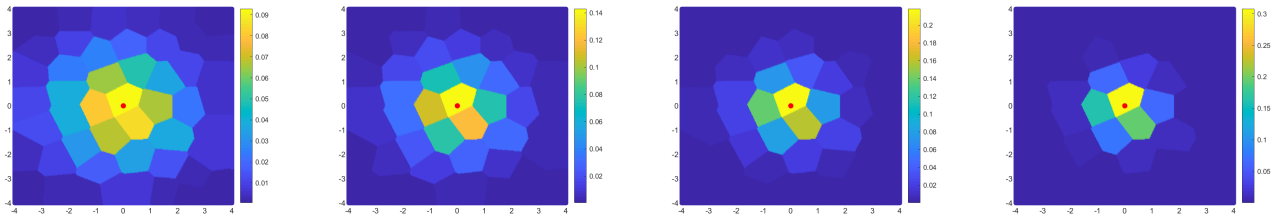
Overall, the major contributions of our paper are as follows:

1. To obtain an effective and sparse representation of

The first two authors contribute equally to this work. Tao Dai is the corresponding author.

This work is supported in part by the National Natural Science Foundation of China under Grant 61771273, Guangdong Basic and Applied Basic Research Foundation on 2019A1515110344, the China Postdoctoral Science Foundation under Grant 2019M660645, the RD Program of Shenzhen under Grant JCYJ20180508152204044, and the project "PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019)".

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) Shannon entropy ($q = 1$) (b) $q = 2$ (c) $q = 3$ (d) $q = 4$

Fig. 1: A toy example demonstrating the inadequacy of current aggregation models. **(a):** the prediction of Shannon-mutual-entropy-based method. **(b)-(d):** the prediction of Tsallis-entropy-based methods with different entropic-index q . When $q = 1$ Tsallis entropy is equivalent to Shannon case. Red point represents a test input and each surrounding sub-area can be regarded as an expert. The shade of color represents the weight of the expert, reflecting the importance of each expert. As shown in the figures, Tsallis-entropy-based methods can effectively distinguish the importance of each expert and lead to a sparse representation of weight distribution. More details can be found in Section 4.1.

expert's weights, we propose a generalized local aggregation framework for scalable GPRs based on Tsallis mutual information.

2. We empirically demonstrate that the entropic-index q in Tsallis entropy can characterize the sparsity of weight distribution and previous aggregation models can prove to be a special case in our framework ($q = 1$). Besides, we design three heuristic algorithms to solve our model in order to obtain adaptive and data-dependent entropic-index.

3. Extensive experiments show that the proposed method can improve the accuracy of predictive distribution and in many cases the improvement of variance prediction is significant.

II. BACKGROUND

A. Gaussian Process Regression (GPR)

Gaussian Process Regression (GPR) is a well studied non-parametric method for solving regression tasks [1], [23]–[25]. In a regression task we are given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ and y_i denote a d-dim feature vector and the scalar target output respectively. Besides we denote $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and $\mathbf{y} = \{y_1, \cdots, y_n\}$. In GPR the basic model is $y = f(\mathbf{x}) + \epsilon$, where ϵ represents $i.i.d.$ Gaussian noise and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, leading to the following estimation:

$$p(\mathbf{y}|\mathbf{f}, \sigma) = \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I}). \tag{1}$$

Furthermore, the regression function follows a Gaussian process, i.e., $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ denote mean and covariance function respectively. Therefore, $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$, where $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\mathbf{K} \in \mathbb{R}^{n\times n}$ represent the mean and the covariance matrix respectively. For a given finite dataset, the covariance matrix $\mathbf{K}$ is characterized by the kernel function, i.e. $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of kernel parameters. For the sake of brevity, it is assumed that $\boldsymbol{\mu}$ is zero. According to Bayes' theorem, we can obtain a closed form expression of the marginal likelihood of $\mathbf{y}$:

$$p(\mathbf{y}|\mathbf{X}, \sigma, \mathbf{K}) = \int p(\mathbf{y}|\mathbf{f}, \sigma)p(\mathbf{f}|\mathbf{K})d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \boldsymbol{\Sigma}), \tag{2}$$

where $\boldsymbol{\Sigma} = \mathbf{K} + \sigma^2\mathbf{I}$. The hyper-parameters σ and $\boldsymbol{\theta}$ can then be optimized by minimizing the negative log-likelihood, which is given by

$$-\log p(\mathbf{y}|\mathbf{X}, \sigma, \mathbf{K}) = \frac{1}{2}\mathbf{y}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} + \frac{1}{2}\log|\boldsymbol{\Sigma}| + \frac{n}{2}\log 2\pi. \tag{3}$$

For a test input $\mathbf{x}_*$, the prediction of the target value y_* also follows the Gaussian distribution:

$$\begin{aligned} p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) &= \mathcal{N}(y_*|\hat{\mu}_*, \hat{\sigma}_*) \\ \hat{\mu}_* &= \mathbf{k}_*^\top\boldsymbol{\Sigma}^{-1}\mathbf{y} \\ \hat{\sigma}_*^2 &= k(\mathbf{x}_*, \mathbf{x}_*; \boldsymbol{\theta}) - \mathbf{k}_*^\top\boldsymbol{\Sigma}^{-1}\mathbf{k}_* + \sigma^2 \\ [\mathbf{k}_*]_i &= k(\mathbf{x}_i, \mathbf{x}_*; \boldsymbol{\theta}). \end{aligned} \tag{4}$$

The determinant and inverse need to to be computed in $\mathcal{O}(n^3)$ which makes GPR infeasible for large datasets.

B. Local Aggregation

Local aggregation models aim to apply GPR on large-scale datasets with a divide-and-conquer strategy. This intuitive idea is to partition the original dataset into several subsets such that each of them is small enough to deal with, and then aggregate the predictions of individual experts by some heuristic aggregation rules. Specifically, we first utilize k-means clustering to partition the training set $\mathcal{D}$ into M subsets $\{\mathcal{D}_k\}_{k=1}^M$. Certainly other clustering algorithms are feasible providing that $\mathcal{D} = \cup_{k=1}^M\mathcal{D}_k$ and $\mathcal{D}_k \cap \mathcal{D}_l = \phi$ when $k \neq l$. For a new input $\mathbf{x}_i, i \notin [n]$, we could obtain the predictive mean $\hat{\mu}_{ik}$ and variance $\hat{\sigma}_{ik}^2$ based on the subset $\mathcal{D}_k$. The next step is to decide how to aggregate the individual predictions, which is the key difference among various aggregation models. In what follows we present a review of several aggregation strategies.

1) Product of Experts (PoE): Product of Experts (PoE) aims to figure out a target probability distribution as the product of a series of predictive distributions, each of which was given by $\{\mathbf{x}_i, \mathcal{D}_k\}$. Since we can acquire a predictive distribution of y_i by applying GPR on each subset, the ultimate predictive distribution can be formulated as:

$$p(y_i|\mathbf{x}_i, \mathcal{D}) = \frac{1}{Z}\prod_{k=1}^{M} p^{\beta_{ik}}(y_i|\mathbf{x}_i, \mathcal{D}_k) = \frac{1}{Z}\prod_{k=1}^{M} \mathcal{N}^{\beta_{ik}}(y_i|\hat{\mu}_{ik}, \hat{\sigma}^2_{ik}), \tag{5}$$

where Z denotes the normalized coefficient to ensure validity. β_{ik} can be regarded as a measure of importance of each expert on $\mathbf{x}_i$. As Gaussian distributions are closed under multiplication, we can obtain an analytical form:

$$\begin{aligned} &p(y_i|\mathbf{x}_i, \mathcal{D}) = \mathcal{N}(y_i|\hat{\mu}_i, \hat{\sigma}^2_i), \\ &\text{where} \quad \hat{\mu}_i = \hat{\sigma}^2_i \sum_{k=1}^{M} \beta_{ik}\hat{\mu}_{ik}\hat{\sigma}^{-2}_{ik}, \quad \hat{\sigma}^{-2}_i = \sum_{k=1}^{M} \beta_{ik}\hat{\sigma}^{-2}_{ik}. \end{aligned} \tag{6}$$

Original PoE assumes $\beta_{ik} = 1(k = 1, 2, \cdots, M)$ [19]. In contrast, Generalized product of Experts (GPoE) computes β_{ik} according to the difference of Shannon mutual information between prior distribution and posterior distribution depending on $\mathcal{D}_k$. The derivation of β_{ik} is as follows:

$$\begin{aligned} \beta_{ik} =& I(y_i; \mathcal{D}_k|\mathbf{x}_i) \\ =& H(y_i|\mathbf{x}_i) - H(y_i|\mathbf{x}_i, \mathcal{D}_k) \\ =& \frac{1}{2}\log\left[\frac{\hat{\sigma}^2_{i0}}{\hat{\sigma}^2_{ik}}\right], \end{aligned} \tag{7}$$

where $\hat{\sigma}^2_{i0}$ denotes prior variance, i.e. $k(\mathbf{x}_i, \mathbf{x}_i) + \sigma^2$.

2) Bayesian Committee Machine (BCM): Bayesian committee machine (BCM) is another approach to combine different estimators in consideration of the prior distribution of y_i [18]. The target probability is defined to be:

$$\begin{aligned} p(y_i|\mathbf{x}_i, \mathcal{D}) \doteq& \frac{1}{Z}\frac{\prod_{k=1}^{M} p^{\beta_{ik}}(y_i|\mathbf{x}_i, \mathcal{D}_k)}{p(y_i|\mathbf{x}_i)^{\sum_{k=1}^{M}\beta_{ik}-1}} \\ =& \frac{1}{Z}\frac{\prod_{k=1}^{M} \mathcal{N}^{\beta_{ik}}(y_i|\hat{\mu}_{ik}, \hat{\sigma}^2_{ik})}{\mathcal{N}(y_i|0, \sigma^2_{i0})^{\sum_{k=1}^{M}\beta_{ik}-1}}. \end{aligned} \tag{8}$$

Closed form of predictive means and variances can be derived similarly to PoE. Original BCM assumes $\beta_{ik} = 1, k = 1, 2, \cdots, M$. Inspired by GPoE, Deisenroth *et al.* proposed robust Bayesian committee machine (RBCM) in which β_{ik} is calculated in the same way of GPoE [21].

Generalized Robust Bayesian Committee Machine (GRBCM) is a variant of RBCM which employs a "communication subset" $\mathcal{D}_c$ to enlarge other subsets [22] and the final predictive distribution bears some resemblence with RBCM. A distinct advantage of GRBCM is that it has been rigorously shown to have consistency which other aggregation models (PoE, GPoE, BCM, RBCM) do not possess.

3) Nested Pointwise Aggregation of Experts (NPAE): NPAE [26] avoids the independent assumptions in BCM and aggregate multiple experts in consideration of all pairwise covariances between the sub-models. It provides a dedicated covariance parameter estimation procedure at the cost of greatly increased prediction complexity. Therefore we omit the comparison with NPAE in experiments.

C. Tsallis Entropy

As an important concept in many disciplines such as statistical mechanics, thermodynamics, and information theory, entropy measures the disorder of a system or uncertainty of an event [27]. The most well-known Shannon entropy takes the form $H(\xi) = -\sum_{i=1}^{n} p(\xi_i)\log p(\xi_i)$, where ξ denotes a random variable. The summation is over all possible states ξ_i and $p(\xi_i)$ is the corresponding probability [28]. Tsallis entropy generalizes this concept by introducing an adjustable entropic index q [29], and has a wide range of applications in statistical mechanics and thermodynamics [30]. Specifically, Tsallis entropy takes the following form, and it can be verified that when $q = 1$, Tsallis entropy is reduced to Shannon entropy:

$$S_q(\xi) = \frac{1}{1-q}\left(\sum_{i=1}^{n} p(\xi_i)^q - 1\right). \tag{9}$$

Recent studies explore the possibility of bridging the interactions between Tsallis entropy and machine learning. For example, Wang *et al* discovered that two popular splitting criteria (Gain ratio and Gini index) can prove to be special cases of Tsallis entropy and therefore proposed a unified framework of leaf splitting in the process of decision tree's construction [31], [32]. Besides, Tsallis entropy has also been exploited in reinforcement learning community. Entropy-regularized Markov decision processes (MDPs) forces the optimal policy to be stochastic via a Shannon entropy term and Lee *et al* recently proposed a Tsallis entropy based regularizer to induce a sparse and multi-modal optimal policy in MDPs [33]. However, whether can we exploit Tsallis entropy to facilitate Gaussian process aggregation models remains unknown.

III. GENERALIZED LOCAL AGGREGATION

In section 2, we review six popular local aggregation algorithms. Among them, GPoE, RBCM and GRBCM introduce a parameter β_{ik} to measure each GPR expert's importance. However, current weight calculations can not focus on performant experts which has been shown in Figure 1. In this section, we propose a generalized local aggregation framework with a novel calculation of β_{ik} based on Tsallis entropy. The new method outperforms previous work for the novel calculation forces a sparser weight distribution compared with Shannon based methods. Of particular note is that the sparsity of Tsallis entropy is also verified in some recent works [33], [34].

A. Proposed Method

For continuous probability distributions, Tsallis entropy is defined as:

$$S_q(\xi) = \frac{1}{1-q}\left(\int (p(\xi))^q \, d\xi - 1\right). \tag{10}$$

Algorithm 1 Vanilla Version of Generalized Local Aggregation

Input:
Data subsets $\{\mathcal{D}_1, ..., \mathcal{D}_M\}$, test input $\mathbf{x}_i$, kernel function $k(\mathbf{x}, \mathbf{x}')$, noise variance σ^2, extropic-index q.

Output:
Final predictive mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$.

Compute prior variance $\hat{\sigma}_{i0}^2 = k(\mathbf{x}_i, \mathbf{x}_i) + \sigma^2$.
for $k \in \{1, ..., M\}$ **do**
  Compute $\hat{\mu}_{ik}, \hat{\sigma}_{ik}^2$ according to equation (4).
  Compute β_{ik} according to equation (11).
end for
Compute $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ according to equation (5), (8) or other aggregation rules.

Our method employs Tsallis mutual information to represent each expert's weight. Tsallis mutual information between y_i and $\mathcal{D}_k$ conditioned on $\mathbf{x}_i$ can be computed as:

$$\begin{aligned}\beta_{ik} =& I_q(y_i; \mathcal{D}_k|\mathbf{x}_i) \\ =& S_q(y_i|\mathbf{x}_i) - S_q(y_i|\mathbf{x}_i, \mathcal{D}_k) \\ =& \frac{q^{\frac{1}{2}}(2\pi)^{\frac{1-q}{2}}}{1-q}\left[(\hat{\sigma}_{i0})^{1-q} - (\hat{\sigma}_{ik})^{1-q}\right].\end{aligned} \tag{11}$$

It is obvious to observe that:

$$\lim_{q\to 1} I_q(y_i; \mathcal{D}_k|\mathbf{x}_i) = \log\left[\frac{\hat{\sigma}_{i0}}{\hat{\sigma}_{ik}}\right] = I(y_i; \mathcal{D}_k|\mathbf{x}_i). \tag{12}$$

The derivation of β_{ik} based on Tsallis mutual information can be regarded as a generalization of previous calculations. Compared with previous works, our method enhances the model's flexibility and generate a sparser representation of expert's weight distribution, which has been shown in Figure 1. Different values of q implicitly characterize the sparsity of weight distribution which can not be expressed by Shannon mutual information. The Vanilla version of our algorithm is summarized in Algorithm 1.

B. Adaptive Entropic-index Optimization

The performance of the proposed method relies on a suitable choice of the entropic-index q. Since the optimal value of q varies from one dataset to another, a natural question to ask is how to adaptively adjust entropic-index dependent on the given dataset. In this paper, we propose three different strategies to solve this problem.

1) Grid search: This method simply exhaustively searches a subset of candidate q values, which are chosen uniformly from a fixed interval. Although this is the most trivial approach, it often works well in practice, and can serve as a baseline for the more sophisticated methods. We denote this method as ***grid-q***.

2) Optimize single q by gradient descent: In this method, the flexibility of the model is enhanced by introducing a parameter q that is shared by every expert. Then the optimal q value can be obtained by minimizing the objective loss with gradient descent or maximizing the sum of log conditional likelihoods with gradient ascent. In order to avoid trivial solutions, we also add a regularization term in the objective function. Specifically, assuming that there are M experts whose associated subsets are $\mathcal{D}_1, \cdots, \mathcal{D}_M$. We define $\mathcal{D}_{-i} \triangleq \{\mathcal{D}_1, \cdots, \mathcal{D}_{i-1}, \mathcal{D}_{i+1}, \cdots, \mathcal{D}_M\}$. For each data point $(\mathbf{x}_i, y_i) \in \mathcal{D}_i$, its loss is defined as the negative log-likelihood condition on all the other subsets plus a regularization term:

$$l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) = -\log p(y_i|q, \mathbf{x}_i, \mathcal{D}_{-i}) + \frac{1}{2}\lambda \sum_{\substack{k=1 \\ k\neq i}}^{M} \beta_{ik}^2. \tag{13}$$

The final objective loss to be minimized is defined as the sum of all the individual losses:

$$\begin{aligned} l(q) =& \sum_{i=1}^{M} \sum_{(\mathbf{x}_i, y_i)\in\mathcal{D}_i} l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\ =& \sum_{i=1}^{M} \sum_{(\mathbf{x}_i, y_i)\in\mathcal{D}_i} \left(-\log p(y_i|q, \mathbf{x}_i, \mathcal{D}_{-i}) + \frac{1}{2}\lambda \sum_{\substack{k=1 \\ k\neq i}}^{M} \beta_{ik}^2 \right). \end{aligned} \tag{14}$$

Note that the likelihood of the data in the i-th subset is conditioned on all the other subsets except itself. β_{ik} is a regularization term of data point $(\mathbf{x}_i, y_i)$ induced by the k-th subset, which is used to prevent experts being over-confident in their own predictions.

The optimal q value can be inferred by applying the gradient descent rule repeatedly: $q_{i+1} = q_i - \eta \frac{\partial l(q_i)}{\partial q}$, where η is the learning rate. The next step is to derive the analytical form of $\frac{\partial l(q)}{\partial q}$. By applying the chain rule, we have:

$$\begin{aligned} &\frac{\partial}{\partial q} l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\ &= \sum_{j=1, j\neq i}^{M} \left(\frac{\partial}{\partial \beta_{ij}} l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \frac{\partial \beta_{ij}}{\partial q} \right), \end{aligned} \tag{15}$$

The first term $\frac{\partial}{\partial \beta_{ij}} l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i})$ depends on the particular aggregation rule being used. For example, if the aggregation rule is RBCM which is given as (8), we have:

$$\begin{aligned} &\frac{\partial}{\partial \beta_{ij}} l(q, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\ =& \frac{\partial}{\partial \beta_{ij}} \left(-\log p^{(RBCM)}(y_i|q, \mathbf{x}_i, \mathcal{D}_{-i}) + \frac{1}{2}\lambda \sum_{\substack{k=1 \\ k\neq i}}^{M} \beta_{ik}^2 \right) \\ =& -\left(\log(\hat{\sigma}_{i0}) - \log(\hat{\sigma}_{ij}) + \frac{y_i^2}{2\hat{\sigma}_{i0}^2} - \frac{(y_j - \hat{\mu}_{ij})^2}{2\hat{\sigma}_{ij}^2} \right) + \lambda\beta_{ij}, \end{aligned} \tag{16}$$

where $\hat{\mu}_{ij}$ and $\hat{\sigma}_{ij}^2$ denote the predictive mean and variance of the j-th expert respectively, $\hat{\sigma}_{i0}^2$ denotes the prior variance. The second term $\frac{\partial \beta_{ij}}{\partial q}$ can also be analytically derived:

Algorithm 2 ***single-q*** for RBCM aggregation

Input:

input-output pair $(\mathbf{x}_i, y_i)$, Data subsets $\{\mathcal{D}_1, \cdots, \mathcal{D}_M\}$, predictions $\{(\hat{\mu}_{i1}, \hat{\sigma}^2_{i1}), \cdots, (\hat{\mu}_{iM}, \hat{\sigma}^2_{iM})\}$ from every expert,
iteration step $iter$, regularization parameter λ, learning rate η

Output:

q^* optimized by gradient descent

$n \leftarrow 0$
Randomly initialize $q^{(n)}$
while $n < iter$ **do**
  Randomly select $\mathcal{D}_i$ (the probability of selecting $\mathcal{D}_i$ is $\frac{|\mathcal{D}_i|}{\sum_{j=1}^{M}|\mathcal{D}_j|}$)
  Randomly select $(\mathbf{x}_i, y_i) \in \mathcal{D}_i$
  for $j \in \{1, ..., M\}$ **do**
    if $j \neq i$ **then**
      Compute $\frac{\partial}{\partial \beta_{ij}} l(q^{(n)}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i})$ according to equation (16)
      Compute $\frac{\partial \beta_{ij}}{\partial q^{(n)}}$ according to equation (17)
    end if
  end for
  $\frac{\partial l(q^{(n)})}{\partial q^{(n)}} \leftarrow \sum_{j=1, j\neq i}^{M} \frac{\partial}{\partial \beta_{ij}} l(q^{(n)}, (\mathbf{x}_i, y_i)_{,-i}) \frac{\partial \beta_{ij}}{\partial q^{(n)}}$ (by equation (14) and equation (15))
  $q^{(n+1)} \leftarrow q^{(n)} - \eta \frac{\partial l(q^{(n)})}{\partial q^{(n)}}$
  $n \leftarrow n + 1$
end while
$q^* \leftarrow q^{(n-1)}$

$$
\begin{aligned}
\frac{\partial \beta_{ij}}{\partial q} =& \frac{1}{(q-1)^2 q^{3/2}} 2^{-q-\frac{1}{2}} \pi^{\frac{1}{2}-q} \hat{\sigma}_{ij}^{-q} \hat{\sigma}_{i0}^{-q} (A + B) \\
A =& -\left(\hat{\sigma}_{ij}\hat{\sigma}_{i0}^{q} - \hat{\sigma}_{i0}\hat{\sigma}_{ij}^{q}\right) \times \\
& \left(2^{\frac{q}{2}+1}\pi^{q/2} q + (2\pi)^{q/2}(q-1)(q\log(2\pi)+1)\right) \\
B =& -2^{\frac{q}{2}+1}\pi^{q/2}(q-1)q \times \\
& \left(\hat{\sigma}_{ij}\hat{\sigma}_{i0}^{q}\log(\hat{\sigma}_{ij}) - \hat{\sigma}_{i0}\hat{\sigma}_{ij}^{q}\log(\hat{\sigma}_{i0})\right).
\end{aligned} \tag{17}
$$

The correctness of this equation can be verified by mathematical software such as Mathematica, or numerical test ($\frac{\partial \beta_{ij}}{\partial q} \approx \frac{\beta_{ij}(q+\Delta q) - \beta_{ij}(q)}{\Delta q}$ for small Δq). We denote this method as ***single-q***.

3) Optimize multiple q's by gradient descent: This method is similar to 2). The key difference is that instead of sharing the same q across all the experts, now each expert is associated with an individual q_i. Similarly, for each data point $(\mathbf{x}_i, y_i) \in \mathcal{D}_i$, its loss takes the form

$$
\begin{aligned}
& l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\
&= -\log p(y_i | q_{1...M}, \mathbf{x}_i, \mathcal{D}_{-i}) + \frac{1}{2}\lambda \sum_{\substack{k=1 \\ k\neq i}}^{M} \beta_{ik}^2.
\end{aligned} \tag{18}
$$

Here we use the notation $q_{1...M}$ as the abbreviation of the sequence $q_1, ..., q_M$. The final objective loss is:

$$
\begin{aligned}
& l(q_1, ..., q_M) \\
=& \sum_{i=1}^{M} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_i} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\
=& \sum_{i=1}^{M} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_i} \left(-\log p(y_i | q_{1...M}, \mathbf{x}_i, \mathcal{D}_{-i}) + \frac{1}{2}\lambda \sum_{\substack{k=1 \\ k\neq i}}^{M} \beta_{ik}^2 \right).
\end{aligned} \tag{19}
$$

Note that q_j is the only free variable in β_{ij}. By similar reasoning when $j \neq i$, we have:

$$
\begin{aligned}
& \frac{\partial}{\partial q_j} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \\
=& \frac{\partial}{\partial \beta_{ij}} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \frac{\partial \beta_{ij}}{\partial q_j}.
\end{aligned} \tag{20}
$$

The method for computing $\frac{\partial \beta_{ij}}{\partial q_j}$ and that of computing $\frac{\partial}{\partial \beta_{ij}} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i})$ remain the same. We denote this method as ***multi-q***.

As concrete examples, Algorithm 2 and 3 summarize the details of applying the ***single-q*** and ***multi-q*** methods in RBCM aggregation respectively.

C. Complexity

Assuming that each expert has equal number of training data. We denote n as the size of the training dataset, m_0 as the size of each subset, n' as the size of test set, t as the iteration steps in the gradient optimization of q. NPAE scales poorly with $\mathcal{O}(n'n^2)$. Our models have the same complexity with Shannon entropy based models in calculating β_i, the measure of importance for each expert. Hence the time complexity in prediction process is exactly the same as previous methods, which scales as $\mathcal{O}(nm_0^2) + \mathcal{O}(n'nm_0)$ for (G)PoE and (R)BCM. Certainly the optimization of the parameter q introduces extra computational cost which scales as $\mathcal{O}(tnm_0)$ but t is relatively small and usually less than 10000. Therefore the overall complexity of our methods scale as $\mathcal{O}(nm_0^2) + \mathcal{O}(n'nm_0) + \mathcal{O}(tnm_0)$.

IV. EXPERIMENTS

In this section, we compare our methods with three popular aggregation approaches: GPoE, RBCM, GRBCM. As discussed in the previous sections, the above models employ Shannon mutual information to balance each expert's importance. By generalizing them with Tsallis entropy, we denote our methods as TEGPoE, TERBCM, TEGRBCM respectively.

The assessment criteria include Standard Mean Square Error (SMSE) and Mean Standardized Log Loss (MSLL) [1]. SMSE can be obtained to normalize MSE by the variance of the targets of the test cases. Next we mainly introduce another criteria: MSLL.

Algorithm 3 ***multi-q*** for RBCM aggregation

Input:

input-output pair $(\mathbf{x}_i, y_i)$, Data subsets $\{\mathcal{D}_1, \cdots, \mathcal{D}_M\}$, predictions $\{(\hat{\mu}_{i1}, \hat{\sigma}^2_{i1}), \cdots, (\hat{\mu}_{iM}, \hat{\sigma}^2_{iM})\}$ from every expert,

iteration step $iter$, regularization parameter λ, learning rate η

Output:

$\{q^*_k, k = 1, 2, \cdots, M\}$ optimized by gradient descent

$n \leftarrow 0$
Randomly initialize $q^{(n)}_k, k = 1, 2, \cdots, M$
while $n < iter$ **do**
  Randomly select $\mathcal{D}_i$ (the probability of selecting $\mathcal{D}_i$ is $\frac{|\mathcal{D}_i|}{\sum_{j=1}^{M}|\mathcal{D}_j|}$)
  Randomly select $(\mathbf{x}_i, y_i) \in \mathcal{D}_i$
  for $j \in \{1, ..., M\}$ **do**
    if $j \neq i$ **then**
      Compute $\frac{\partial}{\partial \beta_{ij}} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i})$
      Compute $\frac{\partial \beta_{ij}}{\partial q^{(n)}_j}$
      $\frac{\partial}{\partial q^{(n)}_j} l(q^{(n)}_{1...M}) \leftarrow$
        $\frac{\partial}{\partial \beta_{ij}} l(q_{1...M}, (\mathbf{x}_i, y_i), \mathcal{D}_{-i}) \frac{\partial \beta_{ij}}{\partial q^{(n)}_j}$
      $q^{(n+1)}_j \leftarrow q^{(n)}_j - \eta \frac{\partial}{\partial q^{(n)}_j} l(q^{(n)}_{1...M})$
    end if
  end for
  $n \leftarrow n + 1$
end while
for $j \in \{1, \cdots, M\}$ **do**
  $q^*_j \leftarrow q^{(n-1)}_j$
end for

As GPR is a probabilistic model, one obtains the negative log predictive density:

$$-\log p(y_*|\mathbf{x}_*, \mathcal{D}) = \frac{1}{2}\log(2\pi\sigma^2_*) + \frac{(y_* - \mu_*)^2}{2\sigma^2_*}. \quad (21)$$

Another trivial density estimation of y_* is Gaussian distribution with the mean and variance of training data. The difference between the two negative log predictive densities measures the accuracy of predictive variances. This can be denoted as Mean Standardized Log Loss (MSLL) by averaging over the test cases. MSLL can be negative and smaller means better.

A. Synthetic Datasets

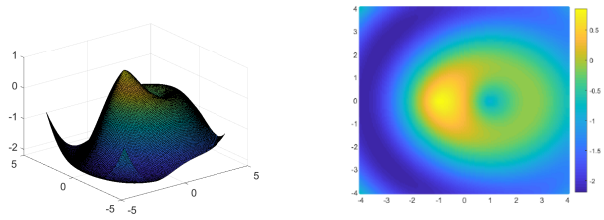
In this section, we conduct experiments on a two-dimensional synthetic dataset to show that different q's values have huge impacts on the behaviors of Gaussian process aggregation models, and it is possible to make great improvements in the predictive accuracy by choosing a suitable q. Considering the function $f(x, y) = \sin\sqrt{(x-1)^2 + y^2} - \sin\sqrt{0.5(x+1)^2 + y^2} - \sin 0.05\sqrt{x^2 + y^2}$, we generate 160×160 uniformly distributed instances (x, y) within the interval $[0, 1] \times [0, 1]$ as the training input . The corresponding outputs z's values are produced via $z = f(x, y) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.04)$ denotes $i.i.d.$ random noise.

q's value	SMSE	MSLL
$q = 1$	0.0665	0.0242
$q = 2$	0.0652	-0.8996
$q = 3$	0.0643	-1.1135
$q = 4$	0.0638	-1.1359

TABLE I: Results of synthetic datasets.

Datasets	Training/Test Size	Dimensions	Area
kin40k	10000/30000	8	Robotics
sarcos	44484/4449	21	Robotics
energy	18300/1375	27	Computer
protein	40000/5730	8	Life
network	400000/34874	3	Computer

TABLE II: Description of the datasets.



(a) $f(x, y)$ (b) Vertical View of $f(x, y)$

Fig. 2: Two-dimensional synthetic datasets.

We select $[0, 0]$ as the test input and use k-means [35] to partition the training dataset into 50 groups then experiment with different q's values to see how they influence the value of each expert's weight. Here we report the results of $q = 1, 2, 3, 4$ as depicted in Figure 1 where each sub-block indicates an expert and their weights are displayed by shade of color . Of particular note is when $q = 1$ the models degenerate to Shannon-entropy-based aggregation models, the results of which are showed in Figure 1. We can observe that the weight distribution dependent on Tsallis entropy is sparser than Shannon case. Total 2000 test are generated according to $(x, y) \sim U(-4, 4) \times U(-4, 4), z = f(x, y)$ and the experimental results are summarized in Table 1, from which we can observe that previous methods of calculating weights ($q = 1$) usually can not generate optimal predictive accuracy, whereas Tsallis entropy based calculating leads to a more satisfactory result.

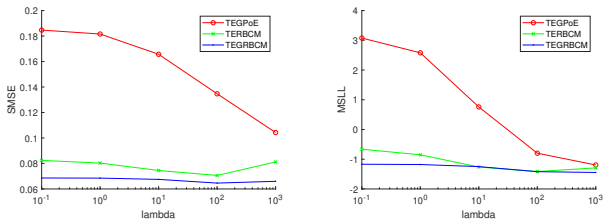
B. Realistic Datasets

We evaluate our models' performance on five realistic datasets: *kin40k* [22], *sarcos* [1], *energy* [36] , *protein* [37] and *network* [38] . Detailed descriptions about the datasets are listed in Table 2.

In order to allow the algorithms to converge faster, and also find better local minimums, we run ***grid-q*** first and then use the optimal q's values as the initial points to guide the search in ***single-q*** and ***multi-q***. The experimental procedure is as follows: 1. We run the ***grid-q*** algorithm and select two different q's values: q_1 corresponds to the optimal MSLL and q_2 corresponds to the optimal MSE.2. Use q_1 and q_2 as different initial q's values, and run the ***single-q*** and ***multi-q*** algorithms.

Experimental results have been summarized in Table 3-5. We can observe that almost all Tsallis entropy based models outperform the Shannon entropy based models, which verifies our claim that the weights calculated with Tsallis mutual information tend to capture more explicit interactions behind data than Shannon case.

Specifically, for SMSE criteria, TEGPoE achieves the optimal performance for *protein* dataset, TERBCM shows advantage on *sarcos* dataset, and TEGRBCM performs best for *kin40k* dataset. So in terms of SMSE, all the three algorithm are equally competitive after being enhanced by Tsallis entropy. Our method demonstrates obvious advantage for MSLL criteria. For *kin40k* dataset, all the three methods significantly reduce MSLL scores, particularly the TEGPoE and TERBCM variants. The same tendency can also be observed in many results across other datasets. For all the experiments, the best MSLL results are always achieved by TEGRBCM. In many cases, our method can greatly improve the performance of TEGPoE and TERBCM with respect to their baselines, so that their gaps between TEGRBCM are notably reduced. As mentioned above, smaller values of MSLL means more explicit predictive uncertainty, which is a core advantage of Gaussian process models.



(a) SMSE (b) MSLL

Fig. 3: Sensitivity analysis of hyperparameter λ.

We also perform experiments to show that the proposed algorithm is relatively insensitive to the specific choice of the hyperparameter λ. We report the experimental results on ***multi-q*** algorithm on the *energy* dataset, though similar phenomenon can be observed across other domains. As we can see in Figure 3, while varying the λ value in the range of $[10^{-1}, 10^{3}]$ both the two criterions MSE and MSLL appear to be stable, especially when λ is in the range $[10^{-1}, 10^{1}]$.

V. CONCLUSION

In this paper, we aim to improve the effectiveness of weight calculating for Gaussian process aggregation models. Current aggregation models compute each expert's weight using Shannon mutual information. As a generalization of Shannon entropy, Tsallis entropy tends to characterize the implicit relationships among data via parameter q . Inspired

Model	SMSE					MSLL				
	kin40k	*sarcos*	*energy*	*protein*	*network*	*kin40k*	*sarcos*	*energy*	*protein*	*network*
GPoE	0.1887	0.0112	**0.0001**	0.3700	**0.0091**	3.8437	1.0700	19.7039	-0.2425	-2.7155
TEGPoE(*grid-q*)	**0.1500**	**0.0068**	**0.0001**	**0.3635**	**0.0091**	**-0.2535**	**-0.9777**	**19.3985**	**-0.3888**	**-2.7452**
TEGPoE(*single-q*)	**0.1498**	**0.0068**	**0.0001**	**0.3660**	**0.0091**	**-0.2605**	**-0.9807**	**19.3593**	**-0.3889**	**-2.7530**
TEGPoE(*multi-q*)	**0.1487**	**0.0067**	**0.0001**	**0.3634**	**0.0091**	**-0.3113**	**-0.9766**	**18.9144**	**-0.3910**	**-2.7452**

TABLE III: Comparison of GPoE and TEGPoE. The boldface terms refer to top three results among all conducted methods. SMSE and MSLL are two assessment criteria and smaller means better.

Model	SMSE					MSLL				
	kin40k	*sarcos*	*energy*	*protein*	*network*	*kin40k*	*sarcos*	*energy*	*protein*	*network*
RBCM	0.0802	0.0082	**0.0001**	**0.4140**	0.0140	-0.8009	-0.1519	18.4559	-0.4879	**-2.7244**
TERBCM(*grid-q*)	**0.0722**	**0.0063**	**0.0001**	**0.3834**	**0.0103**	**-1.3882**	**-1.2087**	**18.4559**	**-0.5430**	-2.3910
TERBCM(*single-q*)	**0.0722**	**0.0063**	**0.0001**	0.4530	**0.0115**	**-1.3888**	**-1.2087**	**18.4405**	**-0.5430**	**-2.6767**
TERBCM(*multi-q*)	**0.0723**	**0.0063**	**0.0001**	**0.3783**	**0.0124**	**-1.3880**	**-1.1959**	**18.3953**	**-0.5433**	**-2.7545**

TABLE IV: Comparison of RBCM and TERBCM.

Model	SMSE					MSLL				
	kin40k	*sarcos*	*energy*	*protein*	*network*	*kin40k*	*sarcos*	*energy*	*protein*	*network*
GRBCM	0.0685	**0.0073**	**0.0001**	**0.4088**	0.0139	-1.1745	**-2.1450**	0.6241	-0.5389	**-2.7256**
TEGRBCM(*grid-q*)	**0.0639**	**0.0066**	**0.0001**	**0.3783**	**0.0103**	**-1.4548**	**-2.1450**	**-3.9794**	**-0.5629**	-2.3949
TEGRBCM(*single-q*)	**0.0639**	0.0074	**0.0001**	0.4331	**0.0115**	**-1.4548**	-2.1457	**-3.9792**	**-0.5629**	**-2.6789**
TEGRBCM(*multi-q*)	**0.0639**	**0.0066**	**0.0001**	**0.3763**	**0.0124**	**-1.4549**	**-2.1440**	**-3.9786**	**-0.5628**	**-2.7560**

TABLE V: Comparison of GRBCM and TEGRBCM.

by this, we propose generalized local aggregation models and demonstrate its validity and effectiveness in both synthetic and realistic datatsets. To adjust entropic-index to varying datasets, we propose three heuristic algorithms to solve our model. Extensive experiments show that, the proposed method can generate a sparse and effective representation of each expert's weight and improve both the mean and variance predictions, and in many cases the improvement of variance prediction is significant. We will continue to explore more effective ways to define entropic-index in future work.

REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, 2006.

[2] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012.

[3] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in Neural Information Processing Systems*, 2008.

[4] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[5] M. Kuss and C. E. Rasmussen, "Gaussian processes in reinforcement learning," in *Advances in neural information processing systems*, 2004, pp. 751–758.

[6] R. Herbrich, N. D. Lawrence, and M. Seeger, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems*, 2003.

[7] S. Keerthi and W. Chu, "A matching pursuit approach to sparse gaussian process regression," in *Advances in Neural Information Processing Systems*, 2006.

[8] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, 2006.

[9] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Artificial Intelligence and Statistics*, 2009.

[10] A. G. d. G. Matthews, J. Hensman, R. Turner, and Z. Ghahramani, "On sparse variational methods and the kullback-leibler divergence between stochastic processes," in *Artificial Intelligence and Statistics*, 2016.

[11] D. Burt, C. E. Rasmussen, and M. Van Der Wilk, "Rates of convergence for sparse variational Gaussian process regression," in *International Conference on Machine Learning*, 2019.

[12] D. Moore and S. J. Russell, "Gaussian process random fields," in *Advances in Neural Information Processing Systems*, 2015.

[13] E. Snelson and Z. Ghahramani, "Local and global sparse gaussian process approximations," in *Artificial Intelligence and Statistics*, 2007, pp. 524–531.

[14] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A distributed variational inference framework for unifying parallel sparse gaussian process regression models." in *International Conference on Machine Learning*, 2016, pp. 382–391.

[15] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," in *Advances in Neural Information Processing Systems*, 2017, pp. 4588–4599.

[16] M. Tavassolipour, S. A. Motahari, and M. T. M. Shalmani, "Learning of gaussian processes in distributed and communication limited systems," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[17] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *arXiv preprint arXiv:1807.01065*, 2018.

[18] V. Tresp, "A bayesian committee machine," *Neural computation*, 2000.

[19] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, 2002.

[20] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.

[21] M. Deisenroth and J. W. Ng, "Distributed gaussian processes," in *International Conference on Machine Learning*, 2015.

[22] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust bayesian committee machine for large-scale gaussian process regression," in *International Conference on Machine Learning*, 2018.

[23] C. M. Bishop, *Pattern recognition and machine learning, 5th Edition*, 2007.

[24] Q. Tang, Y. Wang, and S.-T. Xia, "Student-t process regression with dependent student-t noise," in *European Conference on Artificial Intelligence*, 2016.

[25] Q. Tang, L. Niu, Y. Wang, T. Dai, W. An, J. Cai, and S.-T. Xia, "Student-t process regression with student-t likelihood." in *International Joint Conferences on Artificial Intelligence*, 2017.

[26] D. Rullière, N. Durrande, F. Bachoc, and C. Chevalier, "Nested kriging predictions for datasets with a large number of observations," *Statistics and Computing*, 2018.

[27] R. Frigg and C. Werndl, "Entropy - a guide for the perplexed," *Probabilities in Physics*, 2011.

[28] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 1948.

[29] C. Tsallis, "Possible generalization of boltzmann-gibbs statistics," *Journal of Statistical Physics*, 1988.

[30] ——, *Introduction to nonextensive statistical mechanics: approaching a complex world*, 2009.

[31] Y. Wang, C. Song, and S.-T. Xia, "Unifying decision trees split criteria using tsallis entropy," *arXiv preprint arXiv:1511.08136*, 2015.

[32] ——, "Improving decision trees by tsallis entropy information metric method," in *International Joint Conference on Neural Networks*, 2016.

[33] K. Lee, S. Choi, and S. Oh, "Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning," *IEEE Robotics and Automation Letters*, 2018.

[34] W. Yang, X. Li, and Z. Zhang, "A regularized approach to sparse optimal policy in reinforcement learning," in *Advances in Neural Information Processing Systems*, 2019.

[35] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, 2010.

[36] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*.

[37] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[38] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul, "Ecomark: evaluating models of vehicular environmental impact," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012.