Kernelized Rényi distance for subset selection and similarity scoring

Balaji Vasan Srinivasan and Ramani Duraiswami

Perceptual Interfaces and Reality Laboratory, Department of Computer Science,
University of Maryland, College Park, MD, USA
E-mail: [balajiv,ramani]@umiacs.umd.edu

Abstract

Rényi entropy refers to a generalized class of entropies that have been used in several applications. In this work, we derive a non-parametric distance between distributions based on the quadratic Rényi entropy. The distributions are estimated via Parzen density estimates. The quadratic complexity of the distance evaluation is mitigated with GPU-based parallelization. This results in an efficiently evaluated non-parametric entropic distance - the kernelized Rényi distance or the KRD. We adapt the KRD into a similarity measure and show its application to speaker recognition. We further extend KRD to measure dissimilarities between distributions and illustrate its applications to statistical subset selection and dictionary learning for object recognition and pose estimation.

Keywords: Rényi entropy, graphical processors, similarity scores, speaker recognition, subset selection, Gaussian process regression, object recognition, pose estimation

1. Introduction

The entropy of a distribution measures the amount of information contained by the distribution. The *Shannon entropy* is the most widely used entropic measure. For a random variable X whose probability distribution is $p(x)$, the Shannon entropy is given by,

$$H(X) = -\int p(x)\log p(x)dx \tag{1}$$

The Shannon Entropy is a specific case of a more generalized family of *Rényi entropies* characterized by a parameter α. The Rényi entropy of order α ($\alpha \geq 0$) is given by

$$H_\alpha(x) = \frac{1}{1-\alpha}\log\int p(x)^\alpha dx \tag{2}$$

As $\alpha \to 1$, the Rényi entropy reduces to the Shannon entropy (Eq. 1) in the limits as shown in [1]. The Shannon entropy of a joint probability distribution can be separated into the entropies of the individual random variables of the joint distribution. These properties, coupled with the analytical tractability of the Shannon measures for the commonly encountered parametric distributions, has made it the preferred choice for many problems. Despite this advantage, the Shannon entropy may be suboptimal in certain applications that require entropy estimation from samples [2].

Sample-based entropy estimation generally involves the pdf estimation ($p(x)$) followed by the entropy-integral approximation ($H(X)$ or $H_\alpha(x)$). The pdf estimation is much harder at higher dimensions, leading to an inconsistent entropy estimate which can be detrimental to the underlying application. However, it has been shown that for a quadratic Rényi entropy ($\alpha = 2$), the pdf-estimation step can be bypassed by directly solving the integral with a kernel density estimate plug-in [1]. This results in a consistent estimator even for higher dimension, as is illustrated later in Section 2. Motivated by this, we consider the quadratic Rényi entropy and solve the integral with a kernel density estimate plug-in following [1]. We adapt the resulting distance measure to problems in speaker recognition, object recognition and pose estimation; improvements are seen in each case. Throughout this paper the term *Rényi entropy* will refer to the quadratic Rényi entropy ($\alpha = 2$).

The paper[1] is organized as follows. In Section 2, we present expressions for the non-parametric quadratic Rényi entropy using the kernel density plug-in as well as an expression for the distance between two distribution which we call the *kernelized Rényi distance (KRD)* measure. We illustrate the inconsistency in sample based estimation of KL divergence and show empirically that this is absent for the KRD measure. We finally discuss the acceleration strategies to mitigate the $O(N^2)$ computational cost of KRD evaluation between two distributions obtained from $O(N)$ samples. In Section 3, we adapt the KRD into a similarity measure and show its application to a speaker recognition problem. We adapt the KRD to a dissimilarity measure for a low rank subset selection problem in Section 4. We develop a greedy algorithm based on the KRD, validate the algorithm and apply the algorithm to Gaussian

[1] This paper synthesizes and extends results which were presented in [3] and [4].

process regression and object recognition. Section 5 concludes the paper.

2. Kernelized Rényi Distance (KRD)

The quadratic Rényi entropy (for $\alpha = 2$ in Eq. 2) is given by,

$$H_2(x) = -\log \int p(x)^2 dx. \tag{3}$$

If $p(x)$ is known, the entropy can be computed using the integral above. In many practical scenarios, the density is unknown, and must be estimated from samples drawn from the distribution. There are parametric and non-parametric ways of estimating the density function. In the parametric case, a particular form for the density is assumed and the parameters associated with the form are estimated from the samples, e.g. via the expectation-maximization algorithm. A non-parametric approach to density estimation uses a kernel window and estimates the density as a sum of kernel functions of the available samples from the distribution. Using kernel density estimation for $p(x)$ as in [5], we get

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x, x_i), \tag{4}$$

x_i indicates the sample location, $K_h(x, x_i)$ is a kernel function, quite often the Gaussian kernel,

$$K_h(x_1, x_2) = \frac{1}{h\sqrt{2\pi}} \exp\left(-\frac{|x_1 - x_2|^2}{h^2}\right), \tag{5}$$

with h the bandwidth that must be selected according to the data. This approach is preferred when the underlying distribution is unknown. Provided there are sufficient samples, a non-parametric approach provides unbiased estimates. Plugging-in Eq. (4) for $p(x)$ to Eq. (3), we get

$$\begin{aligned} H_2(x) &= -\log \int \left(\frac{1}{N}\sum_{i=1}^{N} K_h(x, x_i)\right)^2 dx \qquad (6) \\ &= -\log \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \int K_h(x, x_i) K_h(x, x_j) dx. \end{aligned}$$

For the Gaussian kernel,

$$\int K_h(x, x_i) K_h(x, x_j) dx = \hat{K}_{\hat{h}}(x_i, x_j), \tag{7}$$

where $\hat{K}$ is also a Gaussian kernel with bandwidth equalling sum of the bandwidths of the two Gaussian kernels [6]. Using this relation in Eq. (6),

$$H_2(x) = -\log\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{K}_{\hat{h}}(x_i, x_j)\right). \tag{8}$$

Consider two distinct distributions with densities p and q, with p defined by the set of data points, $D_p = \{x_{p1}, \ldots, x_{pN}\}$ and q defined by the set of data-points, $D_q = \{x_{q1}, \ldots, x_{qM}\}$, the distance between $p(x)$ and $q(x)$ is,

$$H_2(p\|q) = -\log\left(\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \hat{K}_{\hat{h}}(x_{pi}, x_{qj})\right). \tag{9}$$

This is called Rényi cross-information potential [1]. This was first defined and analyzed by Principe et al. [1] and has since been used in several applications including clustering [7], visual tracking [8] and source separation [9]. We shall refer to this measure (Eq. 9) as the **Kernelized Rényi Distance (KRD)**. The KRD is symmetric and non-parametric. However, it is not a complete distance, because it does not satisfy triangular inequality.

Accelerating KRD evaluation: The practical use of KRD is hindered by its memory and computational complexity. Evaluating the KRD between two distributions, each represented by N data-points, would require $(\mathrm{O}(N^2))$ operations. It should be noted that the core computation in Eq. (9) is the summation of the Gaussian kernel. There are two main approaches to accelerate the summation; we discuss these briefly here.

ϵ-exact approximations exist to evaluate Eq. 9 in $\mathrm{O}(N)$, e.g. FIGTREE [10]. These algorithms evaluate the KRD so that the error (absolute/relative) is $\leq \epsilon$ in some norm. The key idea here is to utilize the structure of the problem along with special data structures to efficiently approximate the sum in $\mathrm{O}(N \log N)$ time. The advantage of these accelerations is that the computational complexity can be linear $(\mathrm{O}(N))$. However, this performance is data-dependent and perform very badly at large data dimensions (> 10).

The other class of acceleration approaches include parallelizing the summation on multiple-cores, example GPUML [11]. The asymptotic computation complexity is still $\mathrm{O}(N^2)$ but the availability of computational resources lead these approaches to give comparable if not better accelerations compared to the approximation algorithms for some problems [11]. These approaches are effective upto atleast 100 dimensions (more careful strategies can yield good accelerations even beyond this dimension). We therefore chose GPUML to accelerated KRD evaluations in our paper.

Inconsistency of sample-based KL divergence Gockay et al. [12] observe that sample based estimation of the KL-divergence exhibits variability at higher dimensions because it is ratio-based (other ratio-based distances like Chernoff distance are also inconsistent at higher dimensions for sample based estimation). In this experiment, we illustrate this fact by using synthetic data and also show that the KRD measure (Eq. 9) does not exhibit such inconsistency.

In this experiment, we generated 10,000 samples from two Gaussian distributions, $N(\mu, 0.25\mathbf{I})$ and $N(-\mu, 0.25\mathbf{I})$, where $\mu = \{1, \ldots, 1\}$ and $\mathbf{I}$ the identity matrix, for various data dimensions. KL divergence between two Gaussian distributions with means μ_1 and μ_2 and variances Σ_1 and Σ_2 is given by,

$$\begin{aligned} KL(p||q) &= \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} + \frac{1}{2}tr\left[\Sigma_1(\Sigma_1^{-1} - \Sigma_2^{-1})\right] \\ &+ \frac{1}{2}tr\left[\Sigma_2^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\right] \end{aligned} \quad (10)$$

This distance is made symmetric by taking the average of $KL(p||q)$ and $KL(q||p)$. Similarly the quadratic Rényi cross entropy between two Gaussian distribution is given by,

$$KRD(p||q) = \mathcal{N}(\mu_2|\mu_1, \Sigma_1 + \Sigma_2) \quad (11)$$

where $\mathcal{N}(x|\mu, \Sigma)$ is the evaluation of the Gaussian distribution with mean μ and variance Σ evaluated at x [13].

We evaluate the KRD between samples for all the dimensions along with the KL divergence based on the samples. For comparison we also evaluate the KL-divergence and quadratic Rényi entropic distance between the distributions based on the first and second order statistics. As the dimension increases, the distance between distribution increases (as the means of the Gaussian are now more and more far placed) and is expected to be reflected in the corresponding measures. The normalized distance scores across dimension for various sample sizes (N) is shown in Fig. 1.

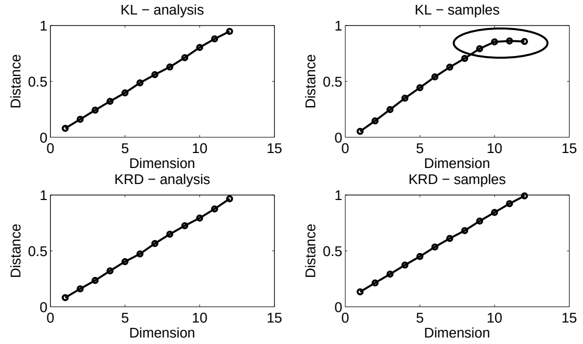


Figure 1: Validation of the Kernelized Renyi Distance; Entropic distances between Gaussian distribution for various dimensions, distances evaluated analytically based on the underlying distribution and from samples (based on density estimates)

It can be seen that the trend followed by the sample-based KRD score compares favorably with the statistics-based distance. However the trend of the sample-based KL divergence is skewed at higher dimensions illustrating the inconsistency. The variances of the corresponding KL sample-based distances are shown in Fig. 2, which indicates the associated instability. The variance of the other measures were $< 10^{-7}$ across several trials.

It was observed that the sample based KL divergence estimates do exhibit the desired trend when estimated from a very large number of samples ($\sim 75,000$ samples for 15 dimensions). However, in a practical scenario, this critical sample size required to remove the underlying inconsistency in the trend is either unknown or is beyond the modeler's control.

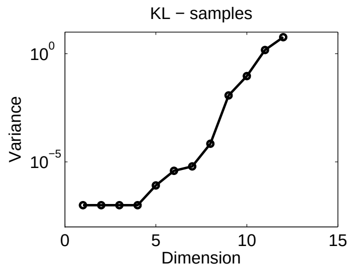


Figure 2: Variance of the KL based on sample-based estimates

3. KRD for similarity measurement

We first explore the application of KRD in Eq. 9 as an inter-class similarity measure (each class represented by a set of feature points) in the context of speaker recognition.

Fig. 3 shows a generic text-independent speaker recognition system that will be used in this paper. Once a speech signal is available, the first step in any recognition system is to extract features vectors from the signals. Once features are extracted, there are many approaches to build the speaker model. Gaussian Mixture Models (GMM) [14] build a semi-parametric model in feature space, and are one of the widely used approaches in speaker recognition. Alternatively, it is possible to measure the distance between feature vectors from the reference and test signals [15], and is the approach followed here. An advantage of such an approach is very low training time.

There have been several information-theoretic and statistical measures that have been used to measure scores between speech signals. Second-order statistical measures [15] like sphericity and Gaussian likelihood have been used in speaker identification, which use only the mean and variance of feature vectors. Soong et al. [16] use a vector quantizer based codebook along with the Euclidean distance to compare speech signals. Information theoretic measures like KL-divergence and Bhattacharya distance have also been used in the speaker recognition framework [17]. However, the underlying feature distributions are assumed to be Gaussian in all these works. This can be limiting when the underlying distribution is non-Gaussian. Semi-parametric Gaussian mixture models [14] address this issue to some extent, and are widely used in speaker recognition. A disadvantage with semiparametric and non-parametric approaches is the associated computational complexity, which make them undesirable for large problems. But however, in the KRD measure (Eq. 9), we have already addressed the computational complexity using GPUs.

To use KRD as a scoring function in speaker recognition (Fig. 3), it is necessary to formulate the speech

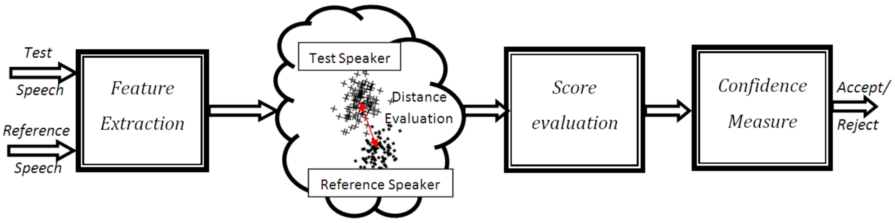


Figure 3: *A modular representation of a generic speaker recognition system*

signals (reference and test) as samples drawn from distributions. The feature selection in the recognition system extracts features from multiple overlapping frames of the speech signal. Suppose there are N and M overlapping frames in the reference and the test signal respectively, and d features are extracted, then we will have $N \times d$ vector representing the reference signal, and a $M \times d$ vector representing the test signal. We formulate this feature set to be samples drawn from the corresponding feature distribution of the speaker. This would make sense in a text-independent speaker recognition framework because the order of the features does not matter in this case. Using Eq. (9), we can thus evaluate the matching score.

For this experiment, we used the speech signals from the TIMIT [18] database, which consists of data from 630 different speakers. Each sample for a speaker contains one sentence uttered by the speaker and there are totally 10 samples per speaker. We extracted 13 mel-frequency cepstral coefficients (MFCC) coefficients from 25ms speech frames with 10ms overlap [19]. For all our experiments, the features were centralized and normalized to unit variance (except for the approaches that used only the first and second order statistics of the feature vectors). The method is of course generic enough to be used with other features.

Experiment 1 - Speaker Verification: Speaker verification system accepts a sample X as a speaker S if the likelihood ratio $\frac{P(X|S)}{P(X|S')} > T$, where T denotes a threshold. The likelihood $P(X|S)$ denotes the probability that the features from the sample X were generated by speaker S. Similarly, $P(X|S')$ denote the probability the features are from an imposter. The threshold T can be adjusted so that the false acceptance rate (FAR) (an imposter being identified as a speaker) and the false rejection rate (FRR) (a speaker being rejected as an imposter) are equal. We used this Equal Error Rate (EER) criterion to evaluate the performance of our measure.

We compared our scoring function with the Gaussian-likelihood measure [15] (GaussLL), Euclidean distance between vector quantized codebook [16] (VQ), KL-based measure [17] (KL_a), KL-scores evaluated from the samples (KL_s), and GMM-UBM based score [14]. The Matlab *kmeans* function was used to build the codebook of size 50. The GMM was built using statistical toolbox in Matlab, and number of mixtures was chosen to be 32 with diagonal covariance for each speaker. The universal background model [14] (UBM) for the imposter was built by collecting feature samples from a large number of speakers in the database. For the GMM, the UBM had 256 mixtures, whereas for other measures the entire set of UBM samples were used.

We evaluate each of the above scores for a test signal with respect to the reference speaker and imposter speaker models and compute the ratio between the two, which is then used for threshold comparisons. The equal error rate obtained in this way is reported in Table 1 for each of the scores. It can be seen that the proposed scoring function outperforms the other approaches in all the cases.

In Table 1, we have also reported the average time taken to evaluate the score between two sets of feature vectors (speaker/imposter). The measures *GaussLL* and KL_a take the least time. However, these measures use only the first and second order statistics for score evaluation and hence inexpensive to compute. While our score is more expensive, it still takes less time than all advanced approaches (VQ, GMM, and KL_s).

Table 1: *EER for various methods in speaker verification experiment. Time reported is the average time of one score evaluation. Time to build the imposter models for GMM and VQ is not included.*

	VQ	KL_s	KL_a	GaussLL	GMM	KRD
Time	0.7s	4.5s	0.03s	0.04s	0.4s	0.16s
EER	5.33	6.67	6.67	6.00	8.00	**4.67**

Experiment 2: Speaker Identification In speaker identification problem, the speaker is known *a priori* to be a member of a set of N speakers and a new test sample must be classified into one of N classes. In this experiment, we used the KRD measure with a 3-nearest neighbor classifier for speaker identification. We repeated the experiment with the GaussLL and VQ measures also using the 3-nearest neighbor classifier. We also built an SVM (with an rbf kernel) based speaker identification system [20] for comparison.

For each case, we use 5 samples for each speaker to do the training and test on the remaining samples. We evaluated the performance of each of the approaches for 10, 25, 50, 75, and 100-class scenarios. The classification results are shown in Table 2. It can be seen that the proposed approach performs better than the other approaches for all the cases.

Table 2: *Classification accuracy for various methods in speaker identification experiment.*

# of speakers	VQ	GaussLL	SVM	KRD
10	**96.00**	94.00	94.00	**96.00**
25	90.40	91.20	82.40	**92.00**
50	70.67	73.87	66.80	**78.40**
75	64.40	71.60	61.07	**74.40**
100	54.80	63.20	55.80	**64.80**

4. KRD based subset selection

With the improvements in learning algorithms, the complexity involved in learning has also increased along with the amount of data available. Therefore choosing the most informative subset of the data for learning has generated more interest. In subset selection, given a set of data of sample size N, we wish to extract a representative sample of size M, with $M < N$ and with the smaller dataset being statistically close to the larger one. The distance measure (Eq. 9) can be used in a greedy strategy to extract this statistically valid subset and the algorithm thus developed has several applications and we discuss two of these here.

Sparse learning algorithms use the sophisticated approaches with very few exemplar points are gaining popularity, for example, SVM[21]. On the other hand, probabilistic algorithms like Relevance Vector Machine (RVM) [22] and Gaussian Process Regression (GPR) [23] which not only provide the predictions, but also a confidence value for the prediction are also gaining popularity. Particularly, Gaussian Process Regression has a non-parametric formulation. However, GPR is hindered by its cubic computational complexity. In order to overcome this problem, sparse approaches are often used. Sparse approaches fall in three classes; 1.) a low rank approximation (chapter 8 in [23]); 2.) Learning from a subset of the original data [24, 25, 26]; 3.) learning using mixture of experts like approach [27].

Dictionary-learning is a key problem in several vision and pattern recognition tasks. For example, a dictionary of codewords learnt via vector quantization (VQ) is used in object recognition [28] and the dictionary is later used for forming histograms from objects. The histogram of the codewords are then used for training and classification of object categories. The key idea in the utilization of VQ in object recognition is to find cluster centers which are then considered as representatives of the set. It is possible to use our subset selection approach in place of VQ to learn dictionaries with the most representative set.

Subset selection algorithm Existing algorithms for subset selection can be categorized into two types, greedy and clustering-based approaches. Greedy approaches [25, 24, 26] define a cost function to minimize and add data to the subset that will minimize the cost. Clustering based approaches (eg. Vector Quantization) cluster datapoints into non-overlapping clusters and use the cluster centers as a low ranked representation. Both approaches are used for sparsification in learning and vision applications. Our objective is to use the KRD to develop a greedy algorithm to select a representative subset of a large dataset.

If the original distribution is denoted as $p(x)$, the subset selection can be formulated as forming a distribution $q(x)$ using data-points from $p(x)$ such that $p(x)$ and $q(x)$ are as close to each other as possible. In other words, we would want to add the next point in the subset to be drawn from the original set in such a way that $H_2(p\|q)$ is minimized by this addition. It is easy to see that for a direct use of the measure in Eq. (9) the subset will be clustered around the mode of the distribution. However for a subset to be actually representative of the data, it would be desirable to capture the significant outlier points as well. This can be accomplished by minimizing the distance between the subset distribution and the data distribution relative to the distance of the distribution with itself. This is done above by taking the ratio of the contribution of each training data element to the two distance measures, and the modified measure is given by,

$$H_2(p\|q) = -\log\left(1 - \frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(\frac{\hat{K}(x_{pi}, x_{qj})}{\hat{K}(x_{pi}, x_{pj})}\right)\right) \tag{12}$$

where the ratio $\frac{\hat{K}(x_{pi},x_{qj})}{\hat{K}(x_{pi},x_{pj})}$ is the relative distance contribution. The subtraction from 1 is done to to formulate subset selection as a minimization. For numerical convenience, we clamp all ratios $\frac{\hat{K}(x_{pi},x_{qj})}{\hat{K}(x_{pi},x_{pi})}$ above 1 to 1 and set $\log 0 = 0$.

Greedy algorithms for subset selection fall into two categories; they either singly add data-points from the original set to a subset till the distance between the original and new distribution is less than a pre-defined threshold, or they add a pre-defined number of data-points in batches. We use the first approach. Suppose a subset of size M needs to be extracted from a dataset of size N, the greedy algorithm would add the data points one-by-one. For each point, the distance measure is evaluated for all the points of the distribution. This step has a complexity of $\mathrm{O}(MN)$ and this is repeated for M points, thus leading to an overall complexity of $\mathrm{O}(M^2N)$. However, we have parallelized this computation efficient on a GPU using GPUML [11].

Table 3: **KRD-based subset section algorithm**

```
Given: Data $D = \{x_1, \ldots, x_N\}$
Initialize subset $I$ to be empty
FOR counter = 1 TO $M$ (input subset size)
  Define set $J$ = all elements in $D$ not in $I$
  Add an element ($el$) from $J$ to $I$
    which minimizes $H(p_D\|p_I)$ using Eq. (12)
  Remove $el$ from $J$
Output $I$
```

We exploit the facts that the distance measure in Eq. (12) is symmetric and that the influence of each sample point is additive (log is a montone function). To minimize the distance at each iteration, we consider the contribution of each data-point in the original dataset to the distance, and add the point to the subset that makes the largest relative distance contribution.

Validation - Kernel density comparison: In order to further validate our approach to subset selection, we drew 2000 samples from the 15 normal density mixtures in [29]. We estimated the underlying density using the standard kernel density estimation, utilizing the entire set of drawn samples. We then used KRD based subset selection to reduce the number of samples to 20% of the sample size, and estimated the kernel densities using this low ranked representation. The results for 6 of the 15 distributions are shown in Fig. 4. It can be seen that our low ranked estimates are similar to those obtained from the entire samples thus validating the approach further. Notice that the KDE on the entire dataset also misses some fine features because of finite sampling.

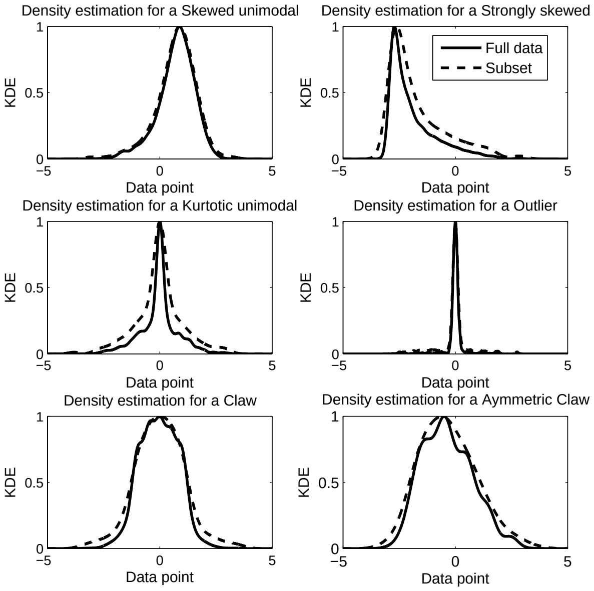


Figure 4: Density estimates of the normal density mixtures in [29] using the entire samples and our low rank subset

Experiment 1: Gaussian Process Regression: Gaussian process regression is a probabilistic kernel regression approach which uses the prior that the regression function ($f(X)$) is sampled from a Gaussian process. For regression, given a set of datapoints $D = \{X, y\}_{i=1}^{N}$, where X is the input and y is the corresponding output, the function model is assumed to be $y = f(x) + \epsilon$ where ϵ is a Gaussian noise process with zero mean and variance σ^2. Rasmussen et al. [23] use the Gaussian process prior with a zero mean function and a covariance function defined by a kernel $K(x, x')$ which is the covariance between x and x', i.e. $f(x) \sim GP(0, K(x, x'))$. They further show that with this prior, the posterior of the output y is also Gaussian with mean m and covariance V given by,

$$\begin{aligned} m &= k(x_*)^T (K + \sigma^2 I)^{-1} y \\ V &= K(x_*, x_*) - k(x_*)^T (K + \sigma^2 I)^{-1} k(x_*) \end{aligned}$$

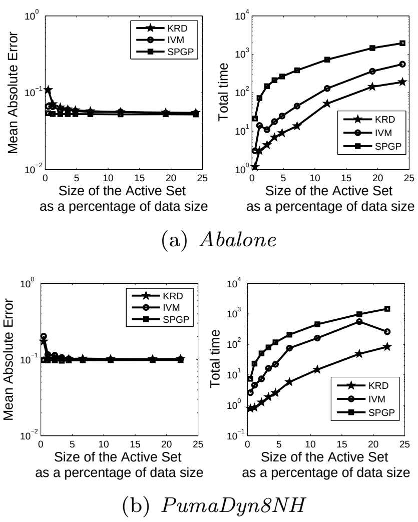
where x_* is the input at which prediction is required and $k(x*) = [K(x_1, x_*), K(x_2, x_*) \ldots, K(x_N, x_*)]$. Here m gives the prediction at $x*$ and V the variance estimate of prediction.

The core complexity in Gaussian processes involves solution of a linear system involving the kernel covariance matrix and hence is O(N^3). One approach to overcome this is to obtain a sparse representation (subset) of the original dataset which retains the information contained in the original data. For example, Online Gaussian Process (OGP) [25] uses a set of Basis Vectors (BVs) to train and predict the GP model. Similarly, the Informative Vector Machine (IVM) [24] uses a KL-like distance measure to select a representative subset by approximating the posterior. Sparse Pseudo-input Gaussian processes (SPGP) [26] performs a sampling on the training points to obtain pseudo training data which is then used for training and prediction. Each of these approaches has a computational complexity of O(MN), where N is the size of the original data and M is the size of the subset. Along the same lines, we propose the use of our subset selection algorithm to obtain a subset of the training data, by using a combined input-output space, an idea inspired by [8] where a joint feature-spatial space is used for tracking. Once the subset was selected, we trained and predicted the Gaussian Process model [23].

In order to test the proposed algorithm with Gaussian process regression, we performed regression with two standard datasets, *Abalone* and *PumaDyn8NH* [30]. We compared the performance with popular sparse data selection methods for Gaussian processes - IVM and SPGP. Fig. 5 shows that our algorithm performed much faster than the other methods for comparable error residues.

Note that the approaches with which we compared our method were tuned low-ranked approximations designed specifically for Gaussian process regression, thus our untuned subset selection performs on par with the other tuned approaches.

Pose Estimation: Motivated by the superior computational performance of the KRD-based sparse GPR, we applied our approach to learn the head pose from human face images. Sparse regression based pose estimation has been done in several papers, for example, [31] uses RVM to train images to learn poses, [32] uses an online Gaussian process algorithm to learn head pose from images. For this experiment, we used the PIE dataset [33] after annotating the image. For the purpose of this experiment, we considered only the horizontal orientations of the human face. The



(a) *Abalone*

(b) *PumaDyn8NH*

Figure 5: Comparison of the performance of the training and prediction with our approach, Informative vector machine and Sparse Pseudo-input Gaussian Process with *Abalone* and *PumaDyn8NH*

images were annotated with a score between −1 (left) to +1 (right) based on the horizontal orientation of the human face. A randomly selected class from the dataset is shown in Fig. 6 along with the score assigned to them.

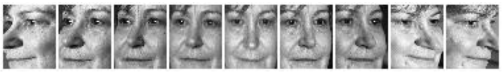


Figure 6: This is a randomly chosen class of pose images from the PIE dataset. The images were assigned scores of {-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75,1} from left-to-right

Each image was projected onto a 30 dimensional subspace using PCA and were trained to learn the scores assigned to the image. Further, we also compared the results with popular sparse learning methods Relevance Vector Machine (RVM) (from [34] and Support Vector Machine (SVM) (from [35]. The error in prediction and performance are tabulated in table 4. In all the experiments, 90% of the images were used for training and the learning method was tested on the remaining 10%. 20% of the training data were selected by our method which was then used for training the GP model. Note that, both RVM and GPR are probabilistic regression approaches and provide a variance in prediction as well, a key difference from SVM. KRD-based GPR is faster than RVM. It is slower than SVM, but the additional computational cost is to provide the variance in predictions.

Experiment 2: Visual words and object recognition We applied our subset selection algorithm to object recognition. The bag-of-features approach [36, 28] have been widely used for object categorization because of its simplicity and good performance. The basic steps in bag-of-feature based object recognition can be summarized as:

1. Features are extracted from an image by either diving it into grids or using interest point detectors.
2. The features are then represented by a set of descriptors. One of the popular descriptors are the Scale-Invariant Feature Transform (SIFT) [37].
3. The next step is to generate a codebook from the descriptors. In this step, the feature descriptors are Vector Quantized (VQ) and the centers of the clusters are defined to be the codewords of the dictionary of object categories.
4. Features from the images can now be expressed as a histogram of all codewords in the dictionary.
5. The histogram is used to train a classifier for object categorization.
6. For an unlabeled image, the histogram of codewords is extracted, and then the trained classifier is used for classification.

Table 4: Comparison of performance of our method with SVM and RVM for pose estimation. Each error entry gives the mean absolute error between the predicted face pose score and the actual score assigned to the image. Note that the prediction using RVM and GPR involved the evaluation of the variance (confidence) also, whereas the SVM computed only the predictions

Method	Mean Absolute Error in prediction	Time taken for prediction (s)
GPR	0.0261	20.7
RVM	0.0431	50.4
SVM	0.0755	9.9

We replace the VQ step above with the KRD-based subset selection approach to learn the codebook. A standard k-means based vector quantizer was used for comparisons in this experiment. We use the SIFT descriptors of the image extracted after running an interest-point detector using the toolbox from [38]. In order to provide a basis for comparison, we also use a VQ based dictionary. Once the dictionaries are obtained, the histogram of codewords are extracted from the image. We use a 5-Nearest Neighbor classifier to compare the performance of the two dictionaries. The images used for the training and testing were obtained from the *Caltech-101* dataset [39].

We randomly choose classes from the Caltech-101 dataset and extracted dictionaries using 5 images from each class with the two approaches mentioned. The dictionaries were used to obtain codeword histogram from each image. The trained histograms are then used to classify unseen test images using a 5 nearest neighbor search. We repeated the experiment for 2, 3, 4, 5 and 10 class prediction, in each case the size of the dictionary was set at 30 times the number of classes trained. Table 5 shows the overall accuracy and time taken for dictionary formation for our approach and VQ based approach. It can be seen that, with comparable accuracy, our approach is much faster than the VQ based approach, especially as the number of classes increases.

Table 5: Accuracy of classification when objects from different number of classes were trained and predicted. The size of the dictionary was set to be 30 times the number of classes of object present. Each entry here indicates the over-all percentage of correct prediction, and the time taken for dictionary formation is given within braces

#-classes	VQ-based	KRD-based
2	**77.8** (24.1s)	71.3 (**18.7**s)
3	62.3 (36.1s)	**63.8** (**26.7**s)
4	**78.4** (95s)	**78.4** (**83**s)
5	61.4 (175.3s)	**62.7** (**103.6**s)
10	47.8 (313.3)	**52.7** (**175**s)

5. Conclusions

In this paper we have explored various applications of a quadratic Rényi entropy based distance measure (kernelized Rényi distance or the KRD) obtained from a non-parametric formulation via kernel density estimation. We use GPU parallelization to accelerate the distance computation resulting in an efficiently computed non-parametric entropic distance. The KRD is adapted into similarity score for speaker recognition. We further developed a KRD-based subset selection algorithm with applications in object recognition and low ranked learning. The results in each case are promising.

References

[1] J. Principe, J. Fisher, and D. Xu, *Information theoretic learning*, S. Haykin, Ed. Wiley-Interscience, 2000.

[2] A. O.H., B. Ma, O. Michel, and J. Gorman, "Alpha divergence for classification, indexing and retrieval," University of Michigan, Tech. Rep., 2001.

[3] B. Srinivasan and R. Duraiswami, "Efficient subset selection via the kernelized Rényi distance," in *IEEE International Conference on Computer Vision*. IEEE Computer Society, September 2009, pp. 1081–1088.

[4] B. Srinivasan, R. Duraiswami, and D. Zotkin, "Kernelized Rényi distance for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[5] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization (Wiley Series in Probability and Statistics)*. Wiley-Interscience, September 1992.

[6] D. Xu, J. Principe, J. Fisher, and H. Wu, "A novel measure for independent component analysis (ICA)," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 1998, pp. 1161–1164.

[7] R. Jenssen, I. Hild, K.E., D. Erdogmus, J. Principe, and T. Eltoft, "Clustering using renyi's entropy," vol. 1, 2003, pp. 523 – 528.

[8] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2005, pp. 176–183.

[9] D. Erdogmus, I. Hild, K.E., and J. Principe, "Independent components analysis using Rényi's mutual information and legendre density estimation," vol. 4, 2001, pp. 2762 –2767.

[10] V. Morariu, B. Srinivasan, V. Raykar, R. Duraiswami, and L. Davis, "Automatic online tuning for fast Gaussian summation," in *Advances in Neural Information Processing Systems*, 2008. [Online]. Available: http://sourceforge.net/projects/figtree/

[11] B. Srinivasan, Q. Hu, and R. Duraiswami, "GPUML: Graphical processors for speeding up kernel machines," in *Workshop on High Performance Analytics - Algorithms, Implementations, and Applications*. Siam International Conference on Data Mining, 2010. [Online]. Available: http://www.umiacs.umd.edu/users/balajiv/GPUML.htm

[12] E. Gokcay and J. Principe, "Information theoretic clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 158–171, Feb 2002.

[13] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[14] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models," in *Digital Signal Processing*, 2000.

[15] F. Bimbot, I. Magrin-Chagnolleau, and L. Mathan, "Second-order statistical measures for text-independent speaker identification," *Speech Communication*, vol. 17, pp. 51–54, 1995.

[16] F. Soong, A. Rosenberg, L. Rabiner, and B. Juang, "A vector quantization approach to speaker recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 10, Apr 1985, pp. 387–390.

[17] J. Campbell, "Speaker recognition: a tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep 1997.

[18] Defense, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT)*, DARPA-ISTO, 1990.

[19] D. P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005, online web resource. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[20] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-carrasquillo, "Support vector machines for speaker and language recognition," vol. 20, 2006, pp. 210–229.

[21] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[22] M. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems*. Morgan Kaufmann, 2000.

[23] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[24] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems*, 2003, pp. 609–616.

[25] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002.

[26] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," pp. 1257–1264, 2006.

[27] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," in *In Advances in Neural Information Processing Systems 14*, 2002, pp. 881–888.

[28] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," vol. 2, 2005.

[29] J. Marron and M. Wand, "Exact mean integrated squared error," *The Annals of Statistics*, vol. 20, no. 2, pp. 712–736, 1992.

[30] L. Torgo, available at http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html.

[31] Y. M., Y. K., K. Kinoshita, S. Lao, and M. Kawade, "Sparse Bayesian regression for head pose estimation," *Proceedings of ICPR 2006*, vol. 3, pp. 507–510, 0-0 2006.

[32] A. Ranganathan and M. Yang, "Online sparse matrix Gaussian process regression and vision applications," in *Proceedings of ECCV '08*. Springer-Verlag, 2008, pp. 468–482.

[33] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE transactions on Pattern Analysis and Machine Inference*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[34] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla, "Multivariate relevance vector machines for tracking," in *European Conference on Computer Vision*. Springer-Verlag, 2006, pp. 124–138.

[35] S. Canu, S. Grandvalet, V. Guigue, and A. Rakotoma-

monjy, “SVM and kernel methods Matlab toolbox,” Perception Systmes et Information, France, 2005.

[36] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, “Discovering object categories in image collections,” in *Proceedings of ICCV*, 2005.

[37] D. Lowe, “Object recognition from local scale-invariant features,” 1999, pp. 1150–1157.

[38] A. Vedaldi and B. Fulkerson, “VLFeat: An open and portable library of computer vision algorithms,” http://www.vlfeat.org/, 2008.

[39] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” 2004, p. 178.