Efficient subset selection via the kernelized Rényi distance

Balaji Vasan Srinivasan and Ramani Duraiswami
Perceptual Interfaces and Reality Laboratory
University of Maryland, College Park, MD 20742, USA
{balajiv,ramani}@umiacs.umd.edu

Abstract

With improved sensors, the amount of data available in many vision problems has increased dramatically and allows the use of sophisticated learning algorithms to perform inference on the data. However, since these algorithms scale with data size, pruning the data is sometimes necessary. The pruning procedure must be statistically valid and a representative subset of the data must be selected without introducing selection bias. Information theoretic measures have been used for sampling the data, retaining its original information content. We propose an efficient Rényi entropy based subset selection algorithm. The algorithm is first validated and then applied to two sample applications where machine learning and data pruning are used. In the first application, Gaussian process regression is used to learn object pose. Here it is shown that the algorithm combined with the subset selection is significantly more efficient. In the second application, our subset selection approach is used to replace vector quantization in a standard object recognition algorithm, and improvements are shown.

1. Introduction

Statistical learning methods are used extensively in many fields including computer vision. With improved imaging, the amount of data available for learning has increased by many folds in the last decade. In order to efficiently learn models from the acquired data, it is necessary to prune the data in a statistically meaningful way. In other words, a sparse subset of the original dataset must be selected for learning in a way that ensures that the sparse subset does not introduce any bias in the learning and retains the information content of the original data. There are different ways of extracting the subset retaining the information content. The Support Vector Machine (SVM) [2] sparsifies the data by retaining only the data close to the inter-class boundary. The Relevance Vector Machine (RVM) [25] uses an EM-based optimization to obtain a sparse representation. Vector Quantization (VQ) divides a large data into clusters having approximately the same number of data points closest to them and uses the cluster centers as a sparse representation. Alternatively, information theoretic measures [4] like entropy and divergences have also been used to prune large datasets into a representative subset [5, 12]. We propose another information theoretic approach that can *efficiently* select a representative subset from a large dataset.

The most commonly used information theoretic measure is the *entropy* which measures the amount of information contained in a distribution. The widely used Shannon entropy of a random variable X, with probability distribution function (pdf) $p(x)$, 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 entropies called the Rényi entropy. The Rényi entropy [4] of order α is given by

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

As $\alpha \rightarrow 1$, the Rényi entropy reduces to the Shannon entropy (Eq. 1). While the entropy measures the information content of a distribution, the *divergence* measures inter-distribution distance and is also widely used. Some popular divergence measures include the Kullback-Leibler (KL) divergence (based on the Shannon entropy), the Bhattacharya distance and the Rényi (or Jensen-Rényi) divergence. The generalized Rényi divergence between two distributions with pdf $p(x)$ and $q(x)$ is given by

$$H(p\|q) = \frac{1}{1-\alpha}\log\int \frac{p(x)^\alpha}{q(x)^{(1-\alpha)}}dx. \tag{3}$$

The KL divergence is obtained as $\alpha \rightarrow 1$ in Eq. (3) and is

$$D_{KL}(p\|q) = \int p(x)\log\frac{p(x)}{q(x)}dx. \tag{4}$$

The Bhattacharya distance has $\alpha = 0.5$ in Eq. (3), and is

$$D_B = -2\log\int\sqrt{p(x)q(x)}dx. \tag{5}$$

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

The case $\alpha = 1$ is special since the entropy (and divergence) of a joint probability distribution for this value can be separated into the entropies (divergences) of the individual random variables of the joint distribution [4]. This, coupled with the tractability of the measures, have made the Shannon entropy and the KL-divergence the preferred choice for many subset selection problems e.g. [5, 12].

However, the Shannon entropy has certain deficiencies. For random data known only via samples, the Shannon measure is relatively expensive to calculate, and is often computed via histograms or order statistics, which leads to biased estimates [10, 28]. Ref. [10] shows that Shannon entropy estimation approaches do not converge to the actual sample entropy even after bias compensation and further show the need for $1 < \alpha < 4$ to achieve convergence to actual entropy. This is also illustrated in [31], where it is shown that the Bhattacharya and KL divergences computed via histograms exhibit considerable bias.

We derive a distance measure termed the **kernelized Rényi distance (KRD)** based on the Renyi entropy with $\alpha = 2$. We simplify the resulting integral by using a non-parametric kernel density estimate of the pdf, and then speed up the distance evaluation by using a fast matrix vector product. The distance measure thus obtained is used to develop a greedy algorithm to select a subset of a large dataset. Throughout this paper, the term *Rényi entropy* refers to $\alpha = 2$.

The paper is organized as follows. In section 2, we derive the proposed distance measure and introduce an efficient way to evaluate the KRD. In section 3, we discuss the required modifications to the proposed measure to use it as a divergence measure and develop a greedy subset selection algorithm based on it. We provide the results of various experiments in section 4 along with our discussions. Section 5 concludes the paper and discusses some further work.

2. Rényi entropy and our distance measure

The Rényi entropy (for $\alpha = 2$) is given by,

$$H_2(x) = -\log \int p(x)^2 dx. \tag{6}$$

Here $p(x)$ is the pdf of the random variable X. In many practical scenarios, the density is not known, and needs to 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 of the density is assumed and the parameters of the density function are estimated from the samples. 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 [9], we get

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} K(x, x_i), \tag{7}$$

where $K(x, x_i)$ is a kernel function, often Gaussian,

$$K(x_1, x_2) = \frac{1}{h\sqrt{2\pi}} \exp\left(-\frac{|x_1 - x_2|^2}{h^2}\right), \tag{8}$$

with h the bandwidth that must be selected according to the data. Using Eq. (7) in Eq. (6),

$$\begin{aligned} H_2(x) &= -\log \int \left(\frac{1}{N} \sum_{i=1}^{N} K(x, x_i)\right)^2 dx \qquad (9) \\ &= -\log \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \int K(x, x_i) K(x, x_j) dx. \end{aligned}$$

For a Gaussian kernel,

$$\int K(x, x_i) K(x, x_j) dx = \hat{K}(x_i, x_j), \tag{10}$$

where $\hat{K}$ is also Gaussian with bandwidth equalling sum of the bandwidths of the 2 Gaussian kernels [27]. Using this in Eq. (9),

$$H_2(x) = -\log \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \hat{K}(x_i, x_j)\right). \tag{11}$$

Consider two distinct distributions with densities p and q, defined by the datasets, $D_p = \{x_{p1}, x_{p2}, \ldots, x_{pN}\}$ and $D_q = \{x_{q1}, x_{q2}, \ldots, x_{qM}\}$. Then,

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

is the distance between $p(x)$ and $q(x)$. We refer to this as the ***Kernelized Rényi Distance*** **(KRD)** and is the measure we will use for subset selection. A distance measure very similar to the KRD (Eq. 12) has been used in [28] for tracking. However, the measure proposed in that paper is heuristic. Here, we have derived this from information theory. It is also interesting to note that, the Cauchy-Schwartz distance [9] given by,

$$D(p\|q) = -\log \frac{\int p(x) q(x) dx}{\int (p(x))^2 dx \int (q(x))^2 dx}, \tag{13}$$

can also be simplified into a form similar to the one in Eq. (12). The advantages of the KRD are:
(1) because the KRD uses a non-parametric on-the-fly density estimation, it does not require any parametric approximation for distance evaluation;

(2) the KRD is symmetric unlike the popular KL-divergence measure and like the Bhattacharya distance;
(3) because it starts with Rényi entropy of $\alpha = 2$ it is more optimal than measures with $\alpha < 2$ for sample based computations [10, 28].

Efficient evaluation of the KRD: The evaluation of KRD between two distributions, represented by O(N) datapoints, would require O(N^2) operations. It should be noted that the core computation in the KRD evaluation with Eq. 12 is summation of the Gaussian kernel. A number of algorithms have been proposed for fast computation of the sums of Gaussians. A recent paper [17] combines the best features of two of these methods, the improved fast Gauss transform [20] and tree-based methods [13]. We used this method (available as an open source software) to speed up our KRD evaluation to linear time.

3. Subset selection algorithm

Existing algorithms for subset selection can be categorized into two types, greedy and clustering-based approaches. Greedy approaches [5, 12, 23] define a cost function to minimize and adds data to the subset that will minimize the cost. Clustering based approaches (eg. Vector Quantization) cluster datapoints in non-overlapping clusters and use the cluster centers as the low ranked representation. Both these approaches are well known 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. (12) 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. The distance measure in Eq. (12) is therefore modified as,

$$H_2(p\|q) = -\log\left(1 - \frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(\frac{\hat{K}(x_{pi}, x_{qj})}{\hat{K}(x_{pi}, x_{pj})}\right)\right) \tag{14}$$

As mentioned before, the requirement on the subset selection is that the pdf defined from the subset should be as close as possible to the original distribution. Hence, in our KRD based subset selection, we minimize 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. 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 $\log 0 = 0$.

Algorithm for subset selection
Given: Data $D = x_{i=1}^{N}$
Initialize subset I to be empty
Loop from 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. (14)
Remove el from J
End Loop
Output I

Table 1. The greedy algorithm for subset selection using the distance measure

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 incrementally. In this paper we will use the latter 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 from the distribution. Because of the use of fast Gauss summations, this step has a complexity of O(N) and this is repeated for M points, thus leading to an overall complexity of O(MN). Without the fast-matrix-vector product of [17], this would have costed O(MN) for each point selection and O(M^2N) for the entire algorithm. The algorithm is shown in Table 1.

Here, we exploit the facts that the distance measure in Eq. (14) 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.

3.1. Validation

Gaussian mixture recovery: In order to confirm that the proposed algorithm is able to extract a representative subset, data were sampled from a mixture of Gaussians and a subset of this data was obtained using the algorithm. Then the original data and the subset were clustered to determine the means and the variances within each cluster (the number of clusters was chosen as the number of mixtures). In order to further substantiate our algorithm, we selected a random subset of the data and performed clustering. This was repeated for different number of mixtures and in each case 20% of the the data was selected. For a random sampling,

# of mixtures	Mean	Variance
1	0.0087(0.0672)	0.0305(0.0254)
2	0.0314(0.0786)	0.0303(0.0392)
3	0.0769(0.0445)	0.0194(0.0137)
4	0.109(0.2229)	0.0259(0.012)
5	0.0805(0.3834)	0.0493(0.0465)

Table 2. Subsets were selected from data generated from mixtures of Gaussians using our approach and random sampling. The mean error in the mean and variance of each cluster for different mixtures is shown here. The value inside the braces gives the corresponding error for a random sampling.

it can be shown using Chernoff bounds that the sampling may yield a representative subset for large enough subset size, but with some uncertainty. The objective of this validation is to show that our approach is consistent against the uncertain random sampling. The results are shown in Table. 2. As can be seen, our algorithm performs consistently against the random sampling which did not give low error consistently.

Kernel density comparison: In order to further validate our approach to subset selection, we drew 2000 samples from the 15 normal density mixtures in [16]. We estimated the underlying density using the standard kernel density estimation, utilizing the entire set of drawn samples. We then used our 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. 1. It can be seen that our low ranked estimates are similar to those obtained from the entire samples thus validating our approach further. Notice that the KDE on the entire dataset also misses some fine features because of the sample size.

3.2. Applications

There are numerous applications where the KRD (Eqs. (12) and (14)) can be used. For example, it can be used as a similarity measure in tracking as in [28] or for image search as in [11]. However, we limit our experiments and discussion to the subset and discuss two applications.

As mentioned earlier, the complexity of using learning algorithms increases along with the amount of data available. Hence sparse learning algorithms, which use sophisticated approaches with very few exemplar points, are popular, e.g. Support Vector Machines [2] (SVM). Probabilistic algorithms like Relevance Vector Machines [25] (RVM) and Gaussian Process Regression [19] (GPR) which not only provide the *predictions*, but also a *confidence value* for the prediction are also gaining popularity and is the first application considered. Particularly, since Gaussian Process Regression has a non-parametric formulation, it is considered to be a robust learning approach. However, the application of GPR is hindered by its cubic computational complexity. In order to overcome this problem, sparse approaches and fast algorithms are used. Sparse approaches fall in three classes;
(1) learning from a subset of the original data like in [12, 5, 23];
(2) a low rank approximation (chapter 8 in [19]);
(3) using mixture of experts. Our subset selection approach can be directly used in the first class - sparse GPR learning.

Vector quantization is used in object recognition (e.g. [8]) for learning a dictionary of codewords, which can later be used for forming histograms from objects. The histogram of the codewords are then used for training and classification of object categories. Another application of vector quantization is in speaker identification (text dependent and independent) [29]. The key idea in the utilization of VQ in these applications 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 as is shown in the section 4 for object recognition.

4. Experiments

In the first experiment, we tested the performance of the proposed algorithm with GPR and compared it with existing sparse GPR approaches. We then apply the regression approach to estimate the head pose in human faces and compare the performance with other popular sparse approaches namely RVM and SVM based regression. In our second experiment, we used our subset selection approach with the bag-of-features method to perform object class recognition. We compared our approach using the Vector Quantization based approach.

4.1. 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, it is assumed that a set of datapoints $D = \{X, y\}_{i=1}^{N}$, where X is the input and y is the corresponding output such that $y = f(X) + \epsilon$. ϵ is the observation noise, $\sim N(0, \sigma^2)$. For GPR, the prior is that the samples are drawn from a Gaussian process with zero mean and a covariance defined by a kernel function $K(x, x')$ (which is the covariance between x and x'), i.e. $f(x) \sim GP(0, K(x, x'))$ [19]. It is shown in [19] that with this Gaussian process prior, the posterior of the output y is also Gaussian with mean and covariance given by m and V as below,

$$m = k(x_*)^T (K + \sigma^2 I)^{-1} y, \quad (15)$$

$$V = K(x_*, x_*) - k(x_*)^T (K + \sigma^2 I)^{-1} k(x_*) \quad (16)$$

where x_* is the input at which prediction is required, K is the covariance matrix using the kernel function 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 gives the variance of prediction. The core complexity in Gaussian processes involves solving a linear system involving the kernel covari-

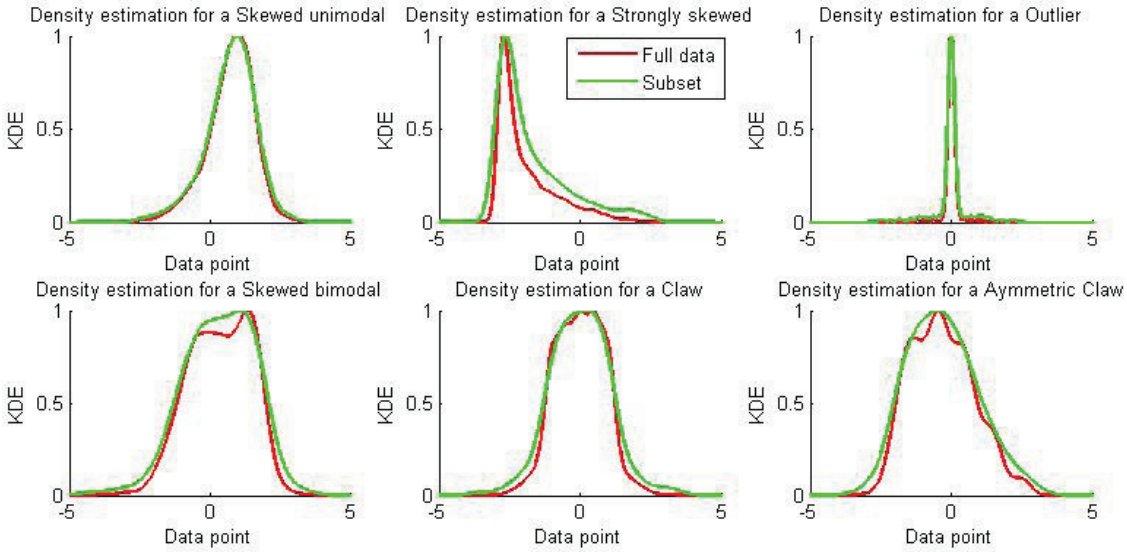
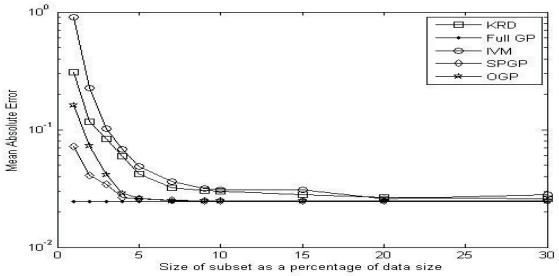
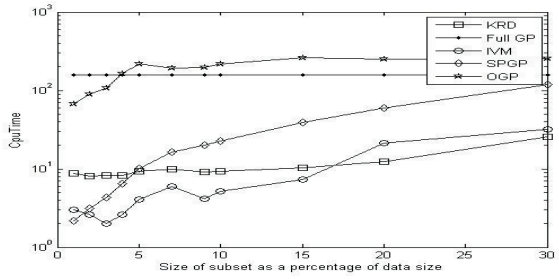


Figure 1. Density estimates of the normal density mixtures in [16] using the entire samples and our low rank subset



(a) Mean Absolute Error

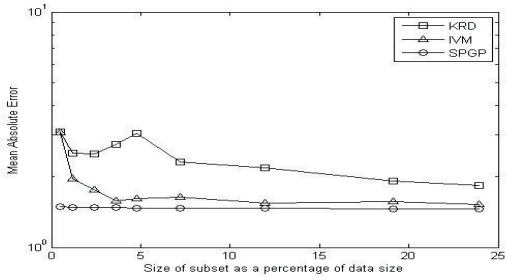


(b) CPU Time

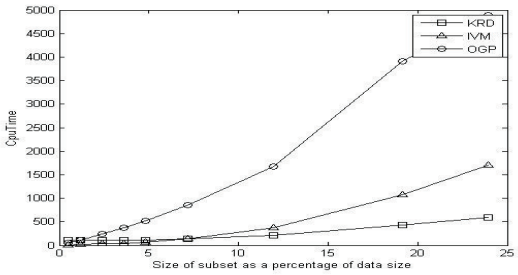
Figure 2. Result with the toy data in [12]; 2(a): The mean absolute error in the prediction of the output variable by several sparse Gaussian process regression approach along the full Gaussian process regression on the toy data. 2(b): The time taken by several sparse Gaussian process regression approach along the full Gaussian process regression on the toy data

ance 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) [5] uses a set of Basis Vectors (BVs) to train and predict the GP model. Similarly, the Informative Vector Machine (IVM) [12] uses a KL-like distance measure to select a representative subset by approximating the posterior. Sparse Pseudo-input Gaussian processes (SPGP) [23] 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 [6] where a joint feature-spatial space is used for tracking. Once the subset was selected, we trained and predicted the Gaussian Process model [19].

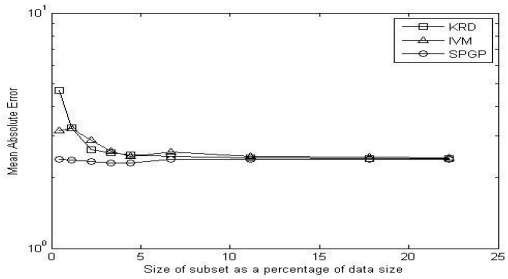
In order to test the proposed algorithm with Gaussian process regression, we first compared the performance on a toy dataset in [12] against IVM [12], OGP [5] and SPGP [23] along with the Gaussian process regression on the full dataset (Full GP). We used the Gaussian kernel for the covariance function in all the experiments with GPR. In each of these methods, the size of the subset was set and the mean absolute error between prediction and actual output were measured along with the time taken by each of these method to make the predictions. The results are shown in Fig. 2. It can be seen that our method performs comparably with all of the sparse Gaussian process methods. Further, it is important to note that the error in prediction using sparse approaches converges to that of full GPR with the use of proper subset size. It was also observed that OGP does not perform any better than full GP in training because it performs an online update of the kernel matrix, however the prediction is sped up because of the use of basis vectors. Further, we also tried to apply the Gaussian Process Regression with subset selection approach with two standard datasets, *Abalone* and *PumaDyn8NH* [1]. We compared the performance with other sparse data selection methods - IVM and SPGP. Fig. 3 shows that although all the three algorithms had the same asymptotic complexity (O(MN)), our algorithm performed much faster than the other methods when applied to large datasets thus indicating the con-



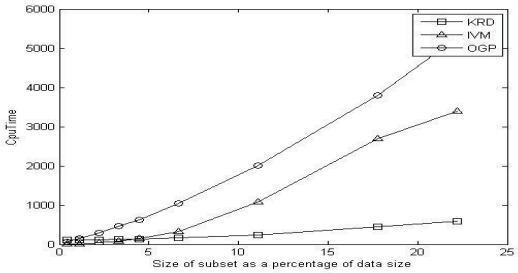
(a) Mean Absolute Error for *Abalone*



(b) CPU Time for *Abalone*



(c) Mean Absolute Error for *PumaDyn8NH*



(d) CPU Time for *PumaDyn8NH*

Figure 3. Comparison of the performance of the training and prediction with our approach, Informative vector machine and Sparse Pseudo-input Gaussian Process with standard datasets; 3(a): Mean absolute error for *Abalone* (4177 datapoints, each with 7 dimensions) dataset; 3(b): Time taken for training and prediction with *Abalone* dataset; 3(c): Mean absolute error for *PumaDyn8NH* (8192 datapoints, each with 7 dimensions) dataset; 3(d): Time taken for training and prediction with *PumaDyn8NH* dataset

stants in complexity is very less in our method.

It should be noted here that the error shown were absolute and not normalized. In the toy-data, the errors in all the methods were of the order of $0.025 - 0.03$ beyond 10% of the data. Also, the output variable in Abalone is its age and the errors obtained were of the order $1.5 - 2$. At these scales, our performance can be asserted to be comparable with other approaches. Further, 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 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, [15] uses RVM to train images to learn poses, [18] uses an online Gaussian process algorithm to learn head pose from images. For this experiment, we used the PIE dataset [21] after annotating the image. For the purpose of this experiment, we considered only the horizontal orientations of the human face. The 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. 4 along with the score assigned to them.

Each image was projected onto a 30 dimensional subspace using PCA and were trained to learn the scores as-

Figure 4. 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

signed to the image. Further, we also compared the results with popular sparse learning methods RVM (from [24] and SVM (from [3]. The error in prediction and performance are tabulated in table 3. 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. It can be seen that our method with Gaussian process makes it comparable or better than the SVM used in [3] both in time and error while providing confidence information as well.

To summarize this experiment, we first used KRD based subset selection approach with Gaussian process regression and compared it with other sparse GP approaches and showed comparable error performance on a toy dataset. We then extended the experiment with standard regression datasets and compared our approach with other popular subset selection approaches in GP and showed that our algorithm performs faster than other approaches for larger data size. Finally we applied our approach to pose estimation and comparing with popular sparse approaches, showed that

Method	Mean Absolute Error in prediction	Time taken for prediction (seconds)
GPR	0.0421	6.0469
RVM	0.0431	37.8281
SVM	0.0755	4.5781

Table 3. 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

our approach has comparable performance with the popular methods as well.

4.2. Visual words and object recognition

We then applied our subset selection algorithm to object recognition. The bag-of-features approach [22, 8] 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) [14].
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.

We propose to replace the VQ step above with the KRD-based subset selection approach to get a representative set of the collection of descriptors. We show that by this approach, for comparable accuracy, there is a marked improvement in the time taken for dictionary formation. We used a standard k-means based vector quantizer for this experiment.

We use the SIFT descriptors of the image extracted after running an interest-point detector using the toolbox from [26]. 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 [7].

	VQ-based	Our method based
2-class	77.8 (27.3s)	71.3 (29.7s)
3-class	62.3 (280.3s)	63.8 (75.8s)
4-class	78.4 (310.1s)	78.4 (249.8s)
5-class	61.4 (3651.5s)	62.7 (729.7s)
6-class	63.4 (**3765.9s**)	59.3 (**1307.2s**)
10-class	47.8 ($\sim$ **5hrs**)	52.7 ($\sim$ **1hr**)

Table 4. 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

In the first part of this experiment, we randomly choose 5 classes from the dataset and extracted dictionaries using 5 images from each class with the two approaches mentioned. The size of the dictionary was set at 300 in all cases. 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 achieved an overall accuracy of 56.7% with our approach compared to the 49.5% accuracy from the Vector Quantization based approach. 5-class confusion matrix for VQ-based approach is given by,

$$\begin{pmatrix} \mathbf{0.3333} & 0.2083 & 0.0213 & 0 & 0.0105 \\ 0.0500 & \mathbf{0.4375} & 0 & 0 & 0 \\ 0.3333 & 0.1667 & \mathbf{0.9362} & 0.7000 & 0.5079 \\ 0.0500 & 0.0208 & 0 & \mathbf{0.3000} & 0 \\ 0.2333 & 0.1667 & 0.0426 & 0 & \mathbf{0.4817} \end{pmatrix},$$

and for our KRD-based approach is given by

$$\begin{pmatrix} \mathbf{0.3833} & 0.2083 & 0 & 0 & 0.0209 \\ 0.0833 & \mathbf{0.6250} & 0 & 0 & 0 \\ 0.1667 & 0.0417 & \mathbf{0.8723} & 0.4667 & 0.4293 \\ 0.0667 & 0.0208 & 0 & \mathbf{0.5000} & 0.0052 \\ 0.3000 & 0.1042 & 0.1277 & 0.0333 & \mathbf{0.5445} \end{pmatrix}$$

In the confusion matrix, along each columns is the predicted labels and each row shows the true label. Except for the 3^{rd} class, the classification accuracy of the other classes is better by using our algorithm. Also the overall accuracy was also better than the VQ based approach. We further repeated the experiment for 2, 3, 4, 5, 6 and 10 class prediction, in each case the size of the dictionary was set at 30 times the number of classes trained. Table 4 shows the over-all 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. We have thus

shown that the dictionary based on our method has comparable performance with the VQ based approach, but takes lesser time for dictionary formation.

5. Conclusion

We developed a new information-theoretic distance measure and used it to develop a subset selection algorithm. The form of the distance function allowed the use of an O(N) fast Gaussian summation algorithm and allowed a speed up of the distance evaluation to linear time. The subset selection algorithm was successfully applied to both synthetic and real problems (Gaussian process regression, and to replace vector quantization). Our approach, while being much more efficient, performed comparably or better than approaches previously used.

Further work: Inspite of the good performance of our approach, one open question that still remains is the size of the subset that could be chosen. It would be interesting to explore the possibility of extending the proposed method to automatically tune the size of the subset based on a defined information theoretic error criteria.

References

[1] http://www.liaad.up.pt/ ltorgo/regression/datasets.html. 5

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998. 1, 4

[3] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. SVM and kernel methods Matlab toolbox. Perception Systmes et Information, France, 2005. 6

[4] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991. 1, 2

[5] L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Comput.*, 14(3):641–668, 2002. 1, 2, 3, 4, 5

[6] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. *Proceedings of CVPR 2003*, 1:I–781–I–788 vol.1, June 2003. 5

[7] 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. page 178, 2004. 7

[8] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. volume 2, 2005. 4, 7

[9] E. Gokcay and J. Principe. Information theoretic clustering. *IEEE Transaction on PAMI*, 24(2):158–171, Feb 2002. 2

[10] A. Hegde, T. Lan, and D. Erdogmus. Order statistics based estimator for renyi entropy. *Machine Learning for Signal Processing, 2005 IEEE Workshop on*, pages 335–339, Sept. 2005. 2, 3

[11] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. *Proceedings of CVPR 2008*, pages 1–8, June 2008. 4

[12] N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003. 1, 2, 3, 4, 5

[13] D. Lee, A. Gray, and A. Moore. Dual-tree fast gauss transforms. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 747–754. MIT Press, Cambridge, MA, 2006. 3

[14] D. G. Lowe. Object recognition from local scale-invariant features. pages 1150–1157, 1999. 7

[15] Y. Ma, Y. Konishi, K. Kinoshita, S. Lao, and M. Kawade. Sparse Bayesian regression for head pose estimation. *Proceedings of ICPR 2006*, 3:507–510, 0-0 2006. 6

[16] J. Marron and M. Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20(2):712–736, 1992. 4, 5

[17] V. I. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. Davis. Automatic online tuning for fast Gaussian summation. In *NIPS*. MIT Press, 2008 http://sourceforge.net/projects/figtree/. 3

[18] A. Ranganathan and M.-H. Yang. Online sparse matrix Gaussian process regression and vision applications. In *Proceedings of ECCV '08*, pages 468–482, Berlin, Heidelberg, 2008. Springer-Verlag. 6

[19] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005. 4, 5

[20] V. C. Raykar and R. Duraiswami. The improved fast Gauss transform with applications to machine learning. In L. Bottou, O. Chapelle, D. Decoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 175–201. MIT Press, 2007. 3

[21] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression database. *IEEE Transactions on PAMI*, 25(12):1615–1618, Dec. 2003. 6

[22] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *Proceedings of ICCV*, 2005. 7

[23] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs, 2006. 3, 4, 5

[24] A. Thayananthan, R. Navaratnam, B. Stenger, P. H. S. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. In *In ECCV*, pages 124–138. Springer-Verlag, 2006. 6

[25] M. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems, San Mateo, CA*. Morgan Kaufmann, 2000. 1, 4

[26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 7

[27] D. Xu, J. C. Principe, J. F. Iii, and H. chun Wu. A novel measure for independent component analysis. pages 1161–1164, 1998. 2

[28] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. *Proceedings of CVPR 2005*, 1:176–183 vol. 1, June 2005. 2, 3, 4

[29] G. Zhou and W. Mikhael. Speaker identification based on discriminative vector quantization. *Micro-NanoMechatronics and Human Science, 2005 IEEE International Symposium on*, 2:617–620 Vol. 2, Dec. 2003. 4