SCALABLE SPARSE APPROXIMATION OF A SAMPLE MEAN

Efrén Cruz Cortés, Clayton Scott

University of Michigan
Electrical Engineering and Computer Science
Ann Arbor, MI

ABSTRACT

We examine the problem of approximating the mean of a set of vectors as a sparse linear combination of those vectors. This problem is motivated by a common methodology in machine learning where a probability distribution is represented as the sample mean of kernel functions. In applications where this kernel mean function is evaluated repeatedly, having a sparse approximation is essential for scalability. However, existing sparse approximation algorithms such as matching and basis pursuit scale quadratically in the sample size, and are therefore not well suited to this problem for large sample sizes. We introduce an approximation bound involving a novel incoherence measure, and propose bound minimization as a sparse approximation strategy. In the context of sparsely approximating a kernel mean function, the bound is efficiently minimized by solving an appropriate instance of the k-center problem, and the resulting algorithm has linear complexity in the sample size.

Index Terms— sparse approximation, kernel methods, k-center problem, incoherence

1. INTRODUCTION

We consider the problem of sparsely approximating the mean of a set of vectors. Although we are motivated by a problem in machine learning, we begin by stating the problem more generally.

Consider a set of n vectors $\{z_1, ..., z_n\} \subseteq V$, where $(V, \langle\cdot,\cdot\rangle)$ is an inner product space of arbitrary dimension, and define $[n] := \{1, \ldots, n\}$. Let $\|\cdot\|$ denote the induced norm on V and, for $\alpha \in \mathbb{R}^n$, define $\|\alpha\|_0 := |\{i \,|\, \alpha_i \neq 0\}|$. Given an integer k, $1 \leq k < n$, we would like to find the best approximation to the sample mean of $\{z_1, ..., z_n\}$ as a linear combination of k sample points. That is, we want to solve:

$$\min_{\alpha} \|\bar{z} - z_\alpha\| \qquad \text{subject to } \|\alpha\|_0 = k \tag{1}$$

where $\bar{z} := (1/n)\sum_{i\in[n]} z_i$ and $z_\alpha := \sum_{i\in[n]} \alpha_i z_i$.

Problem (1) is in the form of the standard sparse approximation problem [1]. Existing sparse approximation algorithms, such as matching pursuit and basis pursuit, require the computation of the matrix $K := (\langle z_i, z_j\rangle)_{i,j\in[n]}$, which causes these algorithms to scale quadratically in n. We are interested in settings where n is potentially large, and in the context of our motivating task, we propose an approximate solution to (1) having linear complexity in n. Instances in which existing algorithms can be implemented to scale linearly in n develop a quadratic dependency on the dimension of V, which in our case is arbitrary, possibly infinite, and are therefore prohibitive.

1.1. Motivation: Sparse Kernel Representations of Probability Distributions

This work is motivated by kernel methods in machine learning. In particular, there are two well-established paradigms for representing a probability distribution as the sample mean of kernel functions. In both paradigms, $z_i = k_\sigma(\cdot, x_i)$, where $x_i \in \mathbb{R}^d$, and k_σ is a kernel function with bandwidth parameter σ. Thus, V is a space of functions.

In kernel density estimation, a random sample $x_1, \ldots, x_n$ is drawn from a distribution with density f. The kernel density estimate (KDE) of f is

$$\widehat{f} = \frac{1}{n}\sum_{i\in[n]} k_\sigma(\cdot, x_i) = \bar{z}\,. \tag{2}$$

In this context, k_σ is usually assumed to satisfy $k_\sigma(x, x') \geq 0$ and $\int k_\sigma(x, x')dx = 1$. For most commonly used kernels we can take $V = L^2(\mathbb{R}^d)$. If k_σ happens to be a reproducing (equivalently, positive semi-definite (PSD)) kernel, we may take V to be the associated reproducing kernel Hilbert space (RKHS), in which case $\langle z_i, z_j\rangle = k_\sigma(x_i, x_j)$.

The second paradigm is the kernel mean embedding of a distribution P. Here k_σ is required to be a PSD kernel, and the embedding maps P to k_σ's corresponding RKHS by the transformation $\Psi(P) := \int k_\sigma(\cdot, x)dP(x)$. In practice, Ψ is estimated from a random sample $x_1, \ldots, x_n \sim P$ by

$$\widehat{\Psi}(P) := \frac{1}{n}\sum_{i\in[n]} k(\cdot, x_i) = \bar{z}\,. \tag{3}$$

ECC and CS were supported in part by NSF Grants 0953135, 1047871, and 1217880.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

The above kernel representations are used to solve a variety of machine learning and signal processing problems, such as regression [2, 3], image registration [4], canonical correlation analysis [5], class proportion estimation [6], clustering [7,8], and transfer learning [9]. For these applications a sparse representation takes the form $z_\alpha = \sum_{i|\alpha_i \neq 0} \alpha_i k_\sigma(\cdot, x_i)$. For $k \ll n$, such a representation would be much more efficient with respect to both storage and evaluation. We also assume that $\langle z, z' \rangle = \langle k_\sigma(\cdot, x), k_\sigma(\cdot, x') \rangle$ can be computed with linear complexity in d, which is the case for many common kernels. We will demonstrate an approach for solving (1) with $O(nkd)$ time complexity. Our approach also suggests online and divide-and-conquer implementations for further scalability.

1.2. Related Work

Efficient representation and evaluation of a sum of kernel functions has been examined from a variety of perspectives. Fast multipole methods approximate the sum of kernel functions using truncated Taylor series expansions [10]. However, these methods are only well motivated in low dimensions, since the error of the truncated Taylor series expansions grows exponentially in the dimension. Techniques for sparse kernel density estimation have been proposed, but these have quadratic complexity in the sample size [11, 12] and also rely on grid-based approximations of the empirical distribution function, which scale poorly with dimension. Methods for collapsing large mixture models to simpler ones have also been investigated, but once again, these approaches scale quadratically in the sample size [13–17]. Finally, [18] examined approximating a mean of kernel functions from the perspective of sparse approximation. Comparisons to this work are given below.

2. SUBSET SELECTION BY INCOHERENCE MAXIMIZATION

To begin, we let $\mathcal{I} \subseteq [n]$ denote an index set and reformulate problem (1) as

$$\min_{\substack{\mathcal{I} \subseteq [n] \\ |\mathcal{I}| = k}} \min_{(\alpha_i)_{i \in \mathcal{I}}} \|\bar{z} - \sum_{i \in \mathcal{I}} \alpha_i z_i\|^2 . \tag{4}$$

Now we can find the optimal α and eliminate it from (4). Given k and for fixed $\mathcal{I}$, let $\alpha_\mathcal{I} \in \mathbb{R}^k$ denote the solution to the inner optimization problem. Notice that $\alpha_\mathcal{I}$ is the solution to an unconstrained quadratic optimization problem, and can be shown to satisfy $\alpha_\mathcal{I} = K_\mathcal{I}^{-1} \kappa_\mathcal{I}$, where $K_\mathcal{I} = (\langle z_i, z_j \rangle)_{i,j \in \mathcal{I}}$ and $\kappa_\mathcal{I}$ is the vector with entries $(1/n) \sum_{j \in [n]} \langle z_l, z_j \rangle$, $l \in \mathcal{I}$. Now write $\alpha_\mathcal{I} = (\alpha_{\mathcal{I},i})_{i \in \mathcal{I}}$ and set $z_\mathcal{I} = \sum_{i \in \mathcal{I}} \alpha_{\mathcal{I},i} z_i$. Then (1) reduces to

$$\min_{\substack{\mathcal{I} \subseteq [n] \\ |\mathcal{I}| = k}} \|\bar{z} - z_\mathcal{I}\| . \tag{5}$$

Our strategy to solve (5) involves finding an upper bound on $\|\bar{z} - z_\mathcal{I}\|$ that depends on $\mathcal{I}$, and then finding the set $\mathcal{I}$ that minimizes the bound. To do this, we define $\nu_\mathcal{I} := \min_{j \notin \mathcal{I}} \max_{i \in \mathcal{I}} \langle z_i, z_j \rangle$, which is a measure of the "incoherence" of $\{z_i \,|\, i \in \mathcal{I}\}$, and establish the following:

Theorem 1. *Assume $\langle z_i, z_i \rangle = C \;\forall\, i \in [n]$, for some $C > 0$. Then, for every $\mathcal{I} \subseteq [n]$,*

$$\|\bar{z} - z_\mathcal{I}\| \leq \left(1 - \frac{|\mathcal{I}|}{n}\right) \sqrt{\frac{1}{C}(C^2 - \nu_\mathcal{I}^2)} . \tag{6}$$

The proof is placed in the last section. Given a fixed $|\mathcal{I}|$, from this result we see that choosing $\mathcal{I}$ so as to minimize the bound is the same as choosing $\mathcal{I}$ so as to maximize $\nu_\mathcal{I}$.

Noumir et al. [18] establish a bound of the same form based on the coherence parameter $\mu_\mathcal{I} = \max_{\substack{i,j \in \mathcal{I} \\ i \neq j}} |\langle z_i, z_j \rangle|$ instead of $\nu_\mathcal{I}$, and they advocate selecting $\mathcal{I}$ by minimizing $\mu_\mathcal{I}$. However, their bound only holds for the set $\mathcal{I}$ which minimizes $\mu_\mathcal{I}$, which does not suggest bound minimization as a viable strategy. Furthermore, the bound actually increases as $\mu_\mathcal{I}$ decreases. Additionally, their algorithm evidently requires the computation of the complete gram matrix of the data, which is an $O(n^2)$ procedure. In the next section we show that, in the context of approximating means of radial kernels, maximizing $\nu_\mathcal{I}$ reduces to the well known k-center problem.

3. THE K-CENTER PROBLEM

We now develop an algorithm for maximizing $\nu_\mathcal{I}$, subject to $|\mathcal{I}| = k$, in the context of our motivating application. Thus, let $x_1, \ldots, x_n \in \mathbb{R}^d$ and let $\bar{z} = (1/n) \sum_{i \in [n]} k_\sigma(\cdot, x_i)$. We assume that k_σ is such that $\langle z_i, z_j \rangle = g_\sigma(\|x_i - x_j\|)$ for some $g_\sigma : \mathbb{R} \longrightarrow \mathbb{R}$. For example, the Gaussian kernel has $g_\sigma(t) = (2\pi\sigma^2)^{-d/2} e^{-t^2/2\sigma^2}$ when V is the RKHS associated with k_σ. Note that $\langle z_i, z_i \rangle = C = g_\sigma(0)$ for each i. We also assume that $g_\sigma(t)$ is strictly decreasing for $t \geq 0$, which is the case for most kernels of interest.

Let $\mathcal{I}^*$ be the set that minimizes the upper bound of Theorem 1. By the assumed monotonicity of g_σ,

$$\mathcal{I}^* = \arg \min_{\substack{\mathcal{I} \subseteq [n] \\ |\mathcal{I}| = k}} \max_{j \notin \mathcal{I}} \min_{i \in \mathcal{I}} \|x_i - x_j\|. \tag{7}$$

Let $X_\mathcal{I} = \{x_i \,|\, i \in \mathcal{I}\}$ and $Y_\mathcal{I} = \{x_j \mid j \notin \mathcal{I}\}$, and for every $x_j \in Y_\mathcal{I}$, define its distance to $X_\mathcal{I}$ as $d(x_j, X_\mathcal{I}) = \min_{i \in \mathcal{I}} \|x_i - x_j\|$. Define also $W(X_\mathcal{I}) = \max_{x_j \in Y_\mathcal{I}} d(x_j, X_\mathcal{I})$. Our goal is therefore to find the set $\mathcal{I}$ of size k for which $W(X_\mathcal{I})$ is minimized. This is known as the k-center problem.

The k-center problem is known to be NP-complete [19], and we will use a greedy 2-approximation algorithm to solve it [20]. By 2-approximation we mean that $W(X_{\mathcal{I}_k}) \leq 2W(X_{\mathcal{I}^*})$, where $\mathcal{I}_k$ is the set chosen by the algorithm. This performace guarantee is the best possible in the sense that there is no ρ-approximation algorithm for any $\rho < 2$ [21]. The algorithm is described in Algorithm 1 [20].

```
**Input**: $x_1, \ldots, x_n, k$
Initialization:
$X \longleftarrow \varnothing$;
$Y \longleftarrow \{x_1, \ldots, x_n\}$;
Choose randomly a first index $u \in [n]$;
$X \longleftarrow X \cup \{x_u\}$;
$Y \longleftarrow Y \backslash \{x_u\}$;
while $|X| < k$ do
    Choose the element $y \in Y$ for which $d(y, X)$ is
    maximized;
    $X \longleftarrow X \cup \{y\}$;
    $Y \longleftarrow Y \backslash \{y\}$;
end
Output: $\mathcal{I}_k = \{i \in [n] \,|\, x_i \in X\}$
```

Algorithm 1: k-center algorithm

From the algorithm we see that at each iteration, we only compute n new distances, those from the set Y to the most recent element of X. Since the dimension is d, each iteration takes nd steps. Upon termination, we will have added k elements to X, so the algorithm's time complexity is $O(nkd)$.

4. SIMULATION

We now evaluate the performance of the algorithm for different values of k. We compare our algorithm to a baseline approach that selects $\mathcal{I}$ uniformly at random.

4.1. Setup

To evaluate our proposed method we have used fifteen data sets, listed in Table 1. Iris is available at the UCI machine learning repository. The next 10 sets are drawn from http://www.fml.tuebingen.mpg.de/Members/ (accessed 2010). C500 is a data set of 500 points randomly sampled from a distribution uniform over a circle of radius five plus bivariate Gaussian noise with .1 variance, C2000 is a data set of 2000 points drawn from the same distribution. G3 are 2000 points drawn from the 3-dimensional standard Gaussian, G5 comes from five dimensions.

We have chosen $k = 100$, and computed $\{\alpha_{\mathcal{I}_j}\}_{1 \leq j \leq k}$ for both the k-center algorithm and the random baseline algorithm in order to evaluate their performances. Since for the k-center method the first element of $\mathcal{I}_k$ is chosen randomly, different runs will yield different results. Therefore, we have run the experiments ten times and averaged over the results.

For the k-center method, the total time per run is $O(nkd + k^3)$. To see this, we used the fact that if we add one new element (say, element i) to the set $\mathcal{I}_j$, forming $\mathcal{I}_{j+1} = \mathcal{I}_j \cup \{i\}$ with $|\mathcal{I}_{j+1}| = j+1$, there is a way to update $\alpha_{\mathcal{I}_j}$ in $O(j^2 + jd)$ steps, see [18]. Therefore, having a target set $\mathcal{I}_k$ of size k, we can compute $\{\alpha_{\mathcal{I}_j}\}_{1 \leq j \leq k}$ in $O(k^3 + k^2 d)$ time. Since we can compute $\{\mathcal{I}_j\}_{1 \leq j \leq k}$ in $O(nkd)$ time, the whole process is $O(nkd + k^3)$, which remains $O(nkd)$ as long as k^2 is $O(nd)$.

Letting $\{x_1, \ldots, x_n\}$ indicate the data, we have chosen k_σ the Gaussian kernel and V its RKHS, so that $\langle z_i, z_j \rangle = k_\sigma(x_i, x_j)$.

For each algorithm, we measure its performance by computing $E^2_{\mathcal{I}} := \|\bar{z} - z_{\mathcal{I}}\|^2 - \|\bar{z}\|^2 = \|z_{\mathcal{I}}\|^2 - 2\,\langle \bar{z}, z_{\mathcal{I}} \rangle$ (we don't compute $\|\bar{z}\|^2$, since it is the same for every $|\mathcal{I}|$, and it takes $O(n^2)$ time), and also by computing the KL distance $D(q \,\|\, p)$, where $p = \sum_{i \in \mathcal{I}} \alpha_i z_i$ and $q = (1/n) \sum_{l \in [n]} z_l$. Note that in this case $z_i(x) = k_\sigma(x, x_i)$.

To compute the KL divergence we made the following approximation: $D(q \,\|\, p) = \mathbb{E}_q(\log \frac{q}{p}) \approx \frac{1}{M} \sum_{m=1}^{M} \log \frac{q(x_m)}{p(x_m)}$, where each of the x_m's are realizations of independent r.v.'s with distribution q. Since both p and q are gaussian mixtures, this is easy to simulate.

When computing the KL divergence, in order for $z_{\mathcal{I}}$ to be a pdf we need $\alpha_{\mathcal{I}}$ to be a pmf. We have taken two approaches to achieve this. In the first one we set to zero all $\alpha_{\mathcal{I},i}$'s with negative values and then renormalize so as to have the $\alpha_{\mathcal{I},i}$'s add up to one. The second approach is like the first one, but we additionally take a preparatory step, in which we impose the constraint $\sum_{i \in \mathcal{I}} \alpha_i = 1$ to problem (4), which can be shown to yield

$$\alpha_{\mathcal{I}} = K_{\mathcal{I}}^{-1} \left(\kappa_{\mathcal{I}} + \frac{\left(1 - \mathbf{1}^T K_{\mathcal{I}}^{-1} \kappa_{\mathcal{I}}\right)}{\mathbf{1}^T K_{\mathcal{I}}^{-1} \mathbf{1}} \mathbf{1} \right), \tag{8}$$

where $\mathbf{1}$ is the vector in $\mathbb{R}^k$ of all ones. We will use the symbol $\alpha^{(1)}_{\mathcal{I}}$ for the result of the first approach, and $\alpha^{(2)}_{\mathcal{I}}$ for the result of the second one. For each method of computing $\alpha_{\mathcal{I}}$, we use the Wilcoxon signed rank test [22] to compare the random and k-center methods across the datasets.

To determine the kernel parameter σ, we used a data dependent heuristic. We let σ be the median distance to the N^{th} nearest neighbor, for $N = 3, 5, 7, 9, 11$. To evaluate which σ performs better, we considered $E^2_{\mathcal{I}_j}$, for $5 \leq j \leq 100$, and measured the ratio of the drop between $j = 5$ and $j = 20$ to the drop between $j = 5$ and $j = 100$. We then chose the σ that maximized this ratio.

4.2. Results

We can see from Table 1 that the k-center algorithm outperforms random selection under the $D(q\|p)$ performance measure for almost all data sets when $j = 50$. To understand why the KL divergence $D(q\|p)$ is larger for the random algorithm than for k center, note that a large $D(q\|p)$ will result when the estimated distribution p does not capture information from the

Table 1: $\mathrm{D}(\bar{z}\|z_\alpha)$

	$\alpha_{\mathcal{I}}^{(1)}$		$\alpha_{\mathcal{I}}^{(2)}$	
	rand	k-cent	rand	k-cent
Iris	0.0725	0.0036	0.0580	0.0070
Banana	0.3225	0.0092	0.3250	0.0080
Image	65535	5.7055	65535	6.3122
Ringnorm	0.0142	0.0351	0.0212	0.0402
Breast Cancer	0.0698	0.0057	0.0657	0.0098
Heart	0.0137	0.0050	0.0124	0.0126
Thyroid	0.5120	0.0068	0.4957	0.0065
Diabetes	0.0494	0.0469	0.0467	0.0449
German	0.0304	0.0325	0.0345	0.0441
Twonorm	0.0108	0.0142	0.0083	0.0161
Waveform	0.0202	0.0195	0.0135	0.0205
C500	0.3048	0.0399	0.3538	0.0384
C2000	15.546	11.443	15.371	11.373
G3	0.0691	0.0022	0.0677	0.0015
G5	0.0143	0.0041	0.0136	0.0034

Values of $\mathrm{D}(\bar{z}\|z_\alpha)$ for different data sets and for $k = 50$. The Wilcoxon test for the first two columns gives a p-value of .0054, and for the last two columns $p = .022$

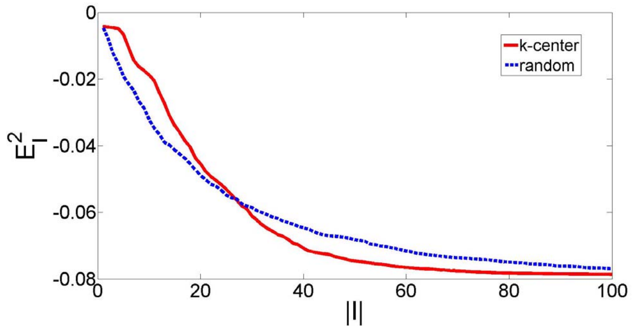
sample mean q. Recall that the random algorithm picks elements from densely populated areas, which in turn causes a poor representation of the tails of q, while the k-center algorithm, having elements far from each other, is able to capture well the tails of q. Note that the resulting p-values are under the typical significant level of .05.

The random algorithm usually outperforms the k-center method under the $E_{\mathcal{I}}^2$ performance measure when $|\mathcal{I}|$ is small, as seen on Figure 1. The initial advantage of random may be due to the fact that at the beginning it picks elements in dense areas, which will represent the mean better than outlying points. Meanwhile, the k-center agorithm begins by selecting points near the boundary of the data support, corresponding to the tails of the distribution. The improvement of the k-center algorithm is noticeable once it has added enough elements so as to approximate the full support of the distribution, while the convergence of the random algorithm slows, since it takes longer to select points out in the tails of the distribution.

5. CONCLUSION

We have shown that for certain radial kernels, the k-center algorithm can be applied to solve the original sparse approximation problem with time complexity $O(nkd+k^3)$. We anticipate that our approach will extend to online and divide-and-conquer implementations, because corresponding algorithms for the k-center problem exist for these settings [23–25].

Fig. 1: k vs rand



Comparison of $E_{\mathcal{I}}^2$ between the random algorithm and the k-center algorithm for the Banana data set.

6. PROOF OF THEOREM

Proof of Theorem 1. The beginning of this proof is similar to the one in [18]. Let $S_{\mathcal{I}} := \{z_i \,|\, i \in \mathcal{I}\}$ and denote $P_{\mathcal{I}}$ the projection operator onto $S := \mathrm{span}(S_{\mathcal{I}})$ and I the identity operator. For optimal α, we have

$$\|\bar{z} - z_{\mathcal{I}}\| = \|\bar{z} - P_{\mathcal{I}}\bar{z}\| = \frac{1}{n}\|\sum_{i\in[n]}(I - P_{\mathcal{I}})z_i\|$$

$$\leq \frac{1}{n}\sum_{i\in[n]}\|(I - P_{\mathcal{I}})z_i\| = \frac{1}{n}\sum_{i\notin\mathcal{I}}\|(I - P_{\mathcal{I}})z_i\|$$

where we have used the triangle inequality, and the last equality is due to the fact that $z_i = P_{\mathcal{I}}z_i$ when $z_i \in S_{\mathcal{I}}$.

Now, since $(z_i - P_{\mathcal{I}}z_i) \perp P_{\mathcal{I}}z_i$, we can use Pythagoras' Theorem in V to get $\|z_i - P_{\mathcal{I}}z_i\|^2 = \|z_i\|^2 - \|P_{\mathcal{I}}z_i\|^2$.

It can be shown that $\|P_{\mathcal{I}}z_i\| = \max\limits_{z\in S,\,\|z\|=1}\langle z_i, z\rangle$. Therefore, for $i \notin \mathcal{I}$,

$$\begin{aligned}\|P_{\mathcal{I}}z_i\| &= \frac{1}{\sqrt{C}}\max_{z\in S,\|z\|=\sqrt{C}}\langle z_i, z\rangle\\ &\geq \frac{1}{\sqrt{C}}\max_{\ell\in\mathcal{I}}\langle z_i, z_\ell\rangle\\ &\geq \frac{1}{\sqrt{C}}\min_{j\notin\mathcal{I}}\max_{\ell\in\mathcal{I}}\langle z_j, z_\ell\rangle = \frac{1}{\sqrt{C}}\nu_{\mathcal{I}}.\end{aligned}$$

Thus, for $i \notin \mathcal{I}$,

$$\|z_i\|^2 - \|P_{\mathcal{I}}z_i\|^2 \leq C - \frac{\nu_{\mathcal{I}}^2}{C}$$

and finally

$$\|\bar{z} - z_{\mathcal{I}}\| \leq \frac{1}{n}\sum_{i\notin\mathcal{I}}\sqrt{C - \frac{\nu_{\mathcal{I}}^2}{C}} = \left(1 - \frac{|\mathcal{I}|}{n}\right)\sqrt{\frac{1}{C}(C^2 - \nu_{\mathcal{I}}^2)}\,.$$

□

7. REFERENCES

[1] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[2] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[3] G. S. Watson, "Smooth regression analysis," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

[4] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International journal of computer vision*, vol. 24, no. 2, pp. 137–154, 1997.

[5] A. Mandal and A. Cichocki, "Non-linear canonical correlation analysis using alpha-beta divergence," *Entropy*, vol. 15, no. 7, pp. 2788–2804, 2013.

[6] D. M. Titterington, "Minimum distance non-parametric estimation of mixture proportions," *Journal of the Royal Statistical Society*, vol. 45, no. 1, pp. 37–46, 1983.

[7] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, 1975.

[8] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[9] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," in *Advances in Neural Information Processing Systems 24*, pp. 2178–2186. 2011.

[10] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 664–671.

[11] X. Hong, S. Chen, and C. J. Harris, "A forward-constrained regression algorithm for sparse kernel density estimation," *Neural Networks, IEEE Transactions on*, vol. 19, no. 1, pp. 193–198, 2008.

[12] M. Schafföner, E. Andelic, M. Katz, S. E. Krüger, and A. Wendemuth, "Memory-effcient orthogonal least squares kernel density estimation using enhanced empirical cumulative distribution functions," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 428–435.

[13] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381–396, 2000.

[14] D. W. Scott and W. F. Szewczyk, "From kernels to mixtures," *Technometrics*, vol. 43, no. 3, pp. 323–335, 2001.

[15] D. Schieferdecker and M. F. Huber, "Gaussian mixture reduction via clustering," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 1536–1543.

[16] P. Bruneau, M. Gelgon, and F. Picarougne, "Parsimonious reduction of gaussian mixture models with a variational-bayes approach," *Pattern Recognition*, vol. 43, no. 3, pp. 850–858, 2010.

[17] A. R. Runnalls, "Kullback-leibler approach to gaussian mixture reduction," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, no. 3, pp. 989–999, 2007.

[18] Z. Noumir, P. Honeine, and Cédric R., "One-class machines based on the coherence criterion," in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*. IEEE, 2012, pp. 600–603.

[19] V. V. Vazirani, *Approximation algorithms*, springer, 2001.

[20] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[21] D. S. Hochbaum, *Approximation algorithms for NP-hard problems*, PWS Publishing Co., 1996.

[22] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[23] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, "Incremental clustering and dynamic information retrieval," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM, 1997, pp. 626–635.

[24] S. Dasgupta, C. Papadimitriou, and U. V. Vazirani, *Algorithms*, McGraw-Hill Science/Engineering/Math, 2006.

[25] S. Dasgupta, "Online and streaming algorithms for clustering," `http://cseweb.ucsd.edu/~dasgupta/291-geom/streaming.pdf`, 2013, Accessed 4-November-2013.